计 算 机 科 学 丛 书

离散数学导学

(英) Andrew Simpson 牛津大学 著 冯速 译

INTERNATIONAL EDITION

Discrete Mathematics by Example

ANDREW SIMPSON

Discrete Mathematics by Example

机械工业出版社
China Machine Press

本书是牛津大学离散数学教材。与传统方法不同的是，本书通过大量的示例和练习，辅以相应的教学方法，使学生在实践中深化对离散数学的理解。

本书特点

- 通过大量易懂的范例和习题介绍离散数学这一课程。
- 把理论知识和一系列实际应用联系起来，为理论计算机科学提供了坚实的基础。
- 采用了Z形式描述技术中的演绎系统，为描述和验证计算机系统奠定了数学基础。

作者简介

Andrew Simpson 目前任教于牛津大学，担任软件工程项目主任。他拥有牛津大学的计算机专业硕士和博士学位。他曾在牛津大学、牛津鲁斯金学院和北伦敦大学讲授各种层次离散数学课程，拥有丰富的教学经验和理论知识，以在教学中善于结合实际需求著称。

ISBN 7-111-15396-0

9 787111 153962

封面设计：陈子平

HZ BOOKS

华章图书

华章网站 http://www.hzbook.com

网上购书：www.china-pub.com

北京市西城区百万庄南街1号 100037

读者服务热线：(010)68995259，68995264

读者服务信箱：hzedu@hzbook.com

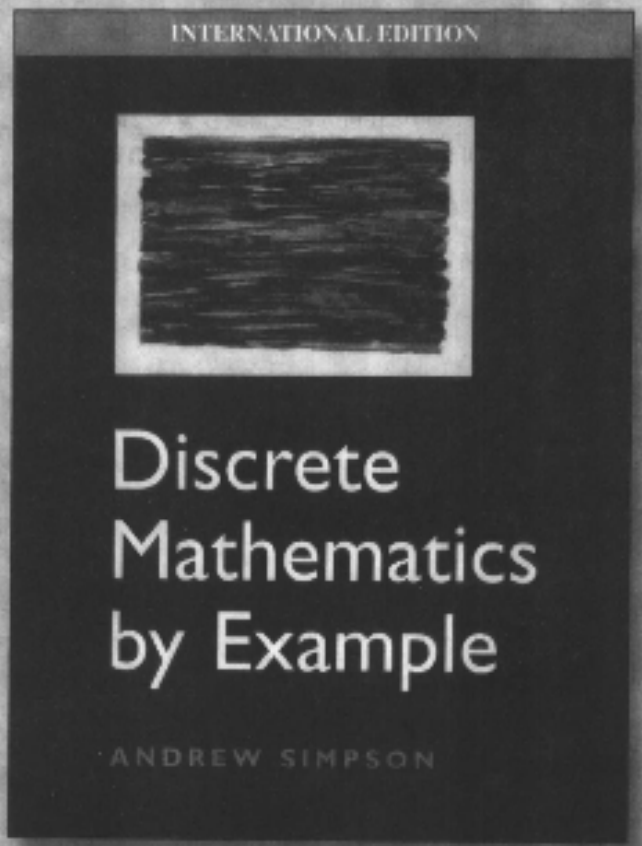
ISBN 7-111-15396-0/O · 383

定价：29.00元

计 算 机 科 学 丛 书

离散数学导学

(英) Andrew Simpson 牛津大学 著 冯速 译



Discrete Mathematics by Example

机械工业出版社
China Machine Press

本书通过大量简单易懂的示例和练习介绍了有关离散数学的基本概念与基础知识，并把理论知识与一系列实际应用联系起来。主要内容包括：命题逻辑和谓词逻辑、集合论、布尔代数、关系、函数、序列、归纳法、图论、组合数学等。通过适当的教学方法，可以加深学生对离散数学的理解。

本书适合所有学习离散数学的学生，并可作为相关专业的教材。

Andrew Simpson, Discrete Mathematics by Example(ISBN 0-07-122914-0).

Copyright © 2002 by McGraw-Hill International (UK) Limited.

Original English edition published by The McGraw-Hill Companies, Inc. All rights reserved. No part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher.

Simplified Chinese translation edition jointly published by McGraw-Hill Education (Asia) Co. and China Machine Press.

本书中文简体字翻译版由机械工业出版社和美国麦格劳-希尔教育(亚洲)出版公司合作出版。未经出版者预先书面许可，不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 McGraw-Hill 公司防伪标签，无标签者不得销售。

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：01-2003-8119

图书在版编目(CIP)数据

离散数学导学/(英)辛普森(Simpson, A.)著；冯速译．－北京：机械工业出版社，2005.1

(计算机科学丛书)

书名原文：Discrete Mathematics by Example

ISBN 7-111-15396-0

Ⅰ.离…　Ⅱ.①辛…②冯…　Ⅲ.离散数学　Ⅳ.O158

中国版本图书馆 CIP 数据核字(2004)第 107910 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

北京牛山世兴印刷厂印刷·新华书店北京发行所发行

2005 年 1 月第 1 版第 1 次印刷

787mm×1092mm 1/16·15 印张

印数：0 001-4 000 册

定价：29.00 元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：(010)68326294

专家指导委员会

（按姓氏笔画顺序）

尤晋元	王　珊	冯博琴	史忠植	史美林
石教英	吕　建	孙玉芳	吴世忠	吴时霖
张立昂	李伟琴	李师贤	李建中	杨冬青
邵维忠	陆丽娜	陆鑫达	陈向群	周伯生
周立柱	周克定	周傲英	孟小峰	岳丽华
范　明	郑国梁	施伯乐	钟玉琢	唐世渭
袁崇义	高传善	梅　宏	程　旭	程时端
谢希仁	裘宗燕	戴　葵		

秘　书　组

武卫东　　温莉芳　　刘　江　　杨海玲

译者序

离散数学是计算机及有关学科的一门最重要的基础课程。正因如此，目前已经有许多离散数学的教科书和教学参考书，其中既有国内学者编写的，也有从国外教材翻译过来的，还有直接引进的原版、影印版图书。而本书在这诸多的教科书中又添加了一本。

然而，与绝大多数离散数学的教科书相比，本书至少有两个不同的地方。

首先，本书的作者听到了来自不同读者群的两个共同的呼声：我们需要更多的例子来掌握这些复杂多样的概念和规则，我们需要知道离散数学与计算的关系。本书对这两个呼声给予了充分的回应。本书不是对定义和规则的罗列，而是对每一个概念和规则均给出了具体的、通常是日常生活或计算中的示例，同时又给出了相应的练习，让读者通过示例和练习在不知不觉中掌握这些概念和规则，领会它们与计算的关系。不仅如此，本书还专门辟出一章来介绍、讲解离散数学在各种类型的系统中的建模以及在相应模型中进行推理的应用，包括对程序设计语言中变量的建模、对现今广为运用的搜索引擎的抽象建模、对数据结构的抽象描述、对数字电路的建模以及对数据库和知识库系统的抽象描述。

其次，本书采用了Z形式描述技术中的演绎系统，并在书中大量使用了这一描述技术。Z形式描述技术是一种新颖、强大的描述技术，通过采用这样一种描述技术，我们既可以了解计算机理论的前沿研究，又可以领会理论研究的重要性和实用性，对培养理论研究的能力、提高理论研究的兴趣都有极大的益处。

这些特色使本书从本质上不同于其他离散数学的参考书，不论是作为教科书还是作为教师和学生的参考用书都具有非常大的价值。

译　者

2004年9月

出版者的话

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，还揭橥了学术的源变，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，华章公司就将工作重点放在了遴选、移译国外优秀教材上。经过几年的不懈努力，我们与Prentice Hall，Addison-Wesley，McGraw-Hill，Morgan Kaufmann等世界著名出版公司建立了良好的合作关系，从它们现有的数百种教材中甄选出Tanenbaum，Stroustrup，Kernighan，Jim Gray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、研究及庋藏。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，有的还专诚为其书的中译本作序。迄今，“计算机科学丛书”已经出版了近百个品种，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，为进一步推广与发展打下了坚实的基础。

随着学科建设的初步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步入一个新的阶段。为此，华章公司将加大引进教材的力度，在“华章教育”的总规划之下出版三个系列的计算机教材：除“计算机科学丛书”之外，对影印版的教材，则单独开辟出“经典原版书库”；同时，引进全美通行的教学辅导书“Schaum's Outlines”系列组成“全美经典学习指导系列”。为了保证这三套丛书的权威性，同时也为了更好地为学校和老师们服务，华章公司聘请了中国科学院、北京大学、清华大学、国防科技大学、复旦大学、上海交通大学、南京大学、浙江大学、中国科技大学、哈尔滨工业大学、西安交通大学、中国人民大学、北京航空航天大学、北京邮电大学、中山大学、解放军理工大学、郑州大学、湖北工学院、中国国家信息安全测评认证中心等国内重点大学和科研机构在计算机的各个领域的著名学者组成“专家指导委员会”，为我们提供选题意见和出版监督。

这三套丛书是响应教育部提出的使用外版教材的号召，为国内高校的计算机及相关专业

的教学度身订造的。其中许多教材均已为M. I. T.，Stanford，U.C. Berkeley，C. M. U. 等世界名牌大学所采用。不仅涵盖了程序设计、数据结构、操作系统、计算机体系结构、数据库、编译原理、软件工程、图形学、通信与网络、离散数学等国内大学计算机专业普遍开设的核心课程，而且各具特色——有的出自语言设计者之手、有的历经三十年而不衰、有的已被全世界的几百所高校采用。在这些圆熟通博的名师大作的指引之下，读者必将在计算机科学的宫殿中由登堂而入室。

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，但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。教材的出版只是我们的后续服务的起点。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，我们的联系方法如下：

电子邮件：hzedu@hzbook.com
联系电话：（010）68995264
联系地址：北京市西城区百万庄南街1号
邮政编码：100037

华章教育 赤诚奉献

经典原版书库（相关部分）

具体数学：计算机科学基础（第2版）	Graham/49.00
组合数学（第3版）	Brualdi/35.00
离散数学及其应用（第5版）	Rosen/79.00
高等微积分	Fitzpatrick/69.00
金融数学	Stampfli/35.00
数值分析（第3版）	Kincaid/75.00
概率论及其在投资、保险、工程中的应用	Bean/55.00
代数	Isaacs/65.00
傅里叶分析与小波分析导论	Pinsky/49.00
数学建模（第3版）	Giordano/59.00（附光盘）
微分方程与边界值问题（第5版）	Zill/69.00
应用回归分析和其他多元方法	Kleinbaum/88.00
概率统计	Stone/89.00
多元数据分析	Lattin/69.00（附光盘）
数理统计和数据分析	Rice/78.00
随机过程导论	Kao/49.00
预测与时间序列（第3版）	Bowerman/89.00
线性代数	Jain/49.00（附光盘）
复变函数及应用（第7版）	Brown/42.00
实分析与复分析（第3版）	Rudin/39.00
数学分析原理（第3版）	Rudin/35.00
泛函分析（第2版）	Rudin/42.00
复分析（第3版）	Ahlfors/35.00
逼近论教程	Cheney/39.00
拓扑学（第2版）	Munkres/59.00
数理金融初步（第2版）	Ross/29.00
纯数学教程（第10版）	Hardy/65.00
初等数论与及应用（第4版）	Rosen/59.00
代数	Artin/59.00
实分析（第3版）	Royden/45.00
流体动力学导论	Batchelor/66.00
组合数学教程	van Lint/58.00
曲线与曲面的微分几何	Carmo/49.00
三角级数（第3版）	Zygmund/88.00
时间序列分析的小波方法	Percival /58.00
数学分析（第2版）	Apostol/49.00
抽象代数基础教程（第2版）	Rotman/49.00
分形分析	Kigami/35.00
图论	Tutte/45.00
计数组合学（卷1）	Stanley/39.00
计数组合学（卷2）	Stanley/59.00
同调代数导论	Weibel/45.00
数学简史	Katz/55.00
图论导引（第2版）	West/59.00
复分析基础及工程应用（第3版）	Saff/59.00
线性代数（第6版）	Leon/59.00

华章数学译丛

数学分析原理（原书第3版）	Rudin/赵慈庚/28.00
金融数学	Stampfli/蔡明超/26.00
泛函分析	Rudin/刘培德/35.00

计算机科学丛书（相关部分）

离散数学及其应用（原书第4版）	Rosen/袁崇义/75.00
组合数学（原书第3版）	Brualdi/冯舜玺/38.00

教师反馈表

McGraw-Hill公司是美国著名的教育图书出版公司，出版了很多著名的计算机、工程类、经济管理类以及人文社科类图书。

我们十分重视对广大教师的服务，开发了教师手册、习题解答等教学课件以及网上资源。如果您确认将本书作为指定教材，请您务必填好以下表格并经系主任签字盖章后寄回我们的联系地址，McGraw-Hill公司将免费向您提供英文原版的教师手册或其他教学课件。

您需要教辅的教材：			
姓名：			
系：			
院/校：			
您所教的课程名称：			
学生人数/学期：	_____人/_____年级	学时：	
您目前采用的教材：	作者： _____ 书名：		
您准备何时用此书授课：			
联系地址：			
邮政编码：		联系电话：	
E-mail:			
您对本书的建议：		系主任签字： 盖章	

我们的联系地址：

McGraw-Hill Beijing Office
麦格劳－希尔北京代表处
北京市海淀区知春路76号
翠宫饭店写字楼1408室　北京100086
Tel：800-810-1936　ext.6023/6032
Fax：010-6263 8354
E-mail: webmaster @ mcgraw-hill.com.cn

目　录

出版者的话

专家指导委员会

译者序

第 1 章　导论 …… 1

1.1　学习动机 …… 1

1.2　教材内容 …… 1

1.3　组织结构 …… 2

第 2 章　数 …… 5

2.1　自然数 …… 5

2.2　Peano 算术 …… 7

2.3　其他类型的数 …… 8

2.4　附加练习 …… 8

2.5　练习解答 …… 8

第 3 章　命题逻辑 …… 11

3.1　原子命题 …… 11

3.2　真值 …… 11

3.3　否定运算符 …… 12

3.4　合取运算符 …… 13

3.5　析取运算符 …… 14

3.6　蕴含运算符 …… 16

3.7　等值运算符 …… 17

3.8　运算的优先级 …… 18

3.9　重言式、矛盾式和不定式 …… 18

3.10　真值表 …… 19

3.11　等值推理 …… 20

3.12　自然演绎 …… 21

3.13　附加练习 …… 24

3.14　练习解答 …… 25

第 4 章　集合论 …… 33

4.1　集合 …… 33

4.2　单集 …… 34

4.3　空集 …… 34

4.4　集合成员 …… 34

4.5　子集 …… 35

4.6　超集 …… 37

4.7　集合的并集 …… 38

4.8　集合的交集 …… 39

4.9　集合的差集 …… 40

4.10　有关集合的推理 …… 41

4.11　集合的势 …… 42

4.12　有穷集合和无穷集合 …… 42

4.13　集合的幂集 …… 43

4.14　集合的广义运算 …… 44

4.15　附加练习 …… 45

4.16　练习解答 …… 46

第 5 章　布尔代数 …… 51

5.1　简介 …… 51

5.2　命题逻辑回顾 …… 51

5.3　集合论回顾 …… 51

5.4　布尔代数基础 …… 52

5.5　简写规定 …… 54

5.6　优先级 …… 54

5.7　集合的布尔代数 …… 54

5.8　命题的布尔代数 …… 54

5.9　布尔代数的同构 …… 54

5.10　对偶性 …… 55

5.11　附加练习 …… 55

5.12　练习解答 …… 56

第 6 章　类型集合论 …… 61

6.1　类型的需要 …… 61

6.2　再论空集 …… 63

6.3　集合描述 …… 64

6.4　特征组 …… 65

6.5　缩写 …… 66

6.6　笛卡儿积 …… 66

6.7　公理定义 …… 70

6.8　附加练习 …… 71

6.9　练习解答 …… 72

第 7 章　谓词逻辑 …… 77

7.1　量词的需要 …… 77

7.2　全称量词 …… 77

7.3　存在量词 …… 78

7.4　可满足性和有效性 …… 80

7.5　量词的否定 …… 80

7.6　自由变元和约束变元 …… 81

7.7　替换 …… 82

7.8 限制 …… 84
7.9 唯一存在量词 …… 86
7.10 等值推理 …… 87
7.11 自然演绎 …… 89
7.12 one-point 规则 …… 91
7.13 附加练习 …… 92
7.14 练习解答 …… 93
第 8 章 关系 …… 101
8.1 二元关系 …… 101
8.2 关系的推理 …… 102
8.3 定义域和值域 …… 102
8.4 关系的逆 …… 103
8.5 关系上的运算 …… 104
8.6 关系的合成 …… 107
8.7 同类关系和异类关系 …… 109
8.8 关系的性质 …… 109
8.8.1 自反性 …… 109
8.8.2 传递性 …… 110
8.8.3 对称性 …… 111
8.8.4 非对称性 …… 112
8.8.5 反对称性 …… 112
8.8.6 完全性 …… 113
8.9 顺序与等价 …… 113
8.9.1 偏序 …… 113
8.9.2 全序 …… 114
8.9.3 等价关系 …… 114
8.10 闭包 …… 114
8.10.1 自反闭包 …… 114
8.10.2 传递闭包 …… 115
8.10.3 自反传递闭包 …… 116
8.10.4 对称闭包 …… 116
8.11 n 元关系 …… 117
8.12 附加练习 …… 117
8.13 练习解答 …… 119
第 9 章 函数 …… 127
9.1 一种特殊的关系 …… 127
9.2 全函数 …… 128
9.3 函数的作用 …… 129
9.4 覆盖 …… 130
9.5 函数的性质 …… 131
9.5.1 单射 …… 131
9.5.2 满射 …… 132
9.5.3 双射 …… 133
9.6 递归函数 …… 133
9.7 附加练习 …… 135
9.8 练习解答 …… 136
第 10 章 序列 …… 141
10.1 元包 …… 141
10.2 顺序的需要 …… 142
10.3 建模序列 …… 142
10.4 空序列 …… 143
10.5 长度 …… 143
10.6 连接 …… 144
10.7 头和尾 …… 144
10.8 限制 …… 145
10.9 逆置 …… 147
10.10 单射序列 …… 148
10.11 再论递归函数 …… 148
10.12 附加练习 …… 149
10.13 练习解答 …… 151
第 11 章 归纳法 …… 155
11.1 数学归纳法 …… 155
11.2 结构归纳法 …… 157
11.3 附加练习 …… 158
11.4 练习解答 …… 158
第 12 章 图论 …… 163
12.1 图 …… 163
12.2 图的集合和元包表示 …… 167
12.3 图的矩阵表示 …… 168
12.4 图的同构 …… 170
12.5 路径 …… 171
12.6 循环 …… 173
12.7 树 …… 174
12.8 带权图 …… 175
12.9 有向图 …… 177
12.10 二叉树 …… 178
12.11 附加练习 …… 180
12.12 练习解答 …… 181
第 13 章 组合数学 …… 187
13.1 阶乘函数 …… 187
13.2 二项式系数 …… 188
13.3 计数 …… 190
13.4 排列 …… 190
13.5 组合 …… 192
13.6 树形图 …… 193

13.7 取样 …… 193
13.8 附加练习 …… 195
13.9 练习解答 …… 195
第 14 章 应用实例 …… 203
14.1 程序变量的建模 …… 203
14.2 元搜索引擎 …… 204
14.2.1 使用集合建模 …… 204
14.2.2 使用序列建模 …… 205
14.2.3 使用关系建模 …… 205
14.2.4 继续讨论 …… 206
14.3 用于栈和队列的序列 …… 208
14.4 数字电路 …… 210
14.5 学校数据库 …… 212
14.6 知识库系统 …… 214
14.7 练习解答 …… 216
参考文献 …… 227

第1章 导 论

1.1 学习动机

对于许多计算机科学及相关学科的学生来说，离散数学可能是一门非常难学的课程。同时，它也是一门非常难教的课程。这门课程之所以难学、难教，有以下三个主要原因。第一，对于很多学生，即便是那些有相当数学背景的学生，学习离散数学通常意味着要以全新的思考方式来考虑问题。第二，也是更有争议的原因是，现在的计算专业大学生在开始大学生活之前没有接受足够的数学训练。第三，也是最根本的原因是，离散数学课程中所教授的一些技术相对比较复杂。

作者曾在四所学校(牛津大学、牛津布鲁克斯大学、伦敦北方大学以及牛津大学鲁斯金学院)对各个层次的学生讲授过离散数学课程，讲授对象包括本科生、研究生和预科生，包括全日制学生和业余学生。不论是哪个层次、哪种类型的学生，在学习离散数学的学生中总能听到两个熟悉的呼声。

第一个熟悉的呼声是：

"我想我理解这门技术，但是我需要更多的例子和习题来加以确认。"

以作者的观点，通过范例学习是很多学生学习离散数学的最佳途径。从各种意义上讲，离散数学是一种"碰撞型运动"，为了精通离散数学，学生需要大量的练习。现在并不缺乏离散数学的教学参考书，但是，除个别外，绝大多数离散数学的教材都没有提供足够的练习和范例来帮助那些觉得离散数学难学的学生。

第二个熟悉的呼声是：

"这与计算有什么关系？"

许多为计算专业的学生所写的离散数学教材都没能把所教的内容和计算的世界很好地联系起来。虽然从学术的角度上看，许多这方面的教材都很优秀，但是，它们没有向读者充分展示所讲的技术与计算应用的关联。

因此，本书有两个目的：

- 第一，通过易懂的范例和习题介绍离散数学这一科目(第 2 章至第 13 章)。
- 第二，把第 2 章至第 13 章所介绍的理论知识和一系列实际应用联系起来(第 14 章)。

本书在每章的最后给出了该章所有练习的解答，以便尽可能为学生提供帮助。衷心希望读者不要滥用这一特殊的安排，不要过早翻看解答。

1.2 教材内容

本书在教材内容上的选择受到了以下事实的影响：以作者的观点，在计算机科学中离散数学有两个用途。第一个是为理论计算机科学提供坚实的基础。大多数"传统"离散数学课程以此为目标进行教学。离散数学在计算机科学中的第二个用途是为形式描述技术奠定数学基础，而形式描述技术则是描述和验证计算机系统的数学表示法。一些离散数学的课程，特别是在英国新式大学所教授的离散数学课程，是特意为此目的而开设的。这一影响可在第 3 章找到，那里介绍的自然演绎系统是 Z 形式描述技术中的演绎系统(参见[WD96])。作者的意图是使本书适合于两种类型的离散数学课程。因此，也许传统离散数学教材的读者对讨论序列的章

节感到意外，这些读者可以跳过这些内容，如第 14 章，那里展示了若干序列的应用实例。另外，希望学习类型集合论和作为形式方法的谓词逻辑的读者也许会对诸如布尔代数、图论、组合学的有关章节感到诧异，这些读者可以跳过这些章节。

有两个因素影响了参考文献的筛选。第一个因素是，由于本书的性质，对于某些课程或读者来说，书中的某些内容可能深度不够；对于这种情况，作者给出了进一步学习或练习用的参考文献。这种情况主要出现在第 12 章和第 13 章，限于篇幅，作者对图论和组合学这两个复杂的主题只做了较浅的论述，这可能导致一些读者渴望进一步的学习。对这两个主题，作者给出了辅助教材的建议。第二个因素是，近年“科普”书籍普遍流行，例如，著名科学家、哲学家和数学家的传记以及数学定理(甚至是数)的历史等都登上了畅销书的排行榜。基于这一形势，作者在本书的适当位置为感兴趣的读者指出了在离散数学的各种课题的发展中起到关键作用的人物的生平及其研究的参考资料○。

1.3 组织结构

本书的组织结构如下：

第 2 章介绍数的基本概念。我们特别关注自然数和 Peano 算术，使用 Peano 算术的自然数表示导出证明和递归函数的核心概念，这些话题会在以后章节进一步讨论。

第 3 章讨论命题逻辑，包括原子命题和命题逻辑运算符。介绍若干建立给定命题真假性的方法：替换、真值表、证明树和等值推理。虽然通过证明树(或自然演绎)的证明是建立命题逻辑的定理的一般方法，但是可以使用的规则却通常与特定的系统相关。我们选择的系统是与形式描述相关的 Z 系统。

第 4 章讨论(无类型)集合论，介绍文氏图和基本集合运算符。

第 5 章讨论布尔代数。命题逻辑和集合论都是布尔代数的实例。在给出布尔代数的抽象概念后，展示它与命题逻辑和集合论的结构的关系。

第 6 章回到集合论的话题。指出无类型集合论的缺点，并介绍类型集合论。在建立类型集合论之后，继续讨论集合描述(set comprehension)和笛卡儿积等集合论的概念，这些概念是第 8、9、10 章的基础。

第 7 章在第 2 章和第 6 章的基础之上引出谓词逻辑。介绍存在量词和全称量词，然后讨论自由变元、约束变元以及替换。

在第 6 章介绍了笛卡儿积之后，第 8 章讨论关系。介绍关系的基本概念，然后讲述关系的运算和性质。

第 9 章描述函数的概念。函数是特殊的关系。与关系一样，讨论函数的运算和性质。另外，还讨论递归函数的高级话题。

第 10 章讨论序列。讲解序列的必要性及如何用函数表示序列，然后介绍序列上的运算。

第 11 章讨论如何运用数学归纳法和结构归纳法这两个强有力的技术来证明形式定义的结构的性质。其中，使用数学归纳法证明自然数的性质，使用结构归纳法证明序列等结构的性质。

第 12 章介绍图论。给出图论的基本术语，描述如何用矩阵和集合论表示图。然后讲解树、带权图、二叉树等在计算中起重要作用的特殊图。

第 13 章讨论组合数学的主题。组合数学主要讨论如何确定一系列事件的可能结果的数目。首先回顾阶乘函数等数学概念，然后介绍排列和组合，最后讲解使用树形图来表示可能结果的方法。

○ 在这个方面，[Gri99]是一个很好的参考文献。

最后，第 14 章展示如何将前面章节所学的技术运用于构筑各种系统的模型以及分析这些系统的性质，同时指出将前述话题与计算机科学相关联的一些方法。我们给出不同规模、不同复杂度的例子，试图通过将前面所讲述的内容植入到相关的计算来强调它们的重要性。首先，介绍如何将函数运用于一个小型程序设计语言的建模。然后，讲述万维网的搜索引擎以及如何建立搜索引擎的抽象模型。接着，讨论栈和队列的概念是如何与序列相关联的。从 14.4 节将看到第 5 章介绍的布尔代数的概念和技术可以运用于数字电路的设计。然后，通过展示如何使用谓词逻辑和类型集合论描述一个简单的数据库来示例形式描述。最后，展示在知识库管理系统中如何运用谓词逻辑。

致谢

我要感谢很多人，没有他们就没有本书。第一个要感谢的是 McGraw-Hill 的发行人，他们做出本书的计划，使其得以顺利进行。特别地，要感谢 David Hatter，他热情地草签了这一计划；感谢 Sarah Douglas 和 Conor Graham 在计划最后阶段的耐心和专业。衷心感谢过去几年来我在离散数学课程中所教授的所有学生们，特别是那些提出需要这样的教材的学生们。

与牛津布鲁克斯大学我的同事们的极具价值的讨论对本书的写作有很大的帮助。从 David Duce 那里我及时地获得了许多建设性的建议。另外，Ken Brownsey、Nigel Crook、Barry Holmes 以及 David Lightfoot 的全面的建议也很有用。与牛津大学的 Jim Davies、Andrew Martin 和 Jim Woodcock 以及约克大学的 Steve King 的长期交流对本书的结构和内容都有很大的帮助，特别是 Jim Davies 对本书的封面也提出了有益的建议。Paul Goddard 提出了一些见解深刻的建议。Jon Whiteley 也一样，他甚至对本书进行了彻底的校对。同样，我还要感谢 Duncan Brydon、Jon Hill、Rick McPhee 以及 Andrew Spencer。

我要感谢本书的审稿人，他们对本书的提案给予积极的响应，并提出了建设性、有时是非常友好的建议。他们对本书的审稿非常彻底，这使我非常欣慰。特别地，感谢 Clive Mingham，他所付出的努力超出了所应有的期待。

作者对本书中的所有错误负全面责任。

最后，还要感谢 Becky，在写作期间她对我一味迁就，并在精神上给了我很大的支持。

Andrew Simpson
2001 年 10 月于牛津

第2章　数

本章介绍以后各章所需的若干数学基本概念。首先介绍自然数。然后，讲述称之为Peano算术的自然数的形式表示，利用Peano算术引出证明的概念。最后，介绍另外两类数：整数和实数。

2.1　自然数

的确，离散数学的基础是集合论和命题逻辑，这些本书都将论述。然而，在学习这些论题之前，我们需要讨论一个更基本的概念：自然数(natural number)。

自然数是从0开始的非负整数，即0、1、2、3，等等。所有自然数所组成的集合记为$\mathbb{N}$[○]。

例 2.1　57在$\mathbb{N}$中出现，但-57、5.7及-5.7等不在$\mathbb{N}$中出现。 □

练习 2.1　下面哪些数是自然数？

1. 0
2. -1
3. 1.2 □

有两个自然数上的运算(operation)是读者也许不熟悉但在后面的学习中很重要的：div和mod。这两个运算本质上就是初中或高中所学的除运算和求余运算：对于自然数m和非0自然数n，m div n返回m除以n的整数部分，m mod n返回m除以n的余数部分。

例 2.2

7 div 3=2

7 mod 3=1 □

练习 2.2　计算下列各式。

1. 10 div 10
2. 10 div 1
3. 10 div 3
4. 10 mod 10
5. 10 mod 1
6. 10 mod 3 □

在后面章节中我们将使用的另外两个自然数上的运算符是求和(sum)和求积(product)运算符，分别记作$\sum$和$\prod$。前者是加运算符的扩展，后者是乘运算符的扩展。

例 2.3　求和运算符可用于表示1到5的自然数的和：

$$\sum_{i=1}^{5} i$$

其中，下标$i=1$表示变量i从1开始：整个式子列举i的所有可能取的值(从下标1到上标5)并求这些值的和。即

$$\begin{aligned}\sum_{i=1}^{5} i &= 1+2+3+4+5\\ &= 15\end{aligned}$$ □

例 2.4　式子$\sum_{i=5}^{10}(i^2)$表示5和10之间的所有自然数的平方和，包括5和10的平方。即

$$\begin{aligned}\sum_{i=5}^{10}(i^2) &= 5^2+6^2+7^2+8^2+9^2+10^2\\ &= 25+36+49+64+81+100\\ &= 355\end{aligned}$$ □

练习 2.3　计算下列各式。

1. $\sum_{i=1}^{5}(i^3)$
2. $\sum_{i=1}^{5}(i-1)$

○ 注意：有些离散数学的教科书不把0作为自然数；本书把0作为自然数。

3. $\sum_{i=1}^{5}(i \bmod 3)$ □

式子 $\sum_{i=1}^{5}(i^3)$ 表示对 i^3 的所有可能值求和，其中 i 在 1 和 5 之间取值，包括 1 和 5。但是，当我们把求和公式写成

$$\sum_{i=1}^{0}(i^3)$$

时，它表示对 i^3 的所有可能值求和，其中 i 在 1 和 0 之间取值。显然，不存在这样的取值。因此，求和的结果是 0。

下面的规则指出一般的情况。

规则 2.1 *给定任意的自然数 m 和 n 以及任意的数学公式 t，如果 $m<n$，则*

$$\sum_{i=n}^{m} t = 0$$ □

例 2.5

$$\sum_{i=1}^{0}(i^3) = 0$$ □

例 2.6

$$\sum_{i=10}^{3}(i-1) = 0$$ □

求积运算符 $\prod$ 的语法(书写格式)与求和运算符相同。

例 2.7 式子 $\prod_{i=1}^{5} i$ 表示 1 到 5 之间所有自然数的乘积，包括 1 和 5。即

$$\begin{aligned}\prod_{i=1}^{5} i &= 1\times2\times3\times4\times5\\ &= 120\end{aligned}$$ □

例 2.8

$$\begin{aligned}\prod_{i=5}^{10}(i\times2) &= (5\times2)\times(6\times2)\times(7\times2)\times(8\times2)\times(9\times2)\times(10\times2)\\ &= 10\times12\times14\times16\times18\times20\\ &= 9\ 676\ 800\end{aligned}$$ □

练习 2.4 计算下列各式。

1. $\prod_{i=1}^{5} i^3$
2. $\prod_{i=1}^{5}(i \operatorname{div} 2)$
3. $\prod_{i=1}^{5}\left(\sum_{j=i}^{i+2} j\right)$ □

与求和的情况相同，如果求积运算中的变量的上限小于下限，那么求积的结果是 0。

规则 2.2 *对于任意给定的自然数 m 和 n 以及任意的数学公式 t，如果 $m<n$，则*

$$\prod_{i=n}^{m} t = 0$$ □

例 2.9

$$\prod_{i=1}^{0}(i^3) = 0$$ □

例 2.10

$$\prod_{i=10}^{3}(i-1) = 0$$ □

将在第 12 章使用的另一个数学概念是对数(logarithm)，我们特别关心的是以 2 为底的对数(base 2 logarithm)。

给定任意自然数 n，$\log_2 n$ 表示以 2 为底的 n 的对数。这是满足下面等式的数 x：

$$2^x = n$$

例 2.11 因为 $2^3=8$，我们有 $\log_2 8=3$。因为 $2^5=32$ 及 $2^6=64$，我们有 $\log_2 32=5$ 及 $\log_2 64=6$。 □

更多的情况是，对于自然数 n，$\log_2 n$ 的值不是自然数。例如，可以从 $\log_2 2=1$ 和 $\log_2 4=2$ 得出 $\log_2 3$ 是 1 和 2 之间的某个数，显然 $\log_2 3$ 不是自然数。对于这种情况，我们可以“舍入”或“舍去”结果。例如，用 $\lfloor \log_2 3 \rfloor$ 表示不超过 $\log_2 3$ 的最大自然数，用 $\lceil \log_2 3 \rceil$ 表示不小于 $\log_2 3$ 的最小自然数。即 $\lfloor \log_2 3 \rfloor=1$，$\lceil \log_2 3 \rceil=2$。

练习 2.5 计算下列各式。

1. $\log_2 1$
2. $\log_2 2$
3. $\log_2 128$
4. $\lceil \log_2 50 \rceil$

5. $\lfloor \log_2 50 \rfloor$

6. $\lceil \log_2 70 \rceil$

7. $\lfloor \log_5 70 \rfloor$ □

因为绝大多数读者都熟悉自然数的概念，所以下一节将通过这些概念引出证明(proof)的概念。

2.2 Peano 算术

Peano 算术[1]由一组刻画自然数基本性质的规则组成。其中的 4 条规则，或称公理，如下所示(第 5 条公理将在第 11 章给出)。

1. 0 是自然数。

2. 若 x 是自然数，则 $x+1$ 也是自然数。

3. 不存在满足 $z+1=0$ 的自然数 z。

4. 给定自然数 x 和 y，若 $x+1=y+1$ 则 $x=y$。

前两条公理帮助我们"构造"自然数。首先，根据公理 1，0 是自然数。因为 0 是自然数，根据公理 2，0+1 也是自然数。因为 0+1 是自然数，根据公理 2，0+1+1 也是自然数。因为 0+1+1 是自然数，根据公理 2，0+1+1+1 也是自然数。当然，这一过程可以无休止地持续下去，构造出每个自然数。

第 3 条公理规定，构造自然数的过程确实是从 0 开始：没有比 0 小的自然数。

最后，公理 4 使我们可以决定两个自然数是否相等。

有一件重要的事情需要注意，Peano 算术表明可以用两个概念完整地定义自然数：唯一的元素 0 和一个运算+1。

练习 2.6 使用 Peano 算术的公理，证明 0+1 是自然数。 □

练习 2.6 的证明展示了如何使用 Peano 算术的规则证实 0+1 是自然数。Peano 算术形式地处理自然数，从而使得可以证明这样的性质；本书后面都是如此：提出形式系统，把它作为现实世界的实体的模型，同时，同样重要的是，证明这些形式表示的性质。

当然，如果只能确定两个自然数是否相等，那么对我们来说自然数的用途就很有限了。在第 9 章我们将看到函数(function)的概念；自然数上的函数包括减法、乘法以及除法等概念。当然，可以使用 Peano 算术定义这些函数。

例 2.12 可以如下定义自然数上的函数 *subtract*。

$subtract(x+1, 0)=x+1$

$subtract(x+1, y+1)=subtract(x, y)$

例如，

$subtract(0+1+1, 0)=0+1+1$

注意，只有当 m 不小于 n 时 $subtract(m, n)$ 才有定义，因为这里没有负数的概念。 □

练习 2.7 根据上述 *subtract* 的定义，计算

$subtract(0+1+1+1, 0+1)$ □

例 2.13 可以如下定义自然数上的函数 *add_two*。

$add\_two(0)=0+1+1$

$add\_two(x+1)=(x+1)+1+1$ □

subtract 和 *add_two* 都是递归函数(recursively defined function)的实例。我们将在第 9 章正式学习这样的函数。

练习 2.8 使用上面函数 *add_two* 的定义，证明

$add\_two(0+1)=0+1+1+1$ □

一般来说，证明需要一系列的步骤来确立要证的事实，而不仅仅是一个步骤，就如练习

[1] 以 Giuseppe Peano(1858—1932)命名。有兴趣的读者可以从[Ken80]获得关于 Peano 的生活和工作的记载。

2.8的解答那样。对于这种情况，我们使用一种切实可行、形式的格式来构造证明。下面是一个简单的证明示例(只含两个步骤)[1]。

例 2.14 下面的证明确立

$$add\_two(x+1+1)$$
$$= add\_two(x+1)+1$$

对所有自然数 x 均成立。

$$add\_two(x+1+1)$$
$$=(x+1+1)+1+1 \quad [add\_two \text{ 的定义}]$$
$$=(x+1+1+1)+1 \quad [+\text{的性质}]$$
$$=add\_two(x+1)+1 \quad [add\_two \text{ 的定义}]$$

□

例 2.15 下面的证明确立

$$subtract(add\_two(x+1),\ 0+1+1)$$
$$=x+1$$

对所有自然数 x 均成立。

$$subtract(add\_two(x+1),\ 0+1+1)$$
$$=subtract((x+1)+1+1,\ 0+1+1) \quad [add\_two \text{ 的定义}]$$
$$=subtract((x+1)+1,\ 0+1) \quad [subtract \text{ 的定义}]$$
$$=subtract(x+1,\ 0) \quad [subtract \text{ 的定义}]$$
$$=x+1 \quad [subtract \text{ 的定义}]$$

□

2.3 其他类型的数

我们已经看到，$\mathbb{N}$ 表示非负整数的集合。本书还将考虑另外两个数的集合，第一个是整数集合，记作$\mathbb{Z}$。$\mathbb{Z}$ 在包含所有非负整数的同时，还包含负整数。也就是说，$\mathbb{Z}$ 包含 0、1、−1、2、−2、3、−3，等等。另外，实数集合记作$\mathbb{R}$，除包含所有整数外，还包含所有十进制小数，如 1.2、−1.3、1.333 等等，以及$\sqrt{2}$这样的数。

由上可知，每个包含在$\mathbb{N}$中的数也包含在$\mathbb{Z}$中，每个包含在$\mathbb{Z}$中的数也包含在$\mathbb{R}$中。

练习 2.9 以下各数包含在$\mathbb{N}$、$\mathbb{Z}$和$\mathbb{R}$的哪个集合中？

1. 23
2. −23
3. 2.3
4. −2.3

□

练习 2.10 每个包含在$\mathbb{N}$中的数都包含在$\mathbb{R}$中吗？ □

2.4 附加练习

练习 2.11 使用+1定义“普通的”加法运算符。

练习 2.12 使用 Peano 算术定义$\mathbb{N}$上的运算 *minus_one*。假设 $minus\_one(0)=0$。

练习 2.13 使用 Peano 算术定义$\mathbb{N}$上的乘法运算符 *mult*。

练习 2.14 使用 Peano 算术和函数 *mult* 定义$\mathbb{N}$上的平方运算符 *square*。

练习 2.15 证明，对于任意的自然数 m 和 n，有

$$mult(add\_two(n),\ m)$$
$$=mult(n,\ m)+mult(add\_two(0),\ m)$$

2.5 练习解答

2.1 这些数中只有 0 在$\mathbb{N}$中出现。

2.2

1. 1
2. 10
3. 3
4. 0
5. 0
6. 1

[1] 想要获得书写数学证明的进一步指导的读者可以参考[Sol90]。

2.3

1. $\sum_{i=1}^{5} i^3 = 1^3 + 2^3 + 3^3 + 4^3 + 5^3$

$= 1 + 8 + 27 + 64 + 125$

$= 225$

2. $\sum_{i=1}^{5} (i-1) = (1-1) + (2-1) + (3-1) + (4-1) + (5-1)$

$= 0 + 1 + 2 + 3 + 4$

$= 10$

3. $\sum_{i=1}^{5} (i \bmod 3) = (1 \bmod 3) + (2 \bmod 3) + (3 \bmod 3) + (4 \bmod 3) + (5 \bmod 3)$

$= 1 + 2 + 0 + 1 + 2$

$= 6$

2.4

1. $\prod_{i=1}^{5} i^3 = 1^3 \times 2^3 \times 3^3 \times 4^3 \times 5^3$

$= 1 \times 8 \times 27 \times 64 \times 125$

$= 1\ 728\ 000$

2. $\prod_{i=1}^{5} (i \text{ div } 2) = (1 \text{ div } 2) \times (2 \text{ div } 2) \times (3 \text{ div } 2) \times (4 \text{ div } 2) \times (5 \text{ div } 2)$

$= 0 \times 1 \times 1 \times 2 \times 2$

$= 0$

3. $\prod_{i=1}^{5} \left(\sum_{j=i}^{i+2} j \right) = \sum_{j=1}^{3} j \times \sum_{j=2}^{4} j \times \sum_{j=3}^{5} j \times \sum_{j=4}^{6} j \times \sum_{j=5}^{7} j$

$= (1+2+3) \times (2+3+4) \times (3+4+5) \times (4+5+6) \times (5+6+7)$

$= 6 \times 9 \times 12 \times 15 \times 18$

$= 174\ 960$

2.5

1. 因为 $2^0 = 1$，所以 $\log_2 1 = 0$。

2. 因为 $2^1 = 2$，所以 $\log_2 2 = 1$。

3. 因为 $2^7 = 128$，所以 $\log_2 128 = 7$。

4. 因为 $2^5 = 32$ 而且 $2^6 = 64$，所以 $\lceil \log_2 50 \rceil = 6$。

5. 因为 $2^5 = 32$ 而且 $2^6 = 64$，所以 $\lceil \log_2 50 \rceil = 5$。

6. 因为 $2^6 = 64$ 而且 $2^7 = 128$，所以 $\lceil \log_2 70 \rceil = 7$。

7. 因为 $2^6 = 64$ 而且 $2^7 = 128$，所以 $\lceil \log_2 70 \rceil = 6$。

2.6 公理 1 表明 0 是自然数。公理 2 表明，如果 0 是自然数那么 0+1 也是自然数。因此，0+1 是自然数。

2.7

$subtract(0+1+1+1,\ 0+1)$

$= subtract(0+1+1,\ 0)$

$= 0+1+1$

2.8 根据 *add_two* 的定义，有

$add\_two(0+1) = 0+1+1+1$

2.9

1. [illegible]、[illegible] 和 [illegible]

2. [illegible] 和 [illegible]

3. [illegible]

4. [illegible]

2.10 是的。假设数 n 包含在 [illegible] 中。由上所述，如果 n 包含在 [illegible] 中，那么它包含在 [illegible] 中。因此，n 包含在 [illegible] 中。进而，仍由上所述，如果 n 包含在 [illegible] 中，那么它包含在 [illegible] 中。因此，n 包含在 [illegible] 中。

2.11

$0 + n = n$

$(m+1) + n = (m+n) + 1$

2.12

$minus\_one(0) = 0$

$minus\_one(n+1) = n$

2.13

$mult(0,\ n) = 0$

$mult(0+1, n)=n$

$mult(m+1, n)=mult(m, n)+n$

2.14

$square(n)=mult(n, n)$

2.15

$mult(add\_two(n),m)$

$=mult(n+1+1,m)$

[add_two 的定义]

$=mult(n+1,m)+m$

[$mult$ 的定义]

$=mult(n,m)+m+m$

[$mult$ 的定义]

$=mult(n,m)+mult(0+1,m)+m$

[$mult$ 的定义]

$=mult(n,m)+mult(0+1+1,m)$

[$mult$ 的定义]

$=mult(n,m)+mult(add\_two(0),m)$

[add_two 的定义]

第3章 命题逻辑

本章介绍命题逻辑。我们首先了解命题的概念，然后介绍命题逻辑的运算符，接着介绍证明这些命题有效性的若干方法。想进一步了解这方面问题的读者可以在学完本章的内容后先参考[BMN97]或[CLP00]，然后再参考[Men87]。

3.1 原子命题

在日常生活中，我们经常会遇到或者为真或者为假的陈述。在离散数学中，把这样的陈述称为命题(proposition)。命题所取的真值(truth value)或者是真的或者是假的(见例3.1)，也可能根据具体的条件而变化(见例3.2)。然而，事实是，我们可以尝试把真值与这样的陈述相关联。

例3.1 命题"星期二是星期三的前一天"的真值永远为真。 □

例3.2 命题"今天是星期二"的真值也许是真也许是假，当然，它由陈述的时间决定。 □

对于某些命题，不需要先决定其他命题的真假性就可以试着去决定它们的真假性。这一事实意味着我们可以将这样的命题称为原子命题(atomic proposition)。即它是一个基本的个体，其真假性独立于其他命题。

其他原子命题的例子如下：

- "吉姆是素食者。"
- "正在下雨。"
- "贝奇喜欢吃饼干。"
- "电视上什么都没有。"

当然，如果没有更详细的信息(例如：吉姆和贝奇的信息)，我们不能确定这些陈述是真的还是假的。而且，最后陈述的含糊性例证了为什么数学的形式性通常优于含糊的自然语言：这个陈述的意思是"电视上什么也没有显示"、"电视机的上面什么都没有"，还是"电视上没有什么值得看的东西"呢？

3.2 真值

每个原子命题都可与一个真值相关联。我们考虑的命题逻辑中，原子命题可以取两种真值：true 和 false。而且，每个命题在给定时刻所取的真值唯一。例如：陈述"星期三是紧接着星期二后的那一天"的真值为 true，而陈述"四月是紧接在六月之前的那个月份"的真值为 false。

虽然我们也许不能确定给定命题的真假，但是可以知道它一定或者是真的或者是假的。而且，虽然给定命题的真值与陈述的时间相关，但是，在同一时刻它的真值绝不可能既是真的又是假的。

例3.3 我们不知道命题"杰克喜欢吉尔"是真还是假，但是知道它一定或者是真的或者是假的。 □

例3.4 命题"今天是星期二"有时为真有时为假，但是它一定或者是真的或者是假的。 □

练习3.1 判断下列陈述的真假。

1. $0<1$
2. $1+1=2$
3. $1\times1=2$ □

某些命题逻辑理论涉及第三种真值：未定义(undefined)。这种三值逻辑(three-valued logic)也能把真值和命题相关联，这里命题的真假性是不确定的。例如，如果没有进一步的信息，我们很难判断陈述"杰里米很高兴"的真值。在三值逻辑中，可以方

便地把值未定义赋予这个陈述。然而，在本书中，我们只使用两个真值：true 和 false。因此，在我们的论述中，可以推断上述陈述的真值既可能是 true 也可能是 false，不过我们没有确定它到底是哪个的办法。

练习 3.2 下面哪些命题不需要更多的信息就能确定真值？

1. "乔恩的年龄大于 35 岁。"
2. "地球绕着月亮转。"
3. "3 大于 2。"
4. "乔恩(Jon)的名字含有三个字母。"

□

原子命题连同真值 true 和 false 一起是命题逻辑的构件。就像可以通过加法或者乘法把自然数组合起来构成算术表达式一样，可以通过一系列命题运算符(propositional operator)把原子命题组合起来，构成复合命题(compound proposition)。第一个这样的命题运算符称为否定(negation)运算符。

3.3 否定运算符

给定任意命题 p，可以考虑它的否定，记作$\neg p$。这里，$\neg p$ 也是一个命题，读作"非 p"。

例 3.5 给定命题 t，表示陈述"今天是星期二"，那么命题$\neg t$ 表示陈述"今天不是星期二"。 □

例 3.6 给定命题 r，表示陈述"里克是素食者"，那么$\neg r$ 表示陈述"里克不是素食者"。 □

练习 3.3 给出下列命题的否定命题。

1. "正在下雪。"
2. "乔恩喜欢阿里。"
3. "x 大于 y。" □

如果命题 p 的真值为 true，那么它否定命题$\neg p$ 的真值为 false；而如果命题 q 的真值为 false，那么它的否定命题$\neg q$ 的真值为 true。

例 3.7 如果命题"今天是星期二"是真的，那么命题"今天不是星期二"一定是假的。 □

在此，给出命题逻辑的第一个规则。

规则 3.1

$(\neg \text{true}) \Leftrightarrow \text{false}$

$(\neg \text{false}) \Leftrightarrow \text{true}$ □

上面的陈述$(\neg \text{true}) \Leftrightarrow \text{false}$ 的意思是命题$\neg$ true 和命题 false 是逻辑等值(logically equivalent)的，我们将在本章的后面给出这一概念的形式定义。现在，只需把陈述 $p \Leftrightarrow q$ 简单地看成是"p 和 q 具有相同的真值"就可以了[1]。

在下面的第二个规则中，可以看到一个命题的否定的否定与它原来的真值相同。

规则 3.2 对于任意命题 p，有

$(\neg\neg p) \Leftrightarrow p$ □

例 3.8

$(\neg\neg \text{true}) \Leftrightarrow \text{true}$

$(\neg\neg \text{false}) \Leftrightarrow \text{false}$ □

真值表(truth table)是表示逻辑陈述的真假性的一种方法。在一个命题的真值表中，列出它所包含的所有原子命题的真值的可能值，这样就可以计算出相对于每种组合的该命题的真值。否定命题的真值表如下所示。

p	$\neg$ p
true	**false**
false	**true**

注意，p 的可能值列在左栏，计算结果列在右栏。第一行的结果为 false，因为$\neg$ true 与false 逻辑等值。第二行的结果为 true，因为$\neg$ false 与 true 逻辑等值。在本章，我们用粗体表示真值表结果栏的值。

[1] 注意，某些教材使用 $p \equiv q$ 来表示命题 p 和命题 q 逻辑等值。

练习 3.4 计算下列命题的真值。

1. $\neg(0<1)$
2. $\neg(1+1=2)$
3. ¬("地球绕着月亮转") □

3.4 合取运算符

否定运算符是我们介绍的第一个命题逻辑运算符，它是一个一元运算符，因为它有一个自变量。即，它作用于一个命题，对这一命题取否。我们将要介绍的其他运算符都是二元运算符，它们有两个自变量。

我们介绍的第一个二元运算符是合取(conjunction)运算符，或者称为"与"。它作用于两个命题，如果这两个命题都等值于true，则返回结果true，否则返回结果false。也就是说，当且仅当命题的两个部分都等值于true时，该命题等值于true。

给定两个命题 p 和 q，p 和 q 的合取写作 $p\wedge q$。这个命题读作"p 与 q"，也可以读作"p 与 q 的合取"。

例 3.9 用 v 表示"里克是素食者"，用 c 表示"里克吃巧克力"，那么它们的合取写作 $v\wedge c$。 □

例 3.10 陈述 $0<1\wedge 1<0$ 是假的，因为 $0<1$ 和 $1<0$ 不都为真。 □

练习 3.5 计算下列命题的真值。

1. $(1<0)\wedge(2<1)$
2. $(0<1)\wedge(2<1)$
3. $(0<1)\wedge(1<2)$ □

合取运算的真值表如下所示。

p	q	$p\wedge q$
true	true	**true**
true	false	**false**
false	true	**false**
false	false	**false**

注意，当且仅当 p 和 q 都等值于true时，命题 $p\wedge q$ 等值于true。还要注意，表中枚举出了 p 和 q 的所有可能值，有四种可能的组合：p 为true且 q 为true；p 为true且 q 为false；p 为false且 q 为true；p 为false且 q 为false。

正如我们所期望的那样，可以用真值表来求包含已学过的运算符的命题的真值。

例 3.11 $p\wedge(\neg q)$ 的真值表如下所示。

p	q	$\neg q$	$p\wedge(\neg q)$
true	true	false	**false**
true	false	true	**true**
false	true	false	**false**
false	false	true	**false**

□

由上可知，$p\wedge(\neg q)$ 为true当且仅当 p 等值于true且 q 等值于false。注意，计算这个真值表需要三个阶段。首先，列出 p 和 q 的所有可能值，其次计算 $\neg q$，最后计算 p 和 $\neg q$ 的合取，并将结果写入结果栏。

也可以使用真值表来对包含真值true和真值false的命题进行推理：当然，命题true的真值为true，命题false的真值为false[○]。

例 3.12 $p\wedge$ true 的真值表如下所示。

p	true	$p\wedge$ **true**
true	true	**true**
false	true	**false**

□

练习 3.6 写出 $(\neg p)\wedge(\neg q)$ 的真值表。 □

练习 3.7 写出 $\neg(p\wedge q)$ 的真值表。 □

练习 3.8

1. 练习3.6和练习3.7的真值表有四行，$(p\wedge q)\wedge r$ 的真值表有多少行？

○ 某些教科书使用不同的名字和符号来区分真值的语法化身(即命题true)及真值的语义化身(即真值true)。例如，一些教科书用 T 表示前者，用 t 表示后者。本书不对它们进行区分。在此，true即表示总是true的命题，同时也表示两个真值中的一个。对于false也同样。

2. 一般地，如果命题有 n 个原子命题，那么这个命题的真值表有多少行？

3. 如果一个命题只由真值 true 和 false 组成，那么它的真值表有多少行？ □

对于合取，有一些运算规则。首先，任意命题 p 与它本身的合取的真值等值于 p 的真值。这被称作合取的等幂性(idempotence)。

规则 3.3 合取是幂等的。即对于任意命题 p，有

$(p \wedge p) \Leftrightarrow p$ □

第二，true 与任意命题 p 的合取的真值等值于 p 的真值。

规则 3.4 对于任意命题 p，有

$(p \wedge \text{true}) \Leftrightarrow p$ □

第三，false 与任意命题 p 的合取的真值等值于 false。

规则 3.5 对于任意命题 p，有

$(p \wedge \text{false}) \Leftrightarrow \text{false}$ □

查看 $\wedge$ 的真值表，可以发现下面的规则也成立。

规则 3.6 对于任意命题 p，有

$(p \wedge (\neg p)) \Leftrightarrow \text{false}$ □

练习 3.9 使用上述命题逻辑的规则，计算下列命题的真值。

1. $\neg\neg(\text{true} \wedge \text{false})$
2. $\text{true} \wedge (p \wedge \text{false})$
3. $(\neg(p \wedge \text{false})) \wedge \text{true}$ □

还有两个合取规则。首先，$\wedge$ 满足交换律(commutative)(即，命题的书写顺序与结果无关)，而且，它还满足结合律(associative)(即，考虑多个合取时不需要括号)。

规则 3.7 合取满足交换律。即，对于任意的命题 p 和 q，有

$(p \wedge q) \Leftrightarrow (q \wedge p)$ □

规则 3.8 合取满足结合律。对于任意的命题 p、q 和 r，有

$p \wedge (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$ □

练习 3.10 使用已经学过的命题逻辑规则，计算下列命题的真值。

1. $(\text{false} \wedge p) \wedge q$
2. $p \wedge (\text{false} \wedge q)$ □

3.5 析取运算符

第二个二元逻辑运算符是析取(disjunction)运算符，或者称为“或”。这一运算符需要两个命题，如果两个命题中至少有一个命题等值于 true，那么返回结果 true，否则返回结果 false。也就是说，当且仅当命题的至少一个部分等值于 true 时，整个命题等值于 true。

给定两个命题 p 和 q，它们的析取写作 $p \vee q$，读作“p 或 q”，也可读作“p 与 q 的析取”。

例 3.13 用 v 表示命题“里克是素食者”，用 c 表示命题“里克吃巧克力”。那么，它们的析取写作 $v \vee c$。 □

例 3.14 陈述 $(0<1) \vee (1<0)$ 是真的，因为析取的至少一个部分是真的。这里，$0<1$ 是真的。 □

练习 3.11 计算下列命题的真值。

1. $(1<0) \vee (2<1)$
2. $(0<1) \vee (2<1)$
3. $(0<1) \vee (1<2)$ □

析取的真值表如下所示。

p	q	$p \wedge q$
true	true	**true**
true	false	**true**
false	true	**true**
false	false	**false**

练习 3.12 写出 $\neg(p \vee q)$ 的真值表。 □

练习 3.13 写出 $(\neg p) \vee (\neg q)$ 的真值表。 □

注意，练习 3.6 和练习 3.12 的真值表之间的相似之处，以及练习 3.7 和练习 3.13 的真值表之间的相似之处。这些真值表如下所示。

p	q	$\neg p$	$\neg q$	$(\neg p)\wedge(\neg q)$
true	true	false	false	**false**
true	false	false	true	**false**
false	true	true	false	**false**
false	false	true	true	**true**

p	q	$p\vee q$	$\neg(p\vee q)$
true	true	true	**false**
true	false	true	**false**
false	true	true	**false**
false	false	false	**true**

p	q	$p\wedge q$	$\neg(p\wedge q)$
true	true	true	**false**
true	false	false	**true**
false	true	false	**true**
false	false	false	**true**

p	q	$\neg p$	$\neg q$	$(\neg p)\vee(\neg q)$
true	true	false	false	**false**
true	false	false	true	**ture**
false	true	true	false	**true**
false	false	true	true	**true**

前两个真值表的结果栏完全相同，后两个真值表的结果栏也完全相同。这表明，命题$\neg(p\wedge q)$逻辑等值于命题$(\neg p)\vee(\neg q)$，命题$\neg(p\vee q)$逻辑等值于命题$(\neg p)\wedge(\neg q)$。

德·摩根律[1]刻画了这些等值关系，如下所示。

规则 3.9 （德·摩根律）对于任意的命题 p 和 q，有

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

$\neg(p\vee q)\Leftrightarrow((\neg p)\wedge(\neg q))$ □

同$\wedge$一样，$\vee$也是幂等的。

规则 3.10 析取是幂等的。对于任意命题 p，有

$(p\vee p)\Leftrightarrow p$ □

任意命题与 false 的析取结果逻辑等值于最初的命题。

规则 3.11 对于任意命题 p，有

$(p\vee \text{false})\Leftrightarrow p$ □

任意命题与 true 的析取结果是 true。

规则 3.12 对于任意命题 p，有

$(p\vee \text{true})\Leftrightarrow \text{true}$ □

练习 3.14 使用已经学过的命题逻辑规则，计算下列命题的真值。

1. $(p\vee \text{false})\vee \text{true}$
2. $\neg((p\wedge \text{false})\vee \text{true})$ □

同合取一样，析取既满足结合律，又满足交换律。

规则 3.13 析取满足结合律。对于任意的命题 p、q 和 r，有

$p\vee(q\vee r)\Leftrightarrow(p\vee q)\vee r$ □

规则 3.14 析取满足交换律。对于任意的命题 p 和 q，有

$p\vee q\Leftrightarrow q\vee p$ □

练习 3.15 使用已经学过的命题逻辑规则，计算下列命题的真值。

1. $(\neg(\text{true}\vee p))\wedge(p\vee \text{false})$
2. $p\vee(\text{true}\vee q)$ □

从析取的真值表可以看到，给定命题 p，$p\vee(\neg p)$总等值于 true：当 p 为真时，$p\vee(\neg p)$为真，因为析取的左边为真；当 p 为假时，$p\vee(\neg p)$为真，因为析取的右边为真。这就是众所周知的排中律(law of the excluded middle)，其形式表述如下。

规则 3.15 对于任意命题 p，有

$((\neg p)\vee p)\Leftrightarrow \text{true}$ □

最后两个关于析取和合取的规则表明析取对合取满足分配律，反之，合取对析取也满足分配律。

规则 3.16 析取对合取满足分配律。对于任意的命题 p、q 和 r，有

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

$(p\wedge q)\vee r\Leftrightarrow(p\vee r)\wedge(q\vee r)$ □

规则 3.17 合取对析取满足分配律。对于任意的命题 p、q 和 r，有

---

[1] 以 Augustus de Morgan(1806—1871)的名字命名。有兴趣的读者可以从[Mer90]看到德·摩根在这一领域的工作的记述。

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

$(p \vee q) \wedge r \Leftrightarrow (p \wedge r) \vee (q \wedge r)$ □

这些规则的形式在开始时可能难于理解。如果是这样的话，可以联想一下算术中乘法对于加法的分配律。如，式子

$3 \times (2+5)$

可以写成

$(3 \times 2)+(3 \times 5)$

它们的结果都是 21。合取对于析取的分配律以及析取对于合取的分配律的形式与这一我们熟悉的分配律一样。

例 3.15 命题 true∨(true∧false)逻辑等值于命题(true∨true)∧(true∨false)。

□

练习 3.16

1. 使用已经学过的命题逻辑规则，证明 $p \wedge ((\neg p) \vee q)$ 等值于 $p \wedge q$。

2. 使用已经学过的命题逻辑规则，证明 $(p \wedge q) \vee (\neg p)$ 等值于 $q \vee (\neg p)$。 □

3.6 蕴含运算符

蕴含是我们介绍的第三个逻辑运算符。对于大多数学习离散数学的学生来说，这是最不直观、最难理解的运算符。蕴含写作⇒，命题 $p \Rightarrow q$ 读作"p 蕴含 q"或者"如果 p，则 q"。

例如，陈述"如果下午下雨，邓肯将待在房间里"可以表示成 $r \Rightarrow s$，(r 代表"下午下雨"，s 代表"邓肯待在房间里")。如果把它看作一个约定，那么只在一种情况下可以认为我们受骗了：下午真的下雨了，而且邓肯没有待在房间里。如果下午下了雨而且邓肯待在了房间里，那么约定没有被破坏，我们会感到满意。另外，如果没有下雨，那么我们无需担心邓肯下午做了什么，因为这显然超出了约定的范围。因此，$r \Rightarrow s$ 逻辑等值于 false(即约定被破坏)，当且仅当 r 等值于 true 且 s 等值于 false；否则它逻辑等值于 true。

练习 3.17 假设 e 表示"爱米丽高兴"且 r 表示"雷切尔高兴"，将下列自然语言的陈述转换成逻辑命题。

1. "如果爱米丽高兴，那么雷切尔高兴。"

2. "如果雷切尔高兴，那么爱米丽高兴。"

3. "爱米丽高兴仅当雷切尔高兴。"

4. "不存在如果雷切尔则爱米丽不高兴这样的情况。" □

练习 3.18 判定下列陈述的真假性。

1. "如果玛丽莲·梦露是男人，那么大象会飞。"

2. "如果 1+1=2，那么马德里是西班牙的首都。"

3. "如果玛丽莲·梦露是男人，那么 1+1=2。"

4. "如果马德里是西班牙的首都，那么大象会飞。" □

蕴含的真值表如下所示。

p	q	$p \Rightarrow q$
true	true	**true**
true	false	**false**
false	true	**true**
false	false	**true**

注意，如果 p 等值于 false，那么 $p \Rightarrow q$ 的值总是等值于 true，与 q 的值无关。另一方面，如果 p 等值于 true，那么 $p \Rightarrow q$ 的值总是等值于 q 的值。

练习 3.19 写出 $(p \wedge q) \Rightarrow (p \vee q)$ 的真值表。 □

练习 3.20 写出 $(p \vee q) \Rightarrow (p \wedge q)$ 的真值表。 □

对于为什么如此定义蕴含运算符的一种解释是，考虑 true 和 false 间的"强度顺序"，其中 false 强于 true。而且，对于任意的命题，false 至少和它一样强(false 是最强的命题)，而且 true 至少和它一样弱(true 是最弱的命题)。运用这一手法，命题 $p \Rightarrow q$

等值于 false，当且仅当 p 弱于 q。

因为任意的命题 p 总是与其自身一样强，所以，false⇒false 和 true⇒true 都等值于 true。另外，因为 false 强于 true，所以 false⇒true 等值于 true。因为蕴含式 $p \Rightarrow q$ 等值于 false 仅当前项(antecedent，出现在蕴含运算符⇒之前的命题)弱于后项(consequent，出现在蕴含运算符⇒之后的命题)，所以 true⇒false 逻辑等值于 false。

例 3.16 可以用上面的手法确定命题 true⇒(p⇒(false⇒p))的真假性。

首先，对于任意的命题，false 至少与它一样强，因此

true⇒(p⇒(false⇒p))

逻辑等值于

true⇒(p⇒true)

因为对于任意命题，true 至少与它一样弱，所以上面的命题等值于

true⇒true

最后，由于 true 不比自身弱，所以上面的命题等值于 true。 □

蕴含不具有 ∧ 和 ∨ 的那些迷人的性质；蕴含既不满足交换律，也不满足结合律，同时也不是幂等的。但是，有一种非常有用的等值关系，可以用它来把蕴含转换成与其逻辑等值的析取。

规则 3.18 对任意的命题 p 和 q，有

$(p \Rightarrow q) \Leftrightarrow ((\neg p) \vee q)$ □

例 3.17 蕴含"如果下午下雨，那么邓肯将待在房间里"逻辑等值于析取"下午不下雨或邓肯将待在房间里"。 □

练习 3.21 使用已经学过的逻辑命题规则，计算下列命题的真值。

1. $p \Rightarrow p$
2. $(p \wedge q) \Rightarrow q$ □

练习 3.22 证明⇒不满足以下性质。

1. 结合律。
2. 交换律。
3. 幂等律。 □

3.7 等值运算符

我们介绍的最后的逻辑运算符是等值(equivalence)。本章已经使用了 $p \Leftrightarrow q$ 来表示两个命题逻辑等值。例如，陈述"70 分以上等值于优秀"可以表示成命题 $s \Leftrightarrow d$，其中 s 表示"70 分以上"，d 表示"优秀"。

本质上，对于两个命题 p 和 q，如果它们有相同的真值，那么它们逻辑等值。即，恰好 q 为 true 时 p 为 true。我们用 $p \Leftrightarrow q$ 表示陈述"p 逻辑等值于 q"或者"p 当且仅当 q"。

例 3.18 命题 $0 \leqslant 1 \Leftrightarrow 1 \geqslant 1$ 是真的，而命题 $0 \leqslant 1 \Leftrightarrow 1 > 1$ 是假的。 □

例 3.19 设 v 表示命题"里克是素食者"，c 表示命题"里克吃巧克力"，那么命题"里克是素食者当且仅当里克吃巧克力"写作 $v \Leftrightarrow c$。 □

练习 3.23 确定下列陈述的真假性。

1. "玛丽莲·梦露是男人当且仅当大象会飞。"
2. "1+1=2 当且仅当马德里是西班牙的首都。"
3. "玛丽莲·梦露是男人当且仅当 1+1=2。"
4. "马德里是西班牙的首都当且仅当大象会飞。"

等值的真值表如下所示。 □

p	q	**p⇒q**
true	true	**true**
true	false	**false**
false	true	**false**
false	false	**true**

练习 3.24 写出命题$(p \Rightarrow q) \Leftrightarrow (q \Rightarrow p)$的真值表。 □

等值运算符既满足结合律，也满足交换律。

规则 3.19 等值运算符满足结合律。对于任意的命题 p、q 和 r，有

$((p \Leftrightarrow q) \Leftrightarrow r) \Leftrightarrow (p \Leftrightarrow (q \Leftrightarrow r))$ □

规则 3.20 等值运算符满足交换律。对于任意的命题 p 和 q，有

$(p \Leftrightarrow q) \Leftrightarrow (q \Leftrightarrow p)$ □

另外，任意命题都与其自身等值。

规则 3.21 对于任意命题 p，有

$(p \Leftrightarrow p) \Leftrightarrow \text{true}$ □

相反，任意命题都不与它的否定等值。

规则 3.22 对于任意命题 p，有

$(p \Leftrightarrow (\neg p)) \Leftrightarrow \text{false}$ □

最后，命题 p 等值于命题 q 等同于 p 蕴含 q 且 q 蕴含 p。

规则 3.23 对于任意的命题 p 和 q，有

$(p \Leftrightarrow q) \Leftrightarrow ((p \Rightarrow q) \wedge (q \Rightarrow p))$ □

例 3.20 例如，命题"克莱尔获得优秀当且仅当克莱尔得 70 分以上"逻辑等值于"如果克莱尔获得优秀，那么克莱尔得 70 分以上；并且，如果克莱尔得 70 分以上，那么克莱尔获得优秀"。 □

练习 3.25 使用已经学过的命题逻辑规则，计算下列命题的真值。

1. $(p \Leftrightarrow \text{true}) \Leftrightarrow p$
2. $(p \Leftrightarrow \text{false}) \Leftrightarrow \neg p$ □

练习 3.26 证明$\Leftrightarrow$不是幂等的。 □

3.8 运算的优先级

正如我们在本章已经看到的，可以使用括号来消除给定命题的含义上的模糊性。例如，命题$\neg(p \wedge q)$与$(\neg p) \wedge q$有不同的含义。然而，如果简单地写$\neg p \wedge q$，那么要表达的意思就可能模糊不清了。使用括号能够使命题清晰化。

然而，有时，尤其是当处理复杂命题的时候，像这样来使用括号常常十分令人讨厌。例如，命题$\neg p \vee \neg q \vee \neg r$的含义已经相当清楚，但是，如果必须加上括号，那么就必须写成$(\neg p) \vee ((\neg q) \vee (\neg r))$或$((\neg p) \vee (\neg q)) \vee (\neg r)$（当然，它们是逻辑等值的，在这种情况下括号纯粹是多余的）。后面这种括号形式的命题与原来无括号形式的命题有相同的逻辑含义，但是括号形式的命题看起来比较笨拙。

为使清晰与优雅兼而有之，我们给命题运算符规定优先级顺序：$\neg$的优先级最高（因此绑定最紧密），接着依次是$\wedge$、$\vee$、$\Rightarrow$和$\Leftrightarrow$，$\Leftrightarrow$的优先级最低（因此绑定最松散）。当一个运算符的优先级高于另一个运算符的优先级时，优先级高的运算符在优先级低的运算符"抓住"与其最近的命题之前"抓住"与它最近的命题。

例 3.21 如果我们写$\neg p \wedge q$，那么它与$(\neg p) \wedge q$的含义相同，而与$\neg(p \wedge q)$的含义不同，因为$\neg$比$\wedge$绑定得更紧密。 □

例 3.22 命题$p \vee q \Rightarrow r$与$(p \vee q) \Rightarrow r$的含义相同，而与$p \vee (q \Rightarrow r)$的含义不同，因为$\vee$比$\Rightarrow$绑定得更紧密。 □

练习 3.27 写出下列命题的加括号形式。

1. $\neg p \vee q$
2. $p \vee q \wedge r$
3. $p \wedge q \Rightarrow r$
4. $\neg p \Leftrightarrow \neg q \wedge r \Rightarrow s$ □

3.9 重言式、矛盾式和不定式

对于一个命题，如果无论它所包含的原子命题的值如何，它总是逻辑等值于 true 的话，那么这个命题称为重言式（tautology）。重言式的例子包括$p \vee \neg p$、$p \Rightarrow p$和$(p \wedge q) \vee (p \wedge \neg q) \vee (\neg p \wedge q) \vee (\neg p \wedge \neg q)$。无论各个原子命题的值是什么，这些命题的值总为 true。

假设一个给定的命题包含为数不多的原子命题，判定该命题是否为重言式的一个简单方法是对每个原子命题带入真值。如果对于原子命题的值的所有组合该命题的真值都为 true，那么就可以断定这个命题是重言式。相反，如果可以找到原子命题的值的一

种组合使得命题的真值为 false，那么就可以断定这个命题不是重言式。

例 3.23 给定命题 $p \vee \neg p$，如果把 true 带入到 p，那么该命题等值于 true。另外，如果把 false 带入到 p，那么该命题仍等值于 true。因此，可以断定 $p \vee \neg p$ 是重言式。 □

例 3.24 考虑命题 $(p \vee \neg p) \Rightarrow q$。如果假设原子命题 q 的值为 false，那么，因为 $p \vee \neg p$ 等值于 true，所以无论 p 的值是什么，蕴含式的值都为 false。因此，可以断定上述命题不是重言式。 □

练习 3.28 下列命题中哪些是重言式？

1. $p \vee (\neg p \wedge q)$
2. $p \Rightarrow (q \Rightarrow p)$
3. $(p \Rightarrow q) \Rightarrow (q \Rightarrow p)$
4. $(p \Rightarrow q) \Leftrightarrow \neg(p \wedge \neg q)$ □

总为 false 的命题称为矛盾式(contradiction)。同样，可以通过对原子命题带入真值来决定给定命题是否是矛盾式。

例 3.25 考虑命题 $\neg(p \Rightarrow p)$。如果 p 为 true，那么该命题等值于 false。另外，如果 p 为 false，那么该命题也等值于 false。因此，这个命题是矛盾式。 □

练习 3.29 下列命题中哪些是矛盾式？

1. $(p \wedge q) \wedge (\neg(p \vee q))$
2. $(p \vee \neg p) \Rightarrow (q \wedge \neg q)$
3. $(q \wedge \neg q) \Rightarrow (p \vee \neg p)$ □

如果一个命题既不是重言式也不是矛盾式，那么称它为不定式(contingency)。

例 3.26 考虑命题 $p \Rightarrow q$。如果 p 为 true 且 q 为 false，那么 $p \Rightarrow q$ 为 false。相反，如果 p 和 q 同为 true，那么 $p \Rightarrow q$ 为 true。因此，$p \Rightarrow q$ 为不定式。 □

练习 3.30 下列命题中哪些是不定式？

1. $p \Rightarrow (\neg p \Rightarrow q)$
2. $(p \vee q) \Rightarrow (p \wedge q)$ □

我们已经学习了判断一个命题是重言式(即，逻辑等值于 true)、矛盾式(即，逻辑等值于 false)还是不定式(既不逻辑等值于 true，也不逻辑等值于 false)的一种方法。但是，这种方法只适用于包含较少数目原子命题的相对简单的命题：虽然对于包含许多原子命题的命题进行带入是可行的，但是，这样做非常耗时。判断一个命题是否为重言式(矛盾式或不定式)的更常用方法是使用真值表、自然演绎(natural deduction)和等值推理(equational reasoning)。下面逐一介绍这些方法。

3.10 真值表

根据练习 3.28 的结果，命题 $p \Rightarrow (q \Rightarrow p)$ 是重言式。这一事实可以通过简单地代入 p 和 q 的真值来确认。如果愿意的话，也可以通过真值表来证明。

例 3.27 $p \Rightarrow (q \Rightarrow p)$ 的真值表如下所示。

p	q	$q \Rightarrow p$	$p \Rightarrow (q \Rightarrow p)$
true	true	true	**true**
true	false	true	**true**
false	true	false	**true**
false	false	true	**true**

□

观察最后一栏，可以发现它的每一项都为 true；当出现这种情况时，就可以推断这个命题是重言式。相反，如果所给真值表的最后一栏中的每一项都为 false，那么这个命题就是矛盾式。最后，如果最后一栏中至少有一项为 true，且至少有一项为 false，那么这个命题就是不定式。

练习 3.31 使用真值表判断下列命题是重言式、矛盾式还是不定式。

1. $(p \wedge q) \vee (\neg(p \wedge q))$
2. $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$
3. $(\neg p \Rightarrow q) \Rightarrow (p \Rightarrow \neg q)$
4. $(p \Rightarrow q) \Rightarrow (\neg p \Rightarrow \neg q)$
5. $\neg(p \Rightarrow p) \wedge (\neg p \Rightarrow \neg p)$ □

使用真值表同赋值一样有其自身的缺

点。正如前面看到的那样，一个命题的真值表的大小与该命题中所含原子命题的数量有直接的关系。如果命题中仅含有一个原子命题，那么这个命题的真值表有两行；如果命题含有两个原子命题，那么其真值表含有四行；如果命题含有三个原子命题，那么其真值表含有八行；以此类推。事实上，如果手工计算真值表的话，三个原子命题已经是上限了；多于三个原子命题意味着用于构造真值表的时间负荷将超过这种方法所带来的便利。同赋值的方法一样，这种做法是可行的，但是需要大量的时间才能完成。简而言之，这种做法适合由计算机进行计算，而不适合手工计算。因此，需要一些更加普遍的方法来判断给定命题是否为重言式，这些技术中最常用的是等值推理。

3.11 等值推理

在前面，我们利用一个命题取代另一个命题来决定命题的真假性。在这一过程中，每一步都运用了命题逻辑的规则。因为我们在运用这些规则时非常严谨，所以这些表面看来非形式的过程实际上是相当形式化的(formal)；我们将这种分析的方法称为等值推理(equational reasoning)。

以练习 3.21.1 的解答为例。这里，如下计算 $p \Rightarrow p$ 的真值。

根据规则 3.18，$p \Rightarrow p$ 等值于 $(\neg p) \vee p$。根据规则 3.15，$(\neg p) \vee p$ 等值于 true。

确立 $p \Rightarrow p$ 等值于 true 的一个更加形式化(且清晰)的方法如下所示。

$p \Rightarrow p$

$\Leftrightarrow (\neg p) \vee p$ [规则 3.18]

$\Leftrightarrow \text{true}$ [规则 3.15]

每一步，我们都用一个逻辑等值的命题来取代前一个命题；取代依照命题逻辑的规则。现在，我们可以更加确信这个过程，因为它的每一步取代都是十分显而易见的。

例 3.28 证明 $p \Rightarrow (p \Rightarrow q)$ 逻辑等值于 $p \Rightarrow q$。

$(p \Rightarrow (p \Rightarrow q)) \Leftrightarrow (p \Rightarrow q)$

$\Leftrightarrow (\neg p \vee (p \Rightarrow q)) \Leftrightarrow (p \Rightarrow q)$ [规则 3.18]

$\Leftrightarrow (\neg p \vee (\neg p \vee q)) \Leftrightarrow (p \Rightarrow q)$ [规则 3.18]

$\Leftrightarrow ((\neg p \vee \neg p) \vee q) \Leftrightarrow (p \Rightarrow q)$ [规则 3.13]

$\Leftrightarrow (\neg p \vee q) \Leftrightarrow (p \Rightarrow q)$ [规则 3.10]

$\Leftrightarrow (p \Rightarrow q) \Leftrightarrow (p \Rightarrow q)$ [规则 3.18]

$\Leftrightarrow \text{true}$ [规则 3.21]

□

练习 3.32 练习 3.25 曾要求计算下列命题的真值。

1. $(p \Leftrightarrow \text{true}) \Leftrightarrow p$
2. $(p \Leftrightarrow \text{false}) \Leftrightarrow \neg p$

运用上面所述更加形式化的方法重新写出证明。 □

当通过等值推理来证明两个命题等值时，有两种方法。上面使用的是第一种方法，这种方法考虑整个命题并确立它逻辑等值于 true。还有一种方法，这种方法从等值式的左边出发，通过等值推理确立它与等值式的右边逻辑等值。运用这种方法，练习 3.32 的命题 1 的一个解如下所示。

$p \Leftrightarrow \text{true}$

$\Leftrightarrow (p \Rightarrow \text{true}) \wedge (\text{true} \Rightarrow p)$ [规则 3.23]

$\Leftrightarrow ((\neg p) \vee \text{true}) \wedge (\text{true} \Rightarrow p)$ [规则 3.18]

$\Leftrightarrow ((\neg p) \vee \text{true}) \wedge ((\neg \text{true}) \vee p)$ [规则 3.18]

$\Leftrightarrow \text{true} \wedge ((\neg \text{true}) \vee p)$ [规则 3.12]

$\Leftrightarrow \text{true} \wedge (\text{false} \vee p)$ [规则 3.1]

$\Leftrightarrow \text{true} \wedge (p \vee \text{false})$ [规则 3.14]

$\Leftrightarrow \text{true} \wedge p$ [规则 3.11]

$\Leftrightarrow p \wedge \text{true}$ [规则 3.7]

$\Leftrightarrow p$ [规则 3.4]

练习 3.33 运用等值推理证明下列等值式。

1. $((p \wedge \neg q) \Rightarrow q) \Leftrightarrow (\neg p \vee q)$

2. $(p \Rightarrow q) \wedge p \Leftrightarrow q \wedge p$ □

练习 3.34 运用等值推理证明下列命题等值于 true。

1. $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$

2. $(p \Leftrightarrow \text{true}) \Rightarrow p$ □

练习 3.35 运用等值推理简化下列命题。

1. $(p \Leftrightarrow q) \Rightarrow (q \Leftrightarrow p)$

2. $(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$ □

3.12 自然演绎

在本章的前面，我们把 $p \Rightarrow p$ 和 $p \vee \neg p$ 这样的陈述称为重言式：无论原子命题的真值是什么，这些命题总为 true。在另一层意义上，也称这样的陈述为定理(theorem)：一旦证明了一个命题逻辑等值于 true，那么就可以使用这一命题来证明命题逻辑的其他重言式或定理。

自然演绎(natural deduction)或证明树(proof tree)可用来证明命题逻辑定理的正确性。与等值推理一样，可以通过使用特定的规则操作命题来证明这些定理。同样，与等值推理一样，它的过程也是机械化的，如果希望的话，可以自动进行。

定理的推导由一组合理的规则来规范，其形式如下。

$$\frac{\text{前提}}{\text{结论}}[\text{规则名}]$$

其中，规则名是对规则的称呼，一条规则允许在一个或多个前提(premise)下得到一条或多条结论(conclusion)。

虽然证明树(或自然演绎)是证明命题逻辑定理的一种一般方法，但是，通常不同的系统使用不同的规则。我们所提供的系统与[WD96]所给的形式描述技术 Z 相关。需要注意的是，尽管我们只考虑一个特殊的系统，但是，该系统中的大部分规则对其他演绎系统也适用。另外，有兴趣的读者可以参照[Hun84]，它描述了自然演绎(虽然不是我们提到的特殊系统)的方法和策略。

第一个自然演绎规则是假言推理(modus ponens)，或称蕴含消去规则(⇒elim)。蕴含消去规则如下所示。

$$\frac{p \Rightarrow q \quad p}{q}[\text{蕴含消去规则}]$$

这一规则表明，如果已经证明从 p 能够得到 q，而且也证明了 p 为 true，那么就可以断定 q 为 true。

例 3.29 下面是一个使用蕴含消去规则的例子。

$$\frac{\text{“如果下雨，那么我会被淋湿”} \quad \text{“下雨”}}{\text{“我会被淋湿”}}$$

[蕴含消去规则] □

例 3.30 下面是另一个使用蕴含消去规则的例子。

$$\frac{\text{“如果我玩火，那么我必自焚”} \quad \text{“我玩火”}}{\text{“我必自焚”}}$$

[蕴含消去规则] □

如同其他二元运算符一样，蕴含(⇒)除了有一个“消去”规则外，还有一个“引入”规则。这一规则的一般形式如下所示。

$$\begin{array}{c} [p]_1 \\ \vdots \\ \dfrac{q}{p \Rightarrow q}[\text{蕴含引入规则}_1] \end{array}$$

这一规则表明，如果能够通过演绎系统的规则从命题 p 得到 q 的话，那么就可以得到从 p 可以推导出 q 的结论，也就是说，$p \Rightarrow q$ 为 true。注意，在推导出 q 的过程中，如果需要，可以多次使用 p。

$[p]_1$ 表示 p 是一个被释放了(discharged)的假设(assumption)[○]；这一假设由标签 1

[○] 被释放了的假设指的是某个子演绎的假设，但对整个演绎不再是假设。——译者注

标识。证明树的根是已经证明的定理：当我们构建证明树时，从要证明的定理开始，逐步向下推到叶节点。证明树的叶子或是已经在其他地方证明了的其他定理，或者是一个假设。定理不应该含有没有释放的假设。

例 3.31 可以使用前面两个规则证明 $(p\Rightarrow q)\Rightarrow(p\Rightarrow q)$ 是命题逻辑的定理。这里，$p\Rightarrow q$ 和 p 都是在应用蕴含引入规则后被释放了的假设。

$$\dfrac{\dfrac{\lceil p\Rightarrow q\rceil_1 \quad \lceil p\rceil_2}{q}[\text{蕴含消去规则}]}{p\Rightarrow q}[\text{蕴含引入规则}_2]}{(p\Rightarrow q)\Rightarrow(p\Rightarrow q)}[\text{蕴含引入规则}_1]$$

这棵树的构造如下所示。

我们从要证明的定理 $(p\Rightarrow q)\Rightarrow(p\Rightarrow q)$ 出发。由于最外层的逻辑运算符是蕴含运算符，所以可以想象使用蕴含引入规则来得到这一结果。

$$\dfrac{p\Rightarrow q}{(p\Rightarrow q)\Rightarrow(p\Rightarrow q)}[\text{蕴含引入规则}_1]$$

由蕴含引入规则的机制可知，$p\Rightarrow q$ 是第一个假设，为了得到(上面树中的)$p\Rightarrow q$，可以反复运用这一假设。

向上移动树，现在来看 $p\Rightarrow q$。这一次最外层的运算符还是蕴含运算符，因此仍然可以使用蕴含引入规则来得到这一结果。

$$\dfrac{\dfrac{q}{p\Rightarrow q}[\text{蕴含引入规则}_2]}{(p\Rightarrow q)\Rightarrow(p\Rightarrow q)}[\text{蕴含引入规则}_1]$$

同样，由蕴含引入规则的机制可知，p 是第二个假设，为了得到 q，可以反复使用这一假设。

现在，必须从 p 和 $p\Rightarrow q$ 证明 q。在这里，显然有一个规则可以用来完成这一证明：蕴含消去规则。这样，我们就得到了完整的证明树。

$$\dfrac{\dfrac{\dfrac{\lceil p\Rightarrow q\rceil_1 \quad \lceil p\rceil_2}{q}[\text{蕴含消去规则}]}{p\Rightarrow q}[\text{蕴含引入规则}_2]}{(p\Rightarrow q)\Rightarrow(p\Rightarrow q)}[\text{蕴含引入规则}_1]$$

我们得到的是一棵不含没被释放假设的树，因此，可以得出

$$(p\Rightarrow q)\Rightarrow(p\Rightarrow q)$$

是命题逻辑的定理这一结论。 □

在上面，下标 1 表示使用蕴含引入规则得出结论 $(p\Rightarrow q)\Rightarrow(p\Rightarrow q)$ 是基于 $p\Rightarrow q$ 的假设，而下标 2 表示使用蕴含引入规则得出结论 $p\Rightarrow q$ 是基于 p 的假设[3]。

回想一下，如果命题 q 逻辑等值于 true，那么对于任意命题 p，蕴含 $p\Rightarrow q$ 也逻辑等值于 true。与此相似，在自然演绎的范畴内，如果通过证明树证明了命题 p 为真，那么对任意的命题 q，$q\Rightarrow p$ 也为真。

例 3.32 我们已经看到"猪能飞 $\Rightarrow 1+1=2$"逻辑等值于 true，因为 $(p\Rightarrow \text{true})\Leftrightarrow \text{true}$ 是命题逻辑的定理。可以如下证明它是命题逻辑的定理。

$$\dfrac{\dfrac{}{1+1=2}[\text{数学定理}]}{\text{猪能飞}\Rightarrow 1+1=2}[\text{蕴含引入规则}] \quad \square$$

可以肯定 $1+1=2$ 是一个数学定理。因此，无需前提或者假设去证明它的有效性。虽然在 $1+1=2$ 的证明中没有使用陈述"猪能飞"，但是，由于对于任意命题 p，$p\Rightarrow \text{true}$ 都逻辑等值于 true，因此可以在此使用蕴含引入规则。我们称这样的假设为真空(vacuous)假设，在这种情况下，不需要为相关规则赋下标。因此，尽管在证明 $1+1=2$ 时假设了"猪能飞"，但是在任意步骤中都不需要使用这一假设。

下一个规则称为真引入(true introduction)规则，该规则简单地说明 true 是命题逻辑的定理。

$$\dfrac{}{\text{true}}[\text{真引入规则}]$$

[3] 原书中的下标 1 和下标 2 的解释是颠倒的。 译者注

练习 3.36 使用学过的规则证明 false ⇒true 是命题逻辑的定理。 □

下一个规则是关于合取运算符的规则。如果我们证明了 $p \wedge q$ 为真，那么就可得到 p 为真。

$$\frac{p \wedge q}{p}[\text{合取消去规则 1}]$$

同样，如果我们证明了 $p \wedge q$ 为真，那么就可得到 q 为真。

$$\frac{p \wedge q}{p}[\text{合取消去规则 2}]$$

上式中的 1 和 2 与假设无关，它们是规则名称的组成部分，表明哪一个合取构成了结论。

相反，如果我们证明了 p 和 q 都为真，那么就可得到 $p \wedge q$ 为真。

$$\frac{p \quad q}{p \wedge q}[\text{合取引入规则}]$$

例 3.33 用上面的规则可以证明下列命题

$$((p \wedge q) \wedge r) \Rightarrow (p \wedge (q \wedge r))$$

是命题逻辑的定理。证明如下。

$$\dfrac{\dfrac{\dfrac{\dfrac{[(p \wedge q) \wedge r]_1}{p \wedge q}[\wedge\text{elim1}]^{①}}{p}[\wedge\text{elim1}]^{①} \quad \dfrac{\dfrac{\dfrac{[(p \wedge q) \wedge r]_1}{p \wedge q}[\wedge\text{elim1}]^{①}}{q}[\wedge\text{elim2}]^{②} \quad \dfrac{[(p \wedge q) \wedge r]_1}{r}[\wedge\text{elim2}]^{②}}{q \wedge r}[\wedge\text{intro}]^{③}}{p \wedge (q \wedge r)}[\wedge\text{intro}]^{③}}{((p \wedge q) \wedge r) \Rightarrow (p \wedge (q \wedge r))}[\Rightarrow\text{intro}_1]^{④}$$

注：① 合取消去规则 1；② 合取消去规则 2；③ 合取引入规则；④ 蕴含引入规则$_1$。

□

练习 3.37 使用合取引入规则和合取消去规则，证明

$$(p \wedge q) \Rightarrow (q \wedge p)$$

是命题逻辑的定理。 □

析取引入规则也是很直观的：如果我们证明了 p，那么就可得到 $p \vee q$ 为真。

$$\frac{p}{p \vee q}[\text{析取引入规则 1}]$$

同样，如果我们证明了 q，那么也可得到 $p \vee q$ 为真。

$$\frac{p}{p \vee q}[\text{析取引入规则 2}]$$

练习 3.38 使用已经学过的自然演绎规则证明下列命题是命题逻辑的定理。

1. $p \wedge q \Rightarrow p \vee q$
2. $\text{false} \Rightarrow (\text{true} \vee \text{false})$ □

析取消去规则虽然很直观，但乍看起来却非常复杂。

首先，假设通过自然演绎我们已经从 p 得出了 r，而且也从 q 得出了 r。其次，假设我们已经证明了 $p \vee q$ 成立。如果我们知道 p 为真或者 q 为真，而且知道从 p 和 q 都可以证明 r 为真，那么，就可以得出 r 为真。

这个规则的形式如下所示。

$$\frac{\begin{matrix}[p]_1 & [q]_1 & \\ \vdots & \vdots & \\ r & r & p \vee q\end{matrix}}{r}[\text{析取消去规则}_1]$$

例 3.34 假设我们已经证明意大利或者西班牙将在某足球赛中获胜。另外，根据地理学，我们也知道如果意大利获胜，那么欧洲将是获胜者，还是根据地理学，我们知道如果西班牙获胜，那么欧洲将是获胜者。所以，可以得出某个欧洲队将获胜。形式化的推导过程如下所示。

$$\dfrac{\dfrac{\lceil \text{意大利获胜} \rceil_1}{\text{欧洲获胜}}[\text{地理学}] \quad \dfrac{\lceil \text{西班牙获胜} \rceil_1}{\text{欧洲获胜}}[\text{地理学}] \quad \dfrac{}{\text{意大利获胜} \vee \text{西班牙获胜}}[\text{定理}]}{\text{欧洲获胜}}[\text{析取消去规则}_1]$$

□

练习 3.39 使用已经学过的自然演绎规则，证明

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

是命题逻辑的定理。 □

等值运算符的演绎规则相对来说比较直观。首先，如果我们已经证明了 $p \Leftrightarrow q$ 为真，那么 $p \Rightarrow q$ 和 $q \Rightarrow p$ 都为真。

$$\dfrac{p \Leftrightarrow q}{p \Rightarrow q}[\text{等值消去规则 1}]$$

$$\dfrac{p \Leftrightarrow q}{q \Rightarrow p}[\text{等值消去规则 2}]$$

另外，如果我们已经证明了 $p \Rightarrow q$ 和 $q \Rightarrow p$ 都为真，那么 $p \Leftrightarrow q$ 为真。

$$\dfrac{p \Rightarrow q \quad q \Rightarrow p}{p \Leftrightarrow q}[\text{等值引入规则}]$$

练习 3.40 使用已经学过的自然演绎规则，证明

$$(p \Leftrightarrow q) \Rightarrow ((p \Rightarrow q) \wedge (q \Rightarrow p))$$

是命题逻辑的定理。 □

练习 3.41 拓展练习 3.37 的结果，证明

$$(p \wedge q) \Leftrightarrow (q \wedge p)$$

是命题逻辑的定理。 □

由于证明树与要证明的定理（总为 true 的陈述）相关，如果在证明树中得到了 false，就可以知道我们做了错误的假设，也就是说，在推导中“走错了路”。这个观点可由下面的规则来刻画：如果我们从 p 证明出了 false，那么 p 一定等值于 false，即，$\neg p$ 一定等值于 true。

$$\begin{array}{c} p \\ \vdots \\ \dfrac{\text{false}}{\neg p}[\text{否定引入规则}] \end{array}$$

同样，如果我们从 $\neg p$ 证明出 false，那么 $\neg p$ 一定为 false，即，p 一定为 true。

$$\begin{array}{c} \neg p \\ \vdots \\ \dfrac{\text{false}}{p}[\text{否定消去规则}] \end{array}$$

最后，p 和 $\neg p$ 不能同时为真。如果我们在某处同时证明了 p 和 $\neg p$，那么就必须得到 false 的结果。

$$\dfrac{p \quad \neg p}{\text{false}}[\text{假引入规则}]$$

练习 3.42 使用自然演绎规则，证明下列命题是命题逻辑的定理。

1. $(p \wedge \neg p) \Rightarrow \neg(p \wedge \neg p)$
2. $\neg(p \vee \neg p) \Rightarrow (p \vee \neg p)$ □

3.13 附加练习

练习 3.43 假设 p 代表陈述“乔恩高兴”，q 代表“史蒂夫痛苦”，写出下列命题所表示的自然语言陈述。

1. $\neg p$
2. $p \wedge q$
3. $p \vee \neg q$
4. $p \Rightarrow q$
5. $p \Leftrightarrow q$

练习 3.44 假设 p 代表“乔恩高兴”，q 代表“史蒂夫痛苦”，写出表示下列自然语言陈述的命题形式。

1. “史蒂夫不痛苦。”
2. “史蒂夫不痛苦而且乔恩不高兴。”
3. “史蒂夫痛苦或者乔恩不高兴。”
4. “如果史蒂夫痛苦，那么乔恩高兴。”
5. “乔恩不高兴当且仅当史蒂夫痛苦。”

练习 3.45 假设 p 代表“乔恩高兴”，q 代表“史蒂夫痛苦”，r 代表“阿里生气”，写

出下列命题所表示的自然语言陈述。

1. $(p \wedge q) \Rightarrow r$

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

3. $p \Leftrightarrow (q \wedge r)$

练习 3.46 假设 p 代表"乔恩高兴"，q 代表"史蒂夫痛苦"，r 代表"阿里生气"，写出表示下列自然语言陈述的命题形式。

1. "乔恩高兴，而且并非史蒂夫痛苦且阿里生气。"

2. "如果阿里生气，那么乔恩高兴或者史蒂夫痛苦。"

3. "史蒂夫痛苦当且仅当乔恩高兴且阿里不生气。"

练习 3.47 写出下列各命题的真值表。

1. $p \vee \neg q$

2. $\neg p \wedge q$

3. $p \Rightarrow (p \wedge q)$

4. $\neg p \Leftrightarrow q$

练习 3.48 使用等值推理证明 $p \Rightarrow (p \wedge q)$ 等值于 $p \Rightarrow q$。

练习 3.49 使用等值推理证明

$(\neg p \Rightarrow p) \Leftrightarrow p$

为重言式。

练习 3.50 使用证明树证明 $p \Rightarrow (q \Rightarrow p)$ 是命题逻辑的定理。

练习 3.51 使用证明树证明

$(p \wedge \neg p) \Rightarrow (\text{true} \Rightarrow \text{false})$

是命题逻辑的定理。

练习 3.52 合取式是由合取运算符连接起来的一组命题。例如，$p \wedge q \wedge r$ 是一个合取式。析取式是由析取运算符连接起来的一组命题。例如，$p \vee q \vee r$ 是一个析取式。对于一个命题，如果命题满足下列条件，那么称之为合取范式(conjunctive normal form)。

- 在命题中不出现运算符⇒和⇔。
- ¬只出现在原子命题的前面。
- 如果命题包含∧和∨，那么命题是析取式的合取式。

因此，$p \vee q$、$p \wedge \neg q$ 和 $(p \vee q) \wedge (p \vee r)$ 都是合取范式，而 $\neg(p \wedge q)$ 和 $(p \wedge q) \vee (p \wedge r)$ 都不是合取范式。

使用等值推理，把下列命题转换成合取范式。

1. $p \Rightarrow (q \vee r)$

2. $(p \wedge q) \Rightarrow (q \vee r)$

练习 3.53 对于一个命题，如果它满足下列条件，那么称之为析取范式。

- 在命题中不出现运算符⇒和⇔。
- ¬只出现在原子命题的前面。
- 如果命题包含∧和∨，那么命题是合取式的析取式。

因此，$p \vee q$、$p \wedge \neg q$ 和 $(p \wedge q) \vee (p \wedge r)$ 都是析取范式，而 $\neg(p \wedge q)$ 和 $(p \vee q) \wedge (p \vee r)$ 都不是析取范式。

使用等值推理，把下列命题转换成析取范式。

1. $(p \wedge (q \vee r)) \vee p$

2. $(p \vee (q \wedge r)) \wedge p$

3.14 练习解答

3.1

1. true

2. true

3. false

3.2 可以对除第一个陈述外的其余陈述赋真值：我们没有足够的信息(例如是哪个乔恩)来确定第一个陈述的真假。第二个陈述是假的。第三个和第四个陈述都是真的。

3.3

1. "现在没有下雪。"

2. "乔恩不喜欢阿里。"

3. "x 小于等于 y。"

3.4

1. false

2. false

3. true

3.5

1. false

2. false

3. true

3.6 $(\neg p)\wedge(\neg q)$的真值表如下所示。

p	q	$\neg p$	$\neg q$	$(\neg p)\wedge(\neg q)$
true	true	false	false	**false**
true	false	false	true	**false**
false	true	true	false	**false**
false	false	true	true	**true**

3.7 $\neg(p\wedge q)$的真值表如下所示。

p	q	$p\wedge q$	$\neg(p\wedge q)$
true	true	true	**false**
true	false	false	**true**
false	true	false	**true**
false	false	false	**true**

3.8

1. 这一真值表有 8 行，因为有 8 种可能组合，可能的组合如下表所示。

p	q	r
true	true	true
true	true	false
true	false	true
true	false	false
false	true	true
false	true	false
false	false	true
false	false	false

2. $p\wedge q$ 的真值表有 4 行，$p\wedge q\wedge r$ 的真值表有 8 行，$p\wedge q\wedge r\wedge s$ 的真值表有 16 行。对于每种情况，每个原子命题都有两种可能的值：true 或 false。所以，含有 n 个原子命题的命题的真值表有 2^n 行。

3. 这样的真值表有一行，因为命题 true 只能是 true，而命题 false 只能是 false，即，只有一种值的可能组合。

3.9

1. 根据规则 3.2，$\neg\neg(\text{true}\wedge\text{false})$等值于 $\text{true}\wedge\text{false}$。根据规则 3.5，$\text{true}\wedge\text{false}$ 等值于 false。因此，这一命题逻辑等值于 false。

2. 根据规则 3.5，$p\wedge\text{false}$ 等值于 false。再根据规则 3.5，$\text{true}\wedge\text{false}$ 等值于 false。因此，这一命题逻辑等值于 false。

3. 根据规则 3.5，$p\wedge\text{false}$ 等值于 false。根据规则 3.1，$\neg\text{false}$ 等值于 true。根据规则 3.3 或规则 3.4，$\text{true}\wedge\text{true}$ 等值于 true。因此，这一命题逻辑等值于 true。

3.10

1. 根据规则 3.8，$(\text{false}\wedge p)\wedge q$ 等值于 $\text{false}\wedge(p\wedge q)$。根据规则 3.7，该式等值于$(p\wedge q)\wedge\text{false}$。根据规则 3.5，该式等值于 false。因此，这一命题逻辑等值于 false。

2. 根据规则 3.7，$p\wedge(\text{false}\wedge q)$等值于 $p\wedge(q\wedge\text{false})$。根据规则 3.5，该式等值于 $p\wedge\text{false}$。再根据规则 3.5，该式等值于 false。因此，这一命题逻辑等值于 false。

3.11

1. false

2. true

3. true

3.12

$\neg(p\vee q)$的真值表如下所示。

p	q	$p\vee q$	$\neg(p\vee q)$
true	true	true	**false**
true	false	true	**false**
false	true	true	**false**
false	false	false	**true**

3.13 $(\neg p)\vee(\neg q)$的真值表如下所示。

p	q	$\neg p$	$\neg q$	$(\neg p)\vee(\neg q)$
true	true	false	false	**false**
true	false	false	true	**true**
false	true	true	false	**true**
false	false	true	true	**true**

3.14

1. 根据规则 3.11，$p\vee\text{false}$ 等值于 p。

根据规则 3.12，$p \vee$ true 等值于 true。因此，这一命题逻辑等值于 true。

2. 根据规则 3.5，$p \wedge$ false 等值于 false。根据规则 3.12，false $\vee$ true 等值于 true。根据规则 3.1，$\neg$ true 等值于 false。因此，这一命题逻辑等值于 false。

3.15

1. 先看左边的析取式，根据规则 3.14，true $\vee p$ 等值于 $p \vee$ true。根据规则 3.12，$p \vee$ true等值于 true。根据规则 3.1，$\neg$ true 等值于 false。再看右边的析取式，根据规则 3.11，$p \vee$ false 等值于 p。现在考虑整个命题，根据规则 3.7，false $\wedge p$ 等值于 $p \wedge$ false。最后，根据规则 3.5，$p \wedge$ false 等值于 false。因此，这一命题逻辑等值于 false。

2. 根据规则 3.14，true $\vee q$ 等值于 $q \vee$ true。根据规则 3.13，$p \vee (q \vee$ true$)$等值于 $(p \vee q) \vee$ true。最后，根据规则 3.12，它等值于 true。因此，这一命题逻辑等值于 true。

3.16

1. 根据规则 3.17，$p \wedge ((\neg p) \vee q)$等值于$(p \wedge (\neg p)) \vee (p \wedge q)$。根据规则 3.6，这一陈述等值于 false $\vee (p \wedge q)$。根据规则 3.14，它等值于$(p \wedge q) \vee$ false。最后，根据规则 3.11，它等值于 $p \wedge q$。

2. 根据规则 3.16，$(p \wedge q) \vee - p$ 等值于$(p \vee (\neg p)) \wedge (q \vee (\neg p))$。根据规则 3.15，它等值于 true$\wedge (q \vee (\neg p))$。根据规则 3.7，它等值于$(q \vee (- p)) \wedge$ true。最后，根据规则 3.4，它等值于 $q \vee (\neg p)$。

3.17

1. $e \Rightarrow r$
2. $r \Rightarrow e$
3. $e \Rightarrow r$
4. $\neg (r \Rightarrow (\neg e))$

3.18

1. false⇒false，它等值于 true。
2. true⇒true，它等值于 true。
3. false⇒true，它等值于 true。
4. true⇒false，它等值于 false。

3.19 $(p \wedge q) \Rightarrow (p \vee q)$的真值表如下所示。

p	q	$p \wedge q$	$p \vee q$	$(\mathbf{p} \wedge \mathbf{q}) \Rightarrow (\mathbf{p} \vee \mathbf{q})$
true	true	true	true	**true**
true	false	false	true	**true**
false	true	false	true	**true**
false	false	false	false	**true**

3.20 $(p \vee q) \Rightarrow (p \wedge q)$的真值表如下所示。

p	q	$p \vee q$	$p \wedge q$	$(\mathbf{p} \vee \mathbf{q}) \Rightarrow (\mathbf{p} \wedge \mathbf{q})$
true	true	true	true	**true**
true	false	true	false	**false**
false	true	true	false	**false**
false	false	false	false	**true**

3.21

1. 根据规则 3.18，$p \Rightarrow p$ 等值于$(\neg p) \vee p$。根据规则 3.15，它等值于 true。

2. 根据规则 3.18，$(p \wedge q) \Rightarrow q$ 等值于$(-(p \wedge q)) \vee q$。根据规则 3.9，它等值于$((- p) \vee (\neg q)) \vee q$。根据规则 3.13，它等值于$(\neg p) \vee ((- q) \vee q)$。根据规则 3.15，它等值于$(\neg p) \vee$ true。根据规则 3.12，它等值于 true。

3.22

1. 假如⇒满足结合律，那么对于所有的命题 p、q 和 r，有 $p \Rightarrow (q \Rightarrow r)$逻辑等值于$(p \Rightarrow q) \Rightarrow r$。假设 p 为 false，q 为 true，r 为 false，则前一个命题逻辑等值于 false⇒false，它是 true，而后一个命题逻辑等值于 true⇒false，它是 false。因此，⇒不满足结合律。

2. 假如⇒满足交换律，那么对于所有的命题 p 和 q，一定有 $p \Rightarrow q$ 逻辑等值于 $q \Rightarrow p$。假设 p 为 true，q 为 false，那么 true⇒false 肯定不逻辑等值于 false⇒true。因此，⇒不满足交换律。

3. 假如⇒是幂等的，那么对于所有的命题 p，有 $p \Rightarrow p$ 逻辑等值于 p。假设 p 为

false，那么 $p \Rightarrow p$ 等值于 true。因此，$\Rightarrow$ 不是幂等的。

3.23

1. false⇔false，它等值于 true。
2. true⇔true，它等值于 true。
3. false⇔true，它等值于 false。
4. true⇔false，它等值于 false。

3.24

$(p \Rightarrow q) \Leftrightarrow (q \Rightarrow p)$的真值表如下所示。

p	q	$p \Rightarrow q$	$q \Rightarrow p$	$(p \Rightarrow q) \Leftrightarrow (q \Rightarrow p)$
true	true	true	true	**true**
true	false	false	true	**false**
false	true	true	false	**false**
false	false	true	true	**true**

3.25

1. 根据规则 3.23，$p \Leftrightarrow$ true 等值于 $(p \Rightarrow \text{true}) \wedge (\text{true} \Rightarrow p)$。根据规则 3.18，这一陈述等值于$((\neg p) \vee \text{true}) \wedge ((\neg \text{true}) \vee p)$。根据规则 3.12，它等值于 $\text{true} \wedge ((\neg \text{true}) \vee p)$。根据规则 3.1，它等值于 $\text{true} \wedge (\text{false} \vee p)$。根据规则 3.14，它等值于 $\text{true} \wedge (p \vee \text{false})$。根据规则 3.11，它等值于 $\text{true} \wedge p$。根据规则 3.7，它等值于 $p \wedge \text{true}$。根据规则 3.4，它等值于 p。由于等式左边逻辑等值于 p，而右边是 p，根据规则 3.21，这一命题等值于 true。

2. 根据规则 3.23，$p \Leftrightarrow$false 等值于$(p \Rightarrow \text{false}) \wedge (\text{false} \Rightarrow p)$。根据规则 3.18，这一陈述等值于$((\neg p) \vee \text{false}) \wedge ((\neg \text{false}) \vee p)$。根据规则 3.11，它等值于$(\neg p) \wedge ((\neg \text{false}) \vee p)$。根据规则 3.1，它等值于$(\neg p) \wedge (\text{true} \vee p)$。根据规则 3.14，它等值于$(\neg p) \wedge (p \vee \text{true})$。根据规则 3.12，它等值于$(\neg p) \wedge \text{true}$。根据规则 3.4，它等值于$\neg p$。由于等式左边逻辑等值于$\neg p$，而右边是$\neg p$，根据规则 3.21，这一命题等值于 true。

3.26 假如⇔是幂等的，那么对于任意的命题 p，一定有 $p \Leftrightarrow p$ 逻辑等值于 p。如果 p 为 false，那么 $p \Leftrightarrow p$ 等值于 true。因此，⇔不是幂等的。

3.27

1. $(\neg p) \vee q$
2. $p \vee (q \wedge r)$
3. $(p \wedge q) \Rightarrow r$
4. $(\neg p) \Leftrightarrow (((\neg q) \wedge r) \Rightarrow s)$

3.28

1. 如果 p 为 false 且 q 为 false，那么整个命题为 false，所以它不是重言式。

2. 无论我们为 p 和 q 赋什么值，这一命题都为 true，所以它是重言式。

3. 如果 q 为 true 且 p 为 false，那么整个命题为 false，所以它不是重言式。

4. 无论我们为 p 和 q 赋什么值，这一命题都为 true，所以它是重言式。

3.29

1. 无论我们为 p 和 q 赋什么值，这一命题都为 false，所以它是矛盾式。

2 无论我们为 p 和 q 赋什么值，这一命题都为 false，所以它是矛盾式。

3. 无论我们为 p 和 q 赋什么值，蕴含的第一参数都为 false，因此这一命题总为 true，所以它不是矛盾式。

3.30

1. 无论我们为 p 和 q 赋什么值，这一命题都为 true，所以它不是不定式。

2. 如果 p 为 true 且 q 为 false(或者相反)，那么这一命题为 false。另外，如果 p 和 q 都为 true(或都为 false)，那么这一命题为 true。因此，它是不定式。

3.31

1.

p	q	$p \wedge q$	$\neg(q \wedge p)$	$(p \wedge q) \vee \neg(p \wedge q)$
true	true	true	false	**true**
true	false	false	true	**true**
false	true	false	true	**true**
false	false	false	true	**true**

由于上表的最后一栏的每一项都为true，所以这一命题是重言式。

2.

p	q	$p\Rightarrow q$	$\neg q\Rightarrow\neg p$	$(p\Rightarrow q)\Leftrightarrow(\neg q\Rightarrow\neg p)$
true	true	true	true	**true**
true	false	false	false	**true**
false	true	true	true	**true**
false	false	true	true	**true**

由于上表的最后一栏的每一项都为true，所以这一命题是重言式。

3.

p	q	$\neg p\Rightarrow q$	$p\Rightarrow\neg q$	$(\neg p\Rightarrow q)\Rightarrow(p\Rightarrow\neg q)$
true	true	true	false	**false**
true	false	true	true	**true**
false	true	true	true	**true**
false	false	false	true	**true**

由于上表的最后一栏中有一项false和三项true，所以这一命题为不定式。

4.

p	q	$p\Rightarrow q$	$\neg p\Rightarrow\neg q$	$(p\Rightarrow q)\Rightarrow(\neg p\Rightarrow\neg q)$
true	true	true	true	**true**
true	false	false	true	**true**
false	true	true	false	**false**
false	false	true	true	**true**

由于上表的最后一栏中有一项false和三项true，所以这一命题为不定式。

5.

p	$p\Rightarrow p$	$\neg(p\Rightarrow q)$	$\neg p\Rightarrow\neg p$	$\neg(p\Rightarrow p)\wedge(\neg p\Rightarrow\neg p)$
true	true	false	true	**false**
false	true	false	true	**false**

由于上表的最后一栏的每一项都是false，所以这一命题为矛盾式。

3.32

1. $(p\Leftrightarrow\text{true})\Leftrightarrow p$

$\Leftrightarrow((p\Rightarrow\text{true})\wedge(\text{true}\Rightarrow p))\Leftrightarrow p$ [规则 3.23]

$\Leftrightarrow(((\neg p)\vee\text{true})\wedge((\neg\text{true})\vee p))\Leftrightarrow p$ [规则 3.18]

$\Leftrightarrow(\text{true}\wedge((\neg\text{true})\vee p))\Leftrightarrow p$ [规则 3.12]

$\Leftrightarrow(\text{true}\wedge(\text{false}\vee p))\Leftrightarrow p$ [规则 3.1]

$\Leftrightarrow(\text{true}\wedge(p\vee\text{false}))\Leftrightarrow p$ [规则 3.14]

$\Leftrightarrow(\text{true}\wedge p)\Leftrightarrow p$ [规则 3.11]

$\Leftrightarrow(p\wedge\text{true})\Leftrightarrow p$ [规则 3.7]

$\Leftrightarrow p\Leftrightarrow p$ [规则 3.4]

$\Leftrightarrow\text{true}$ [规则 3.21]

2. $(p\Leftrightarrow\text{false})\Leftrightarrow(\neg p)$

$\Leftrightarrow((p\Rightarrow\text{false})\wedge(\text{false}\Rightarrow p))\Leftrightarrow(\neg p)$ [规则 3.23]

$\Leftrightarrow(((\neg p)\vee\text{false})\wedge((\neg\text{false})\vee p))\Leftrightarrow(\neg p)$ [规则 3.18]

$\Leftrightarrow((\neg p)\wedge((\neg\text{false})\vee p))\Leftrightarrow(\neg p)$ [规则 3.11]

$\Leftrightarrow((\neg p)\wedge(\text{true}\vee p))\Leftrightarrow(\neg p)$ [规则 3.1]

$\Leftrightarrow((\neg p)\wedge(p\vee\text{true}))\Leftrightarrow(\neg p)$ [规则 3.14]

$\Leftrightarrow((\neg p)\wedge\text{true})\Leftrightarrow(\neg p)$ [规则 3.12]

$\Leftrightarrow(\neg p)\Leftrightarrow(\neg p)$ [规则 3.4]

$\Leftrightarrow\text{true}$ [规则 3.21]

3.33

1. $(p\wedge\neg q)\Rightarrow q$

$\Leftrightarrow\neg(p\wedge\neg q)\vee q$ [规则 3.18]

$\Leftrightarrow\neg p\vee q\vee q$ [规则 3.9]

$\Leftrightarrow\neg p\vee q$ [规则 3.10]

2. $(p\Rightarrow q)\wedge p$

$\Leftrightarrow(\neg p\vee q)\wedge p$ [规则 3.18]

$\Leftrightarrow(\neg p\wedge p)\vee(q\wedge p)$ [规则 3.17]

$\Leftrightarrow\neg(p\vee\neg p)\vee(q\vee p)$ [规则 3.9]

$\Leftrightarrow\neg(\neg p\vee p)\vee(q\vee p)$ [规则 3.13]

$\Leftrightarrow\neg\text{true}\vee(q\wedge p)$ [规则 3.15]

$\Leftrightarrow\text{false}\vee(q\wedge p)$ [规则 3.1]

$\Leftrightarrow(q\wedge p)\vee\text{false}$ [规则 3.14]

$\Leftrightarrow q \wedge p$ [规则 3.11]

3.34

1. $(p \Rightarrow q) \Leftrightarrow (\neg q \Rightarrow \neg p)$

$\Leftrightarrow (\neg p \vee q) \Leftrightarrow (\neg q \Rightarrow \neg p)$ [规则 3.18]

$\Leftrightarrow (\neg p \vee q) \Leftrightarrow (\neg\neg q \vee \neg p)$ [规则 3.18]

$\Leftrightarrow (\neg p \vee q) \Leftrightarrow (q \vee \neg p)$ [规则 3.2]

$\Leftrightarrow (q \vee \neg p) \Leftrightarrow (q \vee \neg p)$ [规则 3.13]

$\Leftrightarrow \text{true}$ [规则 3.21]

2. $(p \Leftrightarrow \text{true}) \Rightarrow p$

$\Leftrightarrow p \Rightarrow p$ [练习 3.32]

$\Leftrightarrow \text{true}$ [练习 3.21]

3.35

1. $(p \Leftrightarrow q) \Rightarrow (q \Leftrightarrow p)$

$\Leftrightarrow (p \Leftrightarrow q) \Rightarrow (p \Leftrightarrow q)$ [规则 3.20]

$\Leftrightarrow \text{true}$ [练习 3.32]

2. $(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$

$\Leftrightarrow (p \wedge q) \Leftrightarrow \neg(p \wedge q)$ [规则 3.9]

$\Leftrightarrow \text{false}$ [规则 3.22]

3.36

$$\dfrac{\dfrac{}{\text{true}}[\text{真引入规则}]}{\text{false} \Rightarrow \text{true}}[\text{蕴含引入规则}]$$

3.37

$$\dfrac{\dfrac{\dfrac{[p \wedge q]_1}{q}[\text{合取消去规则 2}] \quad \dfrac{[p \wedge q]_1}{p}[\text{合取消去规则 1}]}{q \wedge p}[\text{合取引入规则}]}{(p \wedge q) \Rightarrow (q \wedge p)}[\text{蕴含引入规则}_1]$$

3.38

1.

$$\dfrac{\dfrac{\dfrac{[p \wedge q]_1}{p}[\text{合取消去规则 1}]}{p \vee q}[\text{析取引入规则 1}]}{p \wedge q \Rightarrow p \vee q}[\text{蕴含引入规则}_1]$$

2.

$$\dfrac{\dfrac{\dfrac{}{\text{true}}[\text{真引入规则}]}{\text{true} \vee \text{false}}[\text{析取消去规则 1}]}{\text{false} \Rightarrow (\text{true} \vee \text{false})}[\text{蕴含引入规则}]$$

3.39 从树根出发，我们有如下证明树。

$$\dfrac{\dfrac{\dfrac{r \quad r \quad [p \vee q]_2}{r}[\text{析取消去规则}_3]}{(p \vee q) \Rightarrow r}[\text{蕴含引入规则}_2]}{((p \Rightarrow r) \wedge (q \Rightarrow r)) \Rightarrow ((p \vee q) \Rightarrow r)}[\text{蕴含引入规则}_1]$$

现在，我们只剩下两个任务：给定假设 p 和 $(p \Rightarrow r) \wedge (q \Rightarrow r)$ 来证明 r，以及给定假设 q 和 $(p \Rightarrow r) \wedge (q \Rightarrow r)$ 来证明 r。依次证明它们如下：

$$\dfrac{\dfrac{[(p \Rightarrow r) \wedge (q \Rightarrow r)]_1}{p \Rightarrow r}[\text{合取消去规则 1}] \quad [p]_3}{r}[\text{蕴含消去规则}]$$

$$\dfrac{\dfrac{[(p \Rightarrow r) \wedge (q \Rightarrow r)]_1}{q \Rightarrow r}[\text{合取消去规则 2}] \quad [q]_3}{r}[\text{蕴含消去规则}]$$

3.40

$$\dfrac{\dfrac{\dfrac{[p\Leftrightarrow q]_1}{p\Rightarrow q}[\text{等值消去规则 1}] \quad \dfrac{[p\Leftrightarrow q]_1}{q\Rightarrow p}[\text{等值消去规则 2}]}{(p\Rightarrow q)\wedge(q\Rightarrow p)}[\text{合取引入规则}]}{(p\Leftrightarrow q)\Rightarrow((p\Rightarrow q)\wedge(q\Rightarrow p))}[\text{蕴含引入规则}_1]$$

3.41

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[p\wedge q]_1}{q}[\wedge\text{elim2}]^{①} \quad \dfrac{[p\wedge q]_1}{p}[\wedge\text{elim1}]^{②}}{[\wedge\text{intro}]^{③}}}{\dfrac{q\wedge p}{(p\wedge q)\Rightarrow(q\wedge p)}[\Rightarrow\text{intro}_1]^{④}} \quad \dfrac{\dfrac{\dfrac{[q\wedge p]_2}{p}[\wedge\text{elim2}]^{①} \quad \dfrac{[q\wedge p]_2}{q}[\wedge\text{elim1}]^{②}}{[\wedge\text{intro}]^{③}}}{\dfrac{p\wedge q}{(q\wedge p)\Rightarrow(p\wedge q)}[\Rightarrow\text{intro}_2]^{⑤}}}{(p\wedge q)\Leftrightarrow(q\wedge p)}[\Leftrightarrow\text{intro}]^{⑥}$$

注：① 合取消去规则 2；② 合取消去规则 1；③ 合取引入规则；④ 蕴含引入规则$_1$；⑤ 蕴含引入规则$_2$；⑥ 等值引入规则。

3.42

1.

$$\dfrac{\dfrac{\dfrac{\dfrac{[p\wedge\neg p]_1}{p}[\text{合取消去规则 1}] \quad \dfrac{[p\wedge\neg p]_1}{\neg p}[\text{合取消去规则 2}]}{\text{false}}[\text{假引入规则}]}{\neg(p\wedge\neg p)}[\text{否定引入规则}]}{(p\wedge\neg p)\Rightarrow\neg(p\wedge\neg p)}[\text{蕴含引入规则}_1]$$

2.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg(p\vee\neg p)]_1}{\neg p\wedge\neg\neg p}[\text{德·摩根律}]}{\neg p}[\text{合取消去规则 1}] \quad \dfrac{\dfrac{[\neg(p\vee\neg p)]_1}{\neg p\wedge\neg\neg p}[\text{德·摩根律}]}{\neg\neg p}[\text{合取消去规则 2}]}{\text{false}}[\text{假引入规则}]}{p\vee\neg p}[\text{否定消去规则}]}{\neg(p\vee\neg p)\Rightarrow(p\vee\neg p)}[\text{蕴含引入规则}_1]$$

3.43

1. “乔恩不高兴。”
2. “乔恩高兴且史蒂夫痛苦。”
3. “乔恩高兴或史蒂夫痛苦。”
4. “如果乔恩高兴，则史蒂夫痛苦。”
5. “乔恩高兴当且仅当史蒂夫痛苦。”

3.44

1. $\neg q$
2. $\neg q\wedge\neg p$
3. $q\vee\neg p$
4. $q\Rightarrow p$
5. $\neg p\Leftrightarrow q$

3.45

1. “如果乔恩高兴且史蒂夫痛苦，则阿里生气。”
2. “或者乔恩高兴，或者史蒂夫痛苦且阿里不生气。”
3. “乔恩高兴当且仅当史蒂夫痛苦且阿里生气。”

3.46

1. $p\wedge\neg(q\wedge r)$
2. $r\Rightarrow(p\vee q)$
3. $q\Leftrightarrow p\wedge\neg r$

3.47

1.

p	q	$\neg q$	$p\vee\neg q$
true	true	false	**true**
true	false	true	**true**
false	true	false	**false**
false	false	true	**true**

2.

p	q	$\neg p$	$\neg p\wedge q$
true	true	false	**false**
true	false	false	**false**
false	true	true	**true**
false	false	true	**false**

3.

p	q	$p\wedge q$	$p\Rightarrow(p\wedge q)$
true	true	true	**true**
true	false	false	**false**
false	true	false	**true**
false	false	false	**true**

4.

p	q	$\neg p$	$\neg p\Leftrightarrow q$
true	true	false	**false**
true	false	false	**true**
false	true	true	**true**
false	false	true	**false**

3.48

$p\Rightarrow(p\wedge q)$

$\Leftrightarrow\neg p\vee(p\wedge q)$ [规则 3.18]

$\Leftrightarrow(\neg p\vee p)\wedge(\neg p\vee q)$ [规则 3.16]

$\Leftrightarrow \text{true}\wedge(\neg p\vee q)$ [规则 3.15]

$\Leftrightarrow(\neg p\vee q)\wedge \text{true}$ [规则 3.8]

$\Leftrightarrow\neg p\vee q$ [规则 3.4]

$\Leftrightarrow p\Rightarrow q$ [规则 3.18]

3.49

$(\neg p\Rightarrow p)\Leftrightarrow p$

$\Leftrightarrow((\neg p\Rightarrow p)\Rightarrow p)\wedge(p\Rightarrow(\neg p\Rightarrow p))$ [规则 3.23]

$\Leftrightarrow(\neg(\neg p\Rightarrow p)\vee p)\wedge(\neg p\vee(\neg p\Rightarrow p))$ [规则 3.18]

$\Leftrightarrow(\neg(p\vee p)\vee p)\wedge(\neg p\vee(p\vee p))$ [规则 3.18]

$\Leftrightarrow(\neg p\vee p)\wedge(\neg p\vee p)$ [规则 3.10]

$\Leftrightarrow \text{true}\wedge \text{true}$ [规则 3.15]

$\Leftrightarrow \text{true}$ [规则 3.4]

3.50

$$\dfrac{\dfrac{[p]_1}{q\Rightarrow p}\,[\text{蕴含引入规则}]}{p\Rightarrow(q\Rightarrow p)}\,[\text{蕴含引入规则}_1]$$

3.51

$$\dfrac{\dfrac{\dfrac{\dfrac{[p\wedge\neg p]_1}{p}\,[\text{合取消去规则 1}]\quad \dfrac{[p\wedge\neg p]_1}{\neg p}\,[\text{合取消去规则 2}]}{\text{false}}\,[\text{假引入规则}]}{\text{true}\Rightarrow\text{false}}\,[\text{蕴含引入规则}]}{(p\wedge\neg p)\Rightarrow(\text{true}\Rightarrow\text{false})}\,[\text{蕴含引入规则}_1]$$

3.52

1. $p\Rightarrow(q\vee r)$

$\Leftrightarrow\neg p\vee(q\vee r)$ [规则 3.18]

$\Leftrightarrow\neg p\vee q\vee r$ [析取结合律]

2. $(p\wedge q)\Rightarrow(q\vee r)$

$\Leftrightarrow\neg(p\wedge q)\vee(q\vee r)$ [规则 3.18]

$\Leftrightarrow(\neg p\vee\neg q)\vee(q\vee r)$ [规则 3.9]

$\Leftrightarrow\neg p\vee\neg q\vee q\vee r$ [析取结合律]

3.53

1. $(p\wedge(q\vee r))\vee p$

$\Leftrightarrow((p\wedge q)\vee(p\wedge r))\vee p$ [规则 3.16]

$\Leftrightarrow(p\wedge q)\vee(p\wedge r)\vee p$ [析取结合律]

2. $(p\vee(q\wedge r))\wedge p$

$\Leftrightarrow(p\wedge p)\vee(p\wedge(q\wedge r))$ [规则 3.17]

$\Leftrightarrow p\vee(p\wedge(q\wedge r))$ [规则 3.3]

$\Leftrightarrow p\vee(p\wedge q\wedge r)$ [合取结合律]

第4章 集 合 论

命题逻辑和集合论是离散数学的两个基本组成模块。第3章我们已经讲述了命题逻辑的基础知识，本章介绍集合论的一些基础理论。

想要了解集合论方面更多知识的读者可以参考[End77]、[Hen86]或[CLP01]。

4.1 集合

简单地讲，集合(set)是一些事物的无序汇集。举例来说，在第2章介绍过的自然数形成一个集合，记作$\mathbb{N}$；同样，所有实数组成一个集合，记作$\mathbb{R}$。另外，世界上所有的国家组成一个集合，称作*Countries*；同样，世界上所有的人组成一个集合，称作*People*。

在上面的例子中，我们通过名称来表示集合，并且对集合中所包含的事物有了一些基本的概念：也就是说，对集合的元素(或成员)有了基本的了解。

有时，需要更加具体一些；有时，仅仅提及抽象的对象还不够，例如表示世界上所有人的*People*。这时，也许需要使用元素来定义集合；例如，可以定义一个包括曾经赢得过世界杯足球赛的所有国家的集合。

做这件事情的方法是将集合中的所有元素列出来，并且在它们的两边加上花括号"{"和"}"。每一个元素都写在花括号内，元素间用逗号分开。所以，可以用下面的方式定义世界杯冠军国家的集合。

Winners = {阿根廷，巴西，英格兰，法国，德国，意大利，乌拉圭}

我们把这种定义集合的方法称为外延法(extension)或枚举法(listing)。

例4.1 格雷厄姆(Graham)有3张CD：一张是Abba的，一张是Westlife的，一张是Madonna的。包含这三张CD的集合称为*graham_cds*(Graham的CD)，表示为：

graham_cds={abba,westlife,madonna}

□

例4.2 撞球游戏中，把所有的球(除了白球)从1到15进行编号。这样，可以用下面的方法定义集合*pool_balls*：

pool_balls = {1，2，3，4，5，6，7，8，9，10，11，12，13，14，15} □

练习4.1 列出下列集合中的元素。

1. 英语字母中所有元音字母构成的集合。

2. 比2大的所有素偶数构成的集合。

3. 在短语"the set of letters"中出现的字母构成的集合。 □

练习4.2 用外延法定义下列集合。

1. 小于6的自然数的集合。

2. 小于6的奇数的集合。

3. 短语"the set of vowels"中出现的元音字母组成的集合。 □

任何一个元素在给定的集合中只能出现一次。例如，集合{1，2，3，3}和{1，2，2，3}是相等的。实际上，它们都等同于{1，2，3}。

例4.3 集合*fruitbowl*只能告诉我们该集合包含什么样的水果，而不能告诉我们每一种水果的数目。

fruitbowl = {香蕉，梨，苹果，橘子，桃子}

这个集合等同于

{香蕉，香蕉，梨，梨，苹果，橘子，桃子} □

另外，元素在集合中出现的顺序也是无

关紧要的。因此，集合{3，1，2}、{1，2，3}和{2，1，3}都是相等的。

例 4.4 集合 *eaten* 只能告诉我们哪种水果已经被吃掉了，而不能告诉我们水果被吃掉的顺序。

eaten={香蕉，橘子，梨}

这个集合等同于

{梨，橘子，香蕉} □

例 4.5 下面的每一个集合都等同于集合{杰克，吉尔}。

{吉尔，杰克}，{杰克，杰克，吉尔}，{杰克，吉尔，杰克}，{杰克，吉尔，吉尔}，{吉尔，杰克，杰克}，{吉尔，杰克，吉尔}，{吉尔，吉尔，杰克} □

当两个集合含有完全相同的元素时，我们称这两个集合相等。若集合 S 和集合 T 相等，写作 $S=T$。例如，因为我们已经知道了集合没有顺序的概念，所以{a，b}和{b，a}是相等的，可以写成

$\{a, b\}=\{b, a\}$

另外，因为集合没有重复的概念，所以

$\{a, b, b, a\}=\{a, b\}$

练习 4.3 下列集合中哪些是相等的？

1. 单词"spear"中包含的字母所组成的集合。

2. 单词"pears"中包含的字母所组成的集合。

3. 单词"spares"中包含的字母所组成的集合。

4. 单词"spears"中包含的字母所组成的集合。 □

4.2 单集

如果一个集合只包含一个元素，那么我们称它为单集(singleton set)。例如，集合{a}就是一个单集，同样集合{b}也是。相反，集合{a，b}就不是一个单集，因为它包含了两个元素。

例 4.6 24 与 30 间的素数构成的集合是单集，即集合{29}。 □

例 4.7 24 与 30 间的偶数构成的集合不是单集，这个集合为{26，28}。 □

练习 4.4 下列集合中哪些是单集？

1. 除了 1 和其本身外，9 的所有约数构成的集合。

2. 除了 1 和其本身外，11 的所有约数构成的集合。

3. 除了 1 和其本身外，16 的所有约数构成的集合。 □

4.3 空集

就像我们在练习 4.4 中所看到的，在有些情况下，一个集合不包含任何元素；这些集合都等同于空集(empty set)，即不含任何元素的集合。我们把空集记作∅或{}。

例 4.8 32 与 36 间的素数构成的集合是空集，大于 2 的素偶数构成的集合也是空集。 □

练习 4.5 下列集合中哪些等同于空集？

1. 按英文字母表的顺序，Z 以后的字母所组成的集合。

2. 世界上不是联合国会员的国家所组成的集合。

3. 没有后继数的自然数所组成的集合。 □

4.4 集合成员

给定一个集合 S，很容易推断出一个元素是否在该集合中。给定一个元素 s，如果元素 s 在集合 S 中出现，就说 s 是 S 的一个成员(member)。记作 $s\in S$。

例 4.9 数字 1 是自然数集合中的一个元素，所以可以写作 $1\in \mathbb{N}$。 □

例 4.10 参考例 4.3，集合 *fruitbowl*

包含元素香蕉，所以可以写作香蕉$\in$ *fruitbowl*。 □

如果元素 t 不在集合 S 中出现，就说 t 不是 S 的一个成员，记作 $t \notin S$。

例 4.11 数 -1 不是自然数集合 $\mathbb{N}$ 中的元素，写作 $-1 \notin \mathbb{N}$。 □

例 4.12 参考例 4.3，集合 *fruitbowl* 不包含元素柠檬，所以写作柠檬 $\notin$ *fruitbowl*。 □

所以，$\notin$ 是 $\in$ 的补。

规则 4.1 对于任意的集合 S 和任意的元素 s，有

$\neg(s \in S) \Leftrightarrow s \notin S$ □

另外，任何元素都不属于空集。因此，对于任何元素 x，陈述 $x \in \varnothing$ 都逻辑等值于 false。

规则 4.2 对于任意的元素 x，有

$x \in \varnothing \Leftrightarrow \text{false}$ □

练习 4.6 假设集合 *Primes* 表示所有素数组成的集合，集合 *Evens* 表示所有偶数组成的集合，下列陈述中哪些为真，哪些为假？

1. $6 \in Primes$
2. $6 \notin Evens$
3. $7 \in Primes$
4. $7 \notin Evens$ □

到现在为止，仅当 x 是一个元素且 X 是一个集合时，才能够写 $x \in X$（我们将在后面对这一限制进行形式定义）。例如，如果集合 S 是由自然数 2、3 和 4 组成的集合，即 $S=\{2, 3, 4\}$，可以写 $2 \in S$ 和 $5 \notin S$，但不能写 $\{2\} \in S$ 或 $\{5\} \notin S$。陈述 $2 \in S$ 为真且陈述 $5 \in S$ 为假。然而，陈述 $\{2\} \in S$ 和 $\{5\} \notin S$ 的值未定义，因为 $\{2\}$ 和 $\{5\}$ 不是元素而是集合。

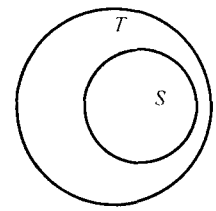
练习 4.7 假设集合 S 是由英文字母 a、b 和 c 所组成的集合，即 $S=\{a, b, c\}$。下面哪些陈述为真，哪些为假，哪些未定义？

1. $a \in S$
2. $d \in S$
3. $\varnothing \in S$
4. $\{d\} \in S$ □

4.5 子集

有的时候，我们想要知道一个集合是否包含在另一个集合之中。例如，在第 2 章，我们看到自然数集合中的每一个元素都包含在整数集合中，同样，整数集合中的每一个元素都包含在实数集合中。

给定两个集合 S 和 T，如果 S 中的每个元素都包含在 T 中，那么我们就说 S 是 T 的一个子集(subset)，写作 $S \subseteq T$。例如，如果 $S=\{1, 3, 5\}$ 且 $T=\{1, 3, 5, 7\}$，则有 $S \subseteq T$。集合的这种关系可以用文氏图(Venn diagram)[1]来形象地表示，$S \subseteq T$ 可以如下表示。



在这里，整个集合 S 都包含在集合 T 内。

例 4.13 由于每个自然数都包含在整数集合中，每个整数都包含在实数集合中，因此，可以写作 $\mathbb{N} \subseteq \mathbb{Z}$ 和 $\mathbb{Z} \subseteq \mathbb{R}$。 □

例 4.14 所有素食者组成的集合 *Vegetarian* 是所有人组成的集合 *People* 的子集。因此，$Vegetarian \subseteq People$。 □

前面介绍过，如果两个集合 S 和 T 包

[1] 文氏图是一种抽象的图形表示，使用它可以表示不同集合之间的关系：子集关系是其中的一个基本例子。对于文氏图的全面介绍，参看[Hun96]。

含完全相同的元素，那么称它们是相等的，写作 $S=T$。在这种情况下，有 $S\subseteq T$ 且 $T\subseteq S$。

规则 4.3 对于任意的集合 S 和 T，有

$(S\subseteq T\wedge T\subseteq S)\Leftrightarrow S=T$ □

例 4.15 定义集合 A 和 B 如下。

$A=\{$巴里，约翰$\}$

$B=\{$约翰，巴里$\}$

在这里，$A\subseteq B$ 并且 $B\subseteq A$。所以，有 $A=B$。

□

空集是所有集合的子集。

规则 4.4 对于任意集合 S，有

$\varnothing\subseteq S$ □

每个集合都是它自身的子集。

规则 4.5 对于任意的集合 S，有

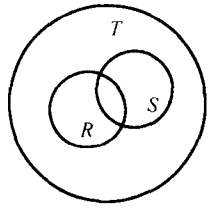
$S\subseteq S$ □

练习 4.8 下列陈述哪些为真，哪些为假？

1. $\varnothing\subseteq\{$猫，狗$\}$
2. $\{$猫$\}\subseteq\{$猫，狗$\}$
3. $\{$猫，狗$\}\subseteq\{$猫，狗$\}$
4. $\{$猫，狗$\}\subseteq\varnothing$
5. $\{$猫，狗$\}\subseteq\{$猫$\}$ □

当一个集合的某些元素不在另一个集合中时，我们称前一个集合不是后一个集合的子集。

例 4.16 在下面的文氏图中，集合 R 不是集合 S 的子集，虽然它是集合 T 的子集。



在这种情况下，可以写作 $R\nsubseteq S$ 和 $R\subseteq T$。另外，对于这个例子，也可以写 $S\subseteq T$、$S\nsubseteq R$、$T\nsubseteq R$、$T\nsubseteq S$。 □

下面这个规则刻画了 $\subseteq$ 和 $\nsubseteq$ 之间的关系。

规则 4.6 对于任意的集合 S 和 T，有

$\neg(S\subseteq T)\Leftrightarrow S\nsubseteq T$ □

例 4.17 整数集合不是自然数集合的子集：$\mathbb{Z}\nsubseteq\mathbb{N}$。 □

练习 4.9 下面的陈述哪些为真，哪些为假？

1. $\varnothing\nsubseteq\varnothing$
2. $\varnothing\nsubseteq\{a\}$
3. $\{a\}\nsubseteq\varnothing$
4. $\{a\}\nsubseteq\{a\}$ □

如果一个集合 S 完全包含在另一个集合 T 中，且 T 中有某些元素不在 S 中出现，那么可以说 S 是 T 的真子集（proper subset），记作 $S\subset T$。

例 4.18 回忆我们以前举过的例子：集合 *eaten* 和集合 *fruitbowl* 的定义如下：

fruitbowl$=\{$香蕉，梨，苹果，橘子，桃子$\}$

eaten$=\{$香蕉，橘子，梨$\}$

因此，有 *eaten*$\subset$*fruitbowl*。 □

下面这个规则刻画了 $\subseteq$、$\subset$ 和 $=$ 之间的关系。

规则 4.7 对于任意的集合 S 和 T，有

$S\subseteq T\Leftrightarrow(S\subset T\vee S=T)$ □

练习 4.10 下列陈述中哪些为真，哪些为假？

1. $\varnothing\subset\{$猫，狗$\}$
2. $\{$猫$\}\subset\{$猫，狗$\}$
3. $\{$猫，狗$\}\subset\{$猫，狗$\}$
4. $\{$猫，狗$\}\subset\varnothing$
5. $\{$猫，狗$\}\subset\{$猫$\}$ □

如果 S 不是 T 的真子集，则记作 $S\not\subset T$。下面这个规则刻画了 $\subset$ 和 $\not\subset$ 之间的关系。

规则 4.8 对于任意的集合 S 和 T，有

$S\not\subset T\Leftrightarrow\neg(S\subset T)$ □

对于任意的集合 S，S 不是它自己的真子集。

规则 4.9 对于任意的集合 S，有

$S \not\subset S$ □

另外，如果一个集合 S 是另一个集合 T 的真子集，那么 T 就不可能是 S 的真子集。

规则 4.10 对于任意的两个集合 S 和 T，有

$S \subset T \Rightarrow T \not\subset S$ □

例 4.19 下面的陈述都是真的。

$\{1, 2\} \not\subset \{1\}$

$\{1, 2\} \not\subset \{1, 2\}$ □

练习 4.11 下列陈述哪些为真，哪些为假？

1. $\varnothing \not\subset \{$猫，狗$\}$
2. $\{$猫$\} \not\subset \{$猫，狗$\}$
3. $\{$猫，狗$\} \not\subset \{$猫，狗$\}$
4. $\{$猫，狗$\} \not\subset \varnothing$
5. $\{$猫，狗$\} \not\subset \{$猫$\}$ □

练习 4.12 设集合 $S=\{1, 2, 3\}$，下列陈述哪些为真，哪些为假？

1. $\{1\} \subseteq S$
2. $\{1, 2, 3\} \subseteq S$
3. $\{1, 2, 3\} \subset S$
4. $\{1\} \subset S$ □

练习 4.13 给定 S 为练习 4.12 所定义的集合，列举出 S 的所有子集。 □

练习 4.14 假设 S 为练习 4.12 所定义的集合，列举出 S 的所有真子集。 □

练习 4.15 证明下面的论断。

$S \subseteq \varnothing \Rightarrow S=\varnothing$ □

与符号 $\in$ 的情况类似，我们必须严谨地考虑符号 $\subseteq$（或它的派生符号）的使用环境。仅当 S 和 T 都是集合时，才可以写 $S \subseteq T$，反之，如果 S 和 T 中任何一个是元素的话，就不能写 $S \subseteq T$[①]。例如，可以这样写：$\{1\} \subseteq \{1, 2, 3\}$（这一陈述是真的）以及 $\{1, 2, 3\} \subseteq \{1\}$（这一陈述是假的）；但是，不能写 $1 \subseteq \{1, 2, 3\}$ 或 $\{1, 2, 3\} \subseteq 1$，因为这两个陈述的值未定义。

练习 4.16 假设 $S=\{a, b, c\}$。下列哪些为真，哪些为假，哪些未定义？

1. $a \subseteq S$
2. $d \subseteq S$
3. $\varnothing \subseteq S$
4. $\{d\} \not\subseteq S$ □

4.6 超集

超集关系与子集关系是互补的：当集合 T 刚好是集合 S 的子集时，S 是 T 的超集。例如，集合 $\{1\}$ 是集合 $\{1, 2\}$ 的子集，所以集合 $\{1, 2\}$ 是集合 $\{1\}$ 的超集（superset）。对于这种情况，写作 $\{1, 2\} \supseteq \{1\}$。

规则 4.11 对于任意的集合 S 和 T，有

$S \supseteq T \Leftrightarrow T \subseteq S$ □

在这里，符号 $\subseteq$ 与 $\supseteq$ 间的关系类似于符号 $\leqslant$ 与 $\geqslant$ 间的关系。例如，对于两个自然数 m 和 n，$m \leqslant n$ 当且仅当 $n \geqslant m$，类似地，对于两个集合 S 和 T，$S \subseteq T$ 当且仅当 $T \supseteq S$。

与符号 $\subseteq$ 的情况类似，关于 $\supseteq$ 也有一系列的派生关系：$\not\supseteq$、$\supset$ 和 $\not\supset$。

例 4.20 下列所有陈述都为真。

$\varnothing \not\supseteq \{a, b\}$

$\{a\} \not\supseteq \{a, b\}$

$\{a, b\} \supset \varnothing$

$\{a, b\} \supset \{a\}$

$\{a\} \not\supset \{a\}$

$\varnothing \not\supset \{a\}$ □

练习 4.17 下列陈述中哪些为真，哪些为假？

1. $\{$杰克$\} \supseteq \{$杰克，吉尔$\}$
2. $\{$杰克$\} \supset \{$杰克，吉尔$\}$
3. $\{$杰克$\} \not\supseteq \{$杰克$\}$
4. $\{$杰克$\} \not\supset \{$杰克$\}$
5. $\{$杰克$\} \supseteq \varnothing$

① 我们将在第 6 章形式地给出这一规则。

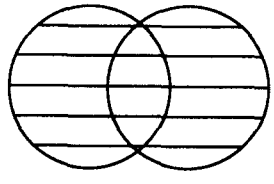
6. {杰克}⊃∅ □

4.7 集合的并集

到目前为止，我们已经介绍了一系列论述集合间关系的方法。例如，可以决定一个元素是否在一个集合中出现，或者一个集合是否是另一个集合的子集。下面将介绍建立在集合上的一些运算符，这些运算符给我们提供了由已知的两个或多个集合构造出新集合的方法。

首先介绍的运算符是集合的并集(set union)。这个运算符将两个集合组合成一个更大的集合，新集合包含原来两个集合的所有元素，请记住：任意一个元素都只能在任意给定集合中出现一次。给定两个集合 S 和 T，S 和 T 的并集记作 $S\cup T$。

参照下面的文氏图，包含在集合 S 和 T 的并集中的元素是图中突出显示的那些元素。



例 4.21 如果 $S=\{1, 2, 3\}$ 且 $T=\{2, 3, 4\}$，则 $S\cup T=\{1, 2, 3, 4\}$。 □

形式地，一个元素 a 出现在集合 S 和 T 的并集中，当且仅当它至少在 S 或 T 中的一个里出现。

规则 4.12 对于任意的元素 a 以及任意的集合 S 和 T，有

$a\in S\cup T\Leftrightarrow(a\in S\vee a\in T)$ □

例 4.22 Graham 和 Christine 决定将他们收藏的 CD 合并起来，Graham 拥有的 CD 为：

graham_cds={abba, westlife, madonna}

Christine 同样有三张 CD：

christine_cds={britney_spears, spice_girls, the_beatles}

那么，他们两个人的收藏的并为 *graham_cds* ∪*christine_cds*，等同于集合

{abba, westlife, madonna , britney_spears, spice_girls, the_beatles } □

这里列举几个与集合的并相关的规则。首先，空集对集合的并运算不产生影响。

规则 4.13 对于任意的集合 S，有

$S\cup\varnothing=S$ □

例 4.23

$\{a, b\}\cup\varnothing=\{a, b\}$ □

其次，任意的集合 S 与其自身的并仍然是它自己。

规则 4.14 对于任意的集合 S，有

$S\cup S=S$ □

例 4.24

$\{a, b\}\cup\{a, b\}=\{a, b\}$ □

集合的并也拥有我们在第3章首次遇到的两个性质：交换律和结合律。

规则 4.15 对于任意的集合 S 和 T，有

$S\cup T=T\cup S$ □

规则 4.16 对于任意的集合 R、S 和 T，有

$R\cup(S\cup T)=(R\cup S)\cup T$ □

例 4.25

$$\{a, b\}\cup(\{b, c\}\cup\{c, d\})$$
$$=(\{a, b\}\cup\{b, c\})\cup\{c, d\}$$ □

最后，两个集合的并集总是至少与这两个集合一样大。

规则 4.17 对于任意的集合 S 和 T，有 □

$S\subseteq S\cup T$

例 4.26

$\{a, b\}\subseteq\{a, b\}\cup\{b, c\}$ □

练习 4.18 计算下列各题。

1. $\{1, 2, 3\}\cup\varnothing$
2. $\{1, 2, 3\}\cup\{1, 2, 3\}$

3. $\{1, 2, 3\} \cup \{4\}$

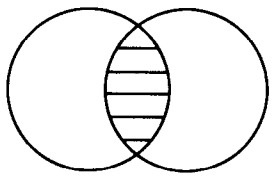
4. $\{1, 2, 3\} \cup \{1, 2, 3, 4\}$

5. $\{1, 2\} \cup \{3, 4\}$ □

4.8 集合的交集

有些时候，我们关注同时出现在两个不同集合中的那些元素。例如，有些人既阅读报纸 *The Guardian* 又收看有线电视，我们想收集有关这些人的情况；如果某个人只符合其中一个标准，那么我们就可以不关心这个人的情况。给定两个集合 S 和 T，那些在 S 和 T 中都出现的元素所组成的集合称为 S 和 T 的交集(intersection)，记作 $S \cap T$。

参看下面的文氏图，集合 S 和 T 的交集中的元素是图中突出显示的那些元素。



例 4.27 如果 $S=\{1, 2, 3\}$ 且 $T=\{2, 3, 4\}$，那么 $S \cap T=\{2, 3\}$。 □

形式地，一个元素 a 出现在集合 S 和 T 的交集中，当且仅当它在 S 和 T 中同时出现。

规则 4.18 对于任意的元素 a 及任意的集合 S 和 T，有

$a \in S \cap T \Leftrightarrow (a \in S \wedge a \in T)$ □

例 4.28 Graham 和 Christine 两个人没有相同的 CD，他们的 CD 收藏分别如下：

graham_cds={abba, westlife, madonna}

christine_cds={britney_spears, spice_girls, the_beatles}

那么，这两个集合的交集为

graham_cds ∩ *christine_cds* = ∅ □

与集合的并集的情况类似，有关集合的交集也有一些规则。首先，任何集合与空集的交集都等于空集。

规则 4.19 对于任意的集合 S，有

$S \cap \varnothing = \varnothing$ □

例 4.29

$\{a, b\} \cap \varnothing = \varnothing$ □

其次，任意集合 S 与其自身的交集等于它自己。

规则 4.20 对于任意的集合 S，有

$S \cap S = S$ □

例 4.30

$\{a, b\} \cap \{a, b\} = \{a, b\}$ □

同集合的并集相同，集合的交集也满足交换律和结合律。

规则 4.21 对于任意的集合 S 和 T，有

$S \cap T = T \cap S$ □

规则 4.22 对于任意的集合 R、S 和 T，有

$R \cap (S \cap T) = (R \cap S) \cap T$ □

例 4.31

$$\{a, b\} \cap (\{b, c\} \cap \{c, d\})$$
$$= (\{a, b\} \cap \{b, c\}) \cap \{c, d\}$$ □

另外，两个集合的交集总是至少与这两个集合一样小。

规则 4.23 对于任意的集合 S 和 T，有

$S \cap T \subseteq S$ □

例 4.32

$\{a, b\} \cap \{b, c\} \subseteq \{a, b\}$ □

最后，$\cup$ 对 $\cap$ 满足分配律，反之，$\cap$ 对 $\cup$ 也满足分配律。

规则 4.24 对于任意的集合 R、S 和 T，有

$R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$

$R \cap (S \cup T) = (R \cap S) \cup (R \cap T)$ □

我们通过下面的例子来说明分配律中的第一个规则。

例 4.33 假设 $R=\{a, b\}$，$S=\{b, c\}$，$T=\{c, d\}$。

这里，式 $R \cup (S \cap T) = (R \cup S) \cap (R \cup T)$ 左

边的计算结果如下：

$$R\cup(S\cap T)=\{a,b\}\cup(\{b,c\}\cap\{c,d\})$$
$$=\{a,b\}\cup\{c\}$$
$$=\{a,b,c\}$$

式 $R\cup(S\cap T)=(R\cup S)\cap(R\cup T)$ 右边的计算结果如下：

$$(R\cup S)\cap(R\cup T)$$
$$=(\{a,b\}\cup\{b,c\})\cap(\{a,b\}\cup\{c,d\})$$
$$=\{a,b,c\}\cap\{a,b,c,d\}$$
$$=\{a,b,c\}$$

这样，可以看到等式的两边的确相等。 □

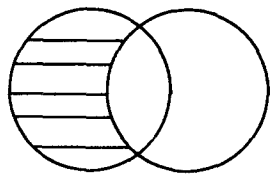
练习 4.19 计算下列各题。

1. $\{1,2,3\}\cap\varnothing$
2. $\{1,2,3\}\cap\{1,2,3\}$
3. $\{1,2,3\}\cap\{4\}$
4. $\{1,2,3\}\cap\{1,2,3,4\}$
5. $\{1,2\}\cap\{3,4\}$ □

4.9 集合的差集

集合的差集运算符使我们可以得到属于集合 S 但不属于集合 T 的元素，记作 $S\setminus T$。例如，读报纸 *The Guardian* 的人的集合记作 G，看有线电视的人的集合记作 C，那么可以考虑看报纸 *The Guardian* 而不看有线电视的人的集合 $G\setminus C$。

参看下面的文氏图，集合 S 和 T 的差集 $S\setminus T$ 中的元素是图中突出显示的那些元素。



例 4.34 如果 $S=\{1,2,3\}$ 且 $T=\{2,3,4\}$，那么 $S\setminus T=\{1\}$。 □

例 4.35 考虑集合 $S=\{a,b,c\}$ 和 $T=\{b,c,d\}$。那么有

$S\setminus T=\{a\}$

$T\setminus S=\{d\}$ □

形式地，一个元素 a 在集合 $S\setminus T$ 中出现，当且仅当它在 S 中出现且不在 T 中出现。

规则 4.25 对于任意的元素 a 及任意的集合 S 和 T，有

$a\in S\setminus T\Leftrightarrow(a\in S\wedge a\notin T)$ □

例 4.36 给定集合 $S=\{$乔恩，吉姆，戴夫$\}$和 $T=\{$吉姆$\}$，那么 $S\setminus T=\{$乔恩，戴夫$\}$：元素"乔恩"和"戴夫"都只在 S 中出现而不在 T 中出现。 □

同样，有关集合的差集也有一些相应的规则。首先，对于任意集合 S，从 S 中取走空集的元素不会对 S 产生影响。

规则 4.26 对于任意的集合 S，有

$S\setminus\varnothing=S$ □

例 4.37

$\{a,b\}\setminus\varnothing=\{a,b\}$ □

其次，给定任意集合 S，从 S 中取走它的所有元素将剩下一个空集。

规则 4.27 对于任意的集合 S，有

$S\setminus S=\varnothing$ □

例 4.38

$\{a,b\}\setminus\{a,b\}=\varnothing$ □

另外，空集没有元素可以取走。

规则 4.28 对于任意的集合 S，有

$\varnothing\setminus S=\varnothing$ □

例 4.39

$\varnothing\setminus\{a,b\}=\varnothing$ □

集合的差集还有下面一些有趣的性质。

规则 4.29 对于任意的集合 R、S 和 T，有

$R\setminus(S\cup T)=(R\setminus S)\cap(R\setminus T)$

$R\setminus(S\cap T)=(R\setminus S)\cup(R\setminus T)$ □

我们用下面的例子说明式 $R\setminus(S\cup T)=(R\setminus S)\cap(R\setminus T)$ 的正确性。

例 4.40 假设 $R=\{a,b\}$，$S=\{b,c\}$，$T=\{c,d\}$。考虑下面的式子。

$R\setminus(S\cup T)=(R\setminus S)\cap(R\setminus T)$

其左边的计算结果如下：

$$R\setminus(S\cup T)=\{a,b\}\setminus(\{b,c\}\cup\{c,d\})$$
$$=\{a,b\}\setminus\{b,c,d\}$$
$$=\{a\}$$

而其右边的计算结果如下：

$$(R\setminus S)\cap(R\setminus T)$$
$$=(\{a,b\}\setminus\{b,c\})\cap(\{a,b\}\setminus\{c,d\})$$
$$=\{a\}\cap\{a,b\}$$
$$=\{a\}$$

这样，可以看到上式的两边的确相等。 □

我们用下面的例子说明式 $R\setminus(S\cap T)=(R\setminus S)\cup(R\setminus T)$ 的正确性。

例 4.41 假设 $R=\{a,b\}$，$S=\{b,c\}$，$T=\{c,d\}$。考虑下面的式子。

$$R\setminus(S\cap T)=(R\setminus S)\cup(R\setminus T)$$

其左边的计算结果如下：

$$R\setminus(S\cap T)=\{a,b\}\setminus(\{b,c\}\cap\{c,d\})$$
$$=\{a,b\}\setminus\{c\}$$
$$=\{a,b\}$$

其右边的计算结果如下：

$$(R\setminus S)\cup(R\setminus T)$$
$$=(\{a,b\}\setminus\{b,c\})\cup(\{a,b\}\setminus\{c,d\})$$
$$=\{a\}\cup\{a,b\}$$
$$=\{a,b\}$$

这样，可以看到上式的两边的确相等。 □

最后，从一个集合 S 中取走元素将得到一个至少与 S 同样小的集合。

规则 4.30 对于任意的集合 S 和 T，有

$S\setminus T\subseteq S$ □

例 4.42

$\{1,2,3\}\setminus\{1,2\}\subseteq\{1,2,3\}$ □

练习 4.20 假设集合 $A=\{1,2,3,4\}$，$B=\{2,4,6,8\}$，$C=\{3,4,5,6\}$，计算下列各题。

1. $A\setminus B$
2. $C\setminus A$
3. $B\setminus C$
4. $B\setminus B$
5. $(A\setminus B)\setminus C$
6. $A\setminus(B\setminus C)$ □

练习 4.21 证明集合的差集 $\setminus$ 不满足交换律。 □

练习 4.22 证明集合的差集 $\setminus$ 不满足结合律。 □

练习 4.23 证明集合的差集 $\setminus$ 不是幂等的。 □

4.10 有关集合的推理

我们已经介绍了集合论的一些基本规则；其中有一些刻画了一个元素在形式为 $S\cap T$ 或 $S\setminus T$ 的集合中的含义。另外，我们还叙述了：当两个集合包含完全相同的元素时这两个集合是相等的；这一陈述可以用下面的规则来刻画。

规则 4.31 给定两个集合 S 和 T，$S=T$ 当且仅当对于任意的元素 x，有

$x\in S\Leftrightarrow x\in T$ □

这样，与前一章所讨论的等值推理的概念相结合，本章所介绍的这些规则为我们提供了判定两个集合是否相等的方法。

假设我们想使用命题逻辑证明 $\cap$ 的交换律是正确的，也就是说，证明对于任意的集合 S 和 T 都有 $S\cap T=T\cap S$。在这种情况下，证明过程如下所示。

$x\in S\cap T$

$\Leftrightarrow x\in S\wedge x\in T$ [规则 4.18]

$\Leftrightarrow x\in T\wedge x\in S$ [规则 3.7]

$\Leftrightarrow x\in T\cap S$ [规则 4.18]

这里，我们假设 x 是任意的元素；由于它是任意的元素，因此可以认为 x 没有任何特殊性，所以，下式对每个元素都成立[○]。

$x\in S\cap T\Leftrightarrow x\in T\cap S$

练习 4.24 证明下列集合论的规则。

○ 在以后章节中我们将对 x 做更细的限制，但是，在此这一解释已经足够。

1. $S \cap S = S$
2. $S \cup S = S$
3. $S \setminus S = \varnothing$
4. $S \cap \varnothing = \varnothing$
5. $S \cup \varnothing = S$
6. $S \setminus \varnothing = S$ □

4.11 集合的势

一个集合 S 的大小，或者说集合的势(cardinality)，是指这个集合所包含的元素的数目。我们把集合 S 的势记作 $\sharp S$。

例如，假设集合 *Friends* 和 *Enemies* 的定义如下。

Friends={佛瑞德，费利西缇}

Enemies={尤恩，伊丽莎白}

这里，很明显集合 *Friends* 包含两个元素，同样集合 *Enemies* 也包含两个元素。这样，可以写作 $\sharp Friends = 2$ 和 $\sharp Enemies = 2$。另外 $\sharp(Friends \cup Enemies) = 4$ 和 $\sharp(Friends \cap Enemies) = 0$。

由上例可知，空集的势为0。

规则 4.32

$\sharp \varnothing = 0$ □

另外，运算符 $\cap$ 和 $\setminus$ 会减少集合的势，而 $\cup$ 会增加集合的势。

规则 4.33 对于任意的集合 S 和 T，有

$\sharp(S \cap T) = \sharp S - \sharp(S \setminus T)$ □

规则 4.34 对于任意的集合 S 和 T，有

$\sharp(S \cup T) = \sharp S + \sharp T - \sharp(S \cap T)$ □

规则 4.35 对于任意的集合 S 和 T，有

$\sharp(S \setminus T) = \sharp S - \sharp(S \cap T)$ □

例 4.43 设两个集合 S 和 T 定义如下。

$S = \{2, 3, 4, 5\}$

$T = \{3, 4, 5, 6\}$

这里，$\sharp S = 4$ 且 $\sharp T = 4$。另外，有

$S \cup T = \{2, 3, 4, 5, 6\}$

$S \cap T = \{3, 4, 5\}$

$S \setminus T = \{2\}$

因此可得如下结果。

$\sharp(S \cup T) = 5$

$\sharp(S \cap T) = 3$

$\sharp(S \setminus T) = 1$

很明显，由于 $3 = 4 - 1$，对此例规则4.33成立。另外，因为 $1 = 4 - 3$，所以对此例规则4.35成立。最后，因为 $5 = 4 + 4 - 3$，所以对此例规则4.34成立。 □

练习 4.25 给出下列集合的势。

1. 单词"hello"中的字母组成的集合。
2. 16与25之间的素数组成的集合。
3. 16的约数组成的集合。
4. $\{1, 2, 3\} \cup \{2, 3, 4\}$
5. $\{1, 2, 3\} \cap \{2, 3, 4\}$
6. $\{1, 2, 3\} \setminus \{2, 3, 4\}$ □

4.12 有穷集合和无穷集合

在练习4.25中，我们看到很多集合的例子，这些集合的元素可以被枚举出来，因此可以进行计数。例如，单词"hello"中的字母组成的集合所包含的元素是 h、e、l 和 o。当我们可以枚举出一个集合中的所有元素，对元素进行计数，并且能够保证计数的过程能够结束时，就说这个集合是有穷的(finite)。例如，16的约数组成的集合是有穷的。相反，如果对于一个给定集合来说这个过程永远不会结束，也就是说，计数元素的过程会永远持续下去，那么我们就说这个集合是无穷的(infinite)。例如，自然数的集合 $\mathbb{N}$ 就是无穷的。

练习 4.26 下列集合中哪些是有穷的，哪些是无穷的？

1. 空集。
2. 整数集合。
3. 奇数的集合。

4. 英文字母表中所有字母组成的集合。

5. 由英文字母组成的长度小于 7 的单词所组成的集合。

6. 英文字母表中的字母组成的所有字符串的集合。

7. 曾经生活在地球上的所有人组成的集合。 □

4.13 集合的幂集

在前面，我们已经介绍过如何考虑任意集合 S 的子集。例如，如果 $T=\{a, b\}$，则 T 的子集为$\varnothing$、$\{a\}$、$\{b\}$和$\{a, b\}$。

这样，我们可以按下面的步骤列举出给定集合的所有子集。首先，考虑势为 0 的子集。在任何时候，势为 0 的子集只有一个，那就是空集。然后，考虑所有势为 1 的子集。对于上面的例子，势为 1 的子集有$\{a\}$和$\{b\}$。接着，考虑所有势为 2 的子集。以此类推，重复这样的步骤直到子集的势等于原来集合的势为止。同样，具有这样的势的子集也只有一个，即原来的集合本身。

按上面的步骤，计算集合$\{1, 2, 3, 4\}$的子集如下。

势为 0 的子集：$\varnothing$

势为 1 的子集：$\{1\}$，$\{2\}$，$\{3\}$，$\{4\}$

势为 2 的子集：$\{1, 2\}$，$\{1, 3\}$，$\{1, 4\}$，$\{2, 3\}$，$\{2, 4\}$，$\{3, 4\}$

势为 3 的子集：$\{1, 2, 3\}$，$\{1, 2, 4\}$，$\{1, 3, 4\}$，$\{2, 3, 4\}$

势为 4 的子集：$\{1, 2, 3, 4\}$

我们很自然地可以想到这些子集构成一个集合。因此，给定集合 S，我们把 S 的所有子集构成的集合或称为 S 的幂集(power set)记作$\mathcal{P}S$[○]。因此，对于上面的例子，我们有

$\mathcal{P}T=\{\varnothing, \{a\}, \{b\}, \{a, b\}\}$

$\mathcal{P}\{1, 2, 3, 4\}=\{\varnothing, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$

例 4.44 4 与 6 间的数的集合的集合是$\mathcal{P}\{4, 5, 6\}$，它等同于

$\{\varnothing, \{4\}, \{5\}, \{6\}, \{4, 5\}, \{4, 6\}, \{5, 6\}, \{4, 5, 6\}\}$ □

例 4.45 在本章前面我们看到，空集是每一个集合(包括它本身)的子集。所以，$\mathcal{P}\varnothing$是包含空集的集合。即

$\mathcal{P}\varnothing=\{\varnothing\}$ □

下面的规则刻画了$\in$与$\subseteq$间的关系。

规则 4.36 对于任意的集合 S 和 T，有

$S\in\mathcal{P}T\Leftrightarrow S\subseteq T$ □

例 4.46 给定集合$\{a, b\}$，那么$\{a\}\in\mathcal{P}\{a, b\}$，而且，等价地，$\{a\}\subseteq\{a, b\}$。 □

另外，因为空集是所有集合的子集，所以，空集是所有集合的幂集的元素。

规则 4.37 对于任意的集合 S，有

$\varnothing\in\mathcal{P}S$ □

最后，因为每个集合都是它自身的子集，所以每一个集合都是它的幂集的元素。

规则 4.38 对于任意的集合 S，有

$S\in\mathcal{P}S$ □

练习 4.27 计算下列各幂集。

1. $\mathcal{P}\{x\}$
2. $\mathcal{P}\{x, y\}$
3. $\mathcal{P}\{x, y, z\}$ □

练习 4.28 集合$\{4, 5, 6\}$的幂集包含 8 个元素。问集合$\{4, 5\}$的幂集包含多少个元素？一般地，如果一个集合里有 n 个元素，那么它的幂集包含多少个元素？ □

规则 4.39 对于任意的集合 S，有

$\#(\mathcal{P}S)=2^{\#(S)}$ □

上面这个规则成立的原因是，对于集合

○ 注意，在某些教科书中，集合 S 的幂集记作 $P(S)$或 2^S，而不是$\mathcal{P}S$。

S的任意一个子集T，以及任意一个元素$s \in S$，s有两种可能性：或者s在T中或者s不在T中。例如，假设S中有两个元素a和b，那么，一共有$2 \times 2 = 4$种可能性(或者说可能的子集)：a和b都在子集中(子集为$\{a, b\}$)；a在而b不在子集中(子集为$\{a\}$)；a不在而b在子集中(子集为$\{b\}$)；a和b都不在子集中(子集为$\varnothing$)。这个论断可以扩展到任意(有限)大小的集合。

练习4.29 计算下列各幂集。

1. $\mathbb{P}\{1, 2\}$
2. $\mathbb{P}\{\varnothing\}$
3. $\mathbb{P}\{\varnothing, \{1, 2\}\}$
4. $\mathbb{P}\mathbb{P}\{1, 2\}$ □

在考虑上面练习中的幂集时，必须注意的是我们构造的是以集合为元素的集合。尽管初看起来这些集合很复杂，但是，积累了经验以后，构造幂集其实是轻而易举的。

考虑集合$\{a\}$，这个集合只包含一个元素a。因此，它有两个子集：$\varnothing$和$\{a\}$。所以，这个集合的幂集为

$\{\varnothing, \{a\}\}$

现在，考虑集合$\{\varnothing\}$。同样，这个集合包含一个元素$\varnothing$。所以，它有两个子集：$\varnothing$和$\{\varnothing\}$，因此，这个集合的幂集为

$\{\varnothing, \{\varnothing\}\}$

对于上面两种情况，我们求集合的幂集的方法是相同的，所求得的幂集的唯一区别是单集包含的元素不同：在第一个幂集中是元素a；而在第二个幂集中是空集$\varnothing$。

4.14 集合的广义运算

我们已经看到，给定两个集合S和T，$S \cup T$表示两个集合的并集。另外还看到，$R \cup S \cup T$表示集合R、S和T的并集。当然，我们还可以进一步讨论$Q \cup R \cup S \cup T$或者$P \cup Q \cup R \cup S \cup T$。当然，当集合的数目变得相当大的时候，用这种方法来表示集合的并集会很麻烦。

广义并(generalised union)运算符为我们提供了一种表示多个集合的并集的巧妙方法。给定一组集合，这些集合的广义并是包含所有出现在至少一个集合中的元素的集合。

例4.47 给定集合R、S和T，其中$R = \{1, 2, 3\}$，$S = \{1, 4\}$，$T = \{1, 5\}$，那么这些集合的广义并为$\{1, 2, 3, 4, 5\}$，当然它等同于$R \cup S \cup T$。 □

我们将集合R、S和T的广义并记作$\bigcup \{R, S, T\}$。正如所看到的那样，这个集合等价于$R \cup S \cup T$，但是使用广义并的运算符来表示这个集合具有更简洁的优势。

规则4.40 *对于任意集合的集合X以及任意元素x，$x \in \bigcup X$当且仅当存在集合$S \in X$使得$x \in S$。* □

练习4.30 假设有如下集合。

$X = \{$杰克，吉尔$\}$

$Y = \{$杰克，理查德$\}$

$Z = \{$玛格丽特，艾马$\}$

计算下列广义并。

1. $\bigcup \{X, Y\}$
2. $\bigcup \{X\}$
3. $\bigcup \{X, Y, Z\}$
4. $\bigcup \{X, X\}$
5. $\bigcup \{X, \varnothing\}$ □

集合的广义交(generalised intersection)运算符的作用与广义并运算符相似。给定一组集合，这些集合的广义交是包含所有出现在每一个集合中的元素的集合。

例4.48 给定集合R、S和T，其中$R = \{1, 2, 3\}$，$S = \{1, 4\}$，$T = \{1, 5\}$，那么这些集合的广义交为$\{1\}$，当然它等同于$R \cap S \cap T$。 □

我们将集合R、S和T的广义交记作$\bigcap \{R, S, T\}$。

规则4.41 *对于任意集合的集合X以及任意元素x，$x \in \bigcap X$当且仅当对于每一*

个集合 $S \in X$ 都有 $x \in S$。 □

练习 4.31 假设有如下集合。

$X=\{$杰克，吉尔$\}$

$Y=\{$杰克，理查德$\}$

$Z=\{$玛格丽特，艾马$\}$

计算下列广义交。

1. $\bigcap\{X, Y\}$
2. $\bigcap\{X\}$
3. $\bigcap\{X, Y, Z\}$
4. $\bigcap\{X, X\}$
5. $\bigcap\{X, \varnothing\}$ □

4.15 附加练习

练习 4.32 使用外延法定义下列集合。

1. 7 与 13 间(包括 7 和 13)的所有自然数组成的集合。

2. 7 与 13 间(包括 7 和 13)的所有奇数组成的集合。

3. 所有满足下面等式的自然数组成的集合。

$7+(2\times x)=13$

4. 所有满足下面不等式的自然数组成的集合。

$7+(2\times x)<13$

练习 4.33 下列集合中哪些相等?

$A=\{a, b\}$

$B=\{b, a\}$

$C=\{a, b, c\}$

$D=\{c, a, b\}$

$E=\{a, b, a, b\}$

$F=\{a, b, c, a\}$

练习 4.34 下列集合(如果存在的话)中哪些相等：$\varnothing$、$\{0\}$和$\{\varnothing\}$?

练习 4.35 下列集合中哪些是单集?

1. $\varnothing$
2. $\{0\}$
3. $\{\varnothing\}$

练习 4.36 给出下列集合的幂集。

1. $\varnothing$
2. $\{$吉恩，理查德$\}$
3. $\{\{$吉恩$\}, \{$理查德$\}\}$
4. $\{\{$吉恩，理查德$\}\}$

练习 4.37 假设 $S=\{r, s, t\}$，下面哪些陈述是有定义的?

1. $r \in S$
2. $r \subseteq S$
3. $\{r\} \in S$
4. $\{r\} \subseteq S$

练习 4.38 考虑下列集合。

$R=\{$杰克$\}$

$S=\{$吉尔$\}$

$T=\{$杰克，吉尔$\}$

下面的陈述中哪些为真，哪些为假?

1. $\varnothing \subset R$
2. $S \supseteq \varnothing$
3. $R \subseteq S$
4. $T \supset T$
5. $T \not\subset R$
6. $\varnothing \not\supseteq S$
7. $R \not\subseteq \varnothing$
8. $S \not\supset T$

练习 4.39 证明下列陈述。

1. $\{1, 2, 3\} \not\subseteq \{2, 3, 4\}$
2. $\{1, 2, 3\} \subset \{1, 2, 3, 4\}$

练习 4.40 画出下列集合的文氏图。

1. $(R \cap S) \cap T$
2. $(R \cap S) \setminus T$
3. $(R \setminus S) \cup T$
4. $(R \cap S) \cup T$

练习 4.41 假设有如下集合。

$A=\{1, 2, 3, 4, 5\}$

$B=\{3, 4, 5, 6, 7\}$

$C=\{2, 4, 6\}$

$D=\{1, 3, 5, 7\}$

计算下列集合。

1. $(A \cap C) \setminus (B \cap D)$
2. $(A \cup D) \setminus (B \cup C)$

3. $\bigcap\{A, B, D\}$

4. $(\bigcup\{A, D\})\bigcap(\bigcup\{B, D\})$

练习 4.42 给定练习 4.41 的集合，计算下列各式。

1. $A\setminus B$

2. $A\setminus(B\setminus C)$

3. $(A\setminus B)\setminus C$

4. $(A\setminus B)\setminus(C\setminus D)$

练习 4.43 给定集合 $A=\{1, 2, 3, 4\}$和下面的条件，计算集合 B 和 C。

$A=B\bigcap C$

$C\setminus A=\{5, 6\}$

$(B\bigcup A)\setminus C=\{7\}$

练习 4.44 证明下面的陈述。

$(B\bigcup C)\bigcap A=(B\bigcap A)\bigcup(C\bigcap A)$

练习 4.45 证明下面的陈述。

$(B\bigcap C)\bigcup A=(B\bigcup A)\bigcap(C\bigcup A)$

练习 4.46 下列集合各包含多少个元素？

1. $\{a, b\}$

2. $\mathcal{P}\{a, b\}$

3. $\mathcal{P}\mathcal{P}\{a, b\}$

4. $\varnothing$

5. $\mathcal{P}\varnothing$

6. $\mathcal{P}\mathcal{P}\varnothing$

练习 4.47 列举出下列集合的元素。

1. $\mathcal{P}\{a, b\}$

2. $\mathcal{P}\varnothing$

3. $\mathcal{P}\mathcal{P}\varnothing$

4. $\mathcal{P}\{\varnothing, \{a, b\}\}$

练习 4.48 下列集合中哪些是有穷集合，哪些是无穷集合？

1. 一年里所有的月份组成的集合。

2. 将 1 到 1 000 000 间的数字排序，所有排序方法组成的集合。

3. 所有能被 7 整除的整数的集合。

4. 所有包含数字 0 的自然数的集合。

练习 4.49 假设集合 $A=\{1, 2\}$，$B=\{2, 3\}$，计算下列各集合。

1. $\mathcal{P}A$

2. $A\bigcap B$

3. $A\bigcup B$

4. $A\setminus B$

5. $\mathcal{P}((A\setminus B)\bigcap B)$

练习 4.50 证明，对于任意的集合 A 和 B，有

$(A\setminus B)\bigcap B=\varnothing$

练习 4.51 证明，对于任意的集合 A 和 B，有

$(A\setminus B)\bigcup B=A\bigcup B$

练习 4.52 计算下列各集合。

1. $\bigcap\{\varnothing\}$

2. $\bigcup\{\varnothing\}$

3. $\bigcap\{\{1\}, \{3\}, \{5\}\}$

4. $\bigcup\{\{1\}, \{3\}, \{5\}\}$

5. $\bigcap\{\{1, 3\}, \{3, 5\}\}$

6. $\bigcup\{\{1, 3\}, \{3, 5\}\}$

练习 4.53 计算下列集合的广义并。

1. {约翰，格雷厄姆}、{安迪，约翰}和{理查德，约翰}

2. {1}和{1，2}

3. {1}和$\varnothing$

4. {{1}}和{{2}}

练习 4.54 计算练习 4.53 中各集合的广义交。

练习 4.55 在什么情况下$\bigcup S=\bigcap S$?

4.16 练习解答

4.1

1. 这些元素是：a，e，i，o，u。

2. 这个集合没有任何元素。

3. 这些元素是 t，h，e，s，o，f，l，r。

4.2

1. $\{0, 1, 2, 3, 4, 5\}$

2. $\{1, 3, 5\}$

3. $\{e, o\}$

4.3 它们都是相等的。

4.4 只有 1 是单集。2 中没有元素。而 3 包含 3 个元素：2、4 和 8。

4.5 1 和 3 等同于空集，而 2 不是。

4.6 1 和 2 为假，3 和 4 为真。

4.7 1 为真，2 为假，3 和 4 未定义。

4.8 1、2 和 3 为真，4 和 5 为假。

4.9 仅第三个陈述为真。

4.10 前两个陈述为真，后三个陈述为假。

4.11 前两个陈述为假，后三个陈述为真。

4.12 1、2 和 4 为真，3 为假。

4.13 $\varnothing$，$\{1\}$，$\{2\}$，$\{3\}$，$\{1, 2\}$，$\{1, 3\}$，$\{2, 3\}$，$\{1, 2, 3\}$。

4.14 $\varnothing$，$\{1\}$，$\{2\}$，$\{3\}$，$\{1, 2\}$，$\{1, 3\}$，$\{2, 3\}$。

4.15 根据规则 4.4，$\varnothing$ 是任意集合的子集，所以 $\varnothing$ 是 S 的子集。根据规则 4.3，因为 S 是 $\varnothing$ 的子集，这两个集合一定相等。

4.16 1 和 2 未定义，3 和 4 为真。

4.17 前三个为假，后三个为真。

4.18

1. $\{1, 2, 3\}$
2. $\{1, 2, 3\}$
3. $\{1, 2, 3, 4\}$
4. $\{1, 2, 3, 4\}$
5. $\{1, 2, 3, 4\}$

4.19

1. $\varnothing$
2. $\{1, 2, 3\}$
3. $\varnothing$
4. $\{1, 2, 3\}$
5. $\varnothing$

4.20

1. $\{1, 3\}$
2. $\{5, 6\}$
3. $\{2, 8\}$
4. $\varnothing$
5. $\{1\}$
6. $\{1, 3, 4\}$

4.21 假设“$\setminus$”满足交换律，即对于任意的集合 S 和 T，有 $S \setminus T = T \setminus S$。取集合 $S=\{0, 1\}$，$T=\{0\}$，那么有 $S \setminus T = \{1\}$ 而 $T \setminus S = \varnothing$。所以，“$\setminus$”不满足交换律。

4.22 假设“$\setminus$”满足结合律，即对于任意的集合 R、S 和 T，有 $R \setminus (S \setminus T) = (R \setminus S) \setminus T$。取 $R=\{1, 2, 3\}$，$S=\{1, 2\}$，$T=\{3\}$。那么有

$$\begin{aligned} & R \setminus (S \setminus T) \\ = & \{1,2,3\} \setminus (\{1,2\} \setminus \{3\}) \\ = & \{1,2,3\} \setminus \{1,2\} = \{3\} \end{aligned}$$

而

$$\begin{aligned} & (R \setminus S) \setminus T \\ = & (\{1,2,3\} \setminus \{1,2\}) \setminus \{3\} \\ = & \{3\} \setminus \{3\} = \varnothing \end{aligned}$$

所以，“$\setminus$”不满足结合律。

4.23 假设“$\setminus$”是幂等的，即对于任意的集合 S，有 $S \setminus S = S$。取 $S=\{1, 2\}$，那么有

$$\begin{aligned} S \setminus S &= \{1, 2\} \setminus \{1, 2\} \\ &= \varnothing \end{aligned}$$

所以，“$\setminus$”不是幂等的。

4.24

1. $x \in S \cap S$
   $\Leftrightarrow x \in S \wedge x \in S$ [规则 4.18]
   $\Leftrightarrow x \in S$ [规则 3.7]
2. $x \in S \cup S$
   $\Leftrightarrow x \in S \vee x \in S$ [规则 4.12]
   $\Leftrightarrow x \in S$ [规则 3.14]
3. $x \in S \setminus S$
   $\Leftrightarrow x \in S \wedge x \notin S$ [规则 4.25]
   $\Leftrightarrow x \in S \wedge -(x \in S)$ [规则 4.1]
   $\Leftrightarrow$ false [规则 3.6]
   $\Leftrightarrow x \in \varnothing$ [规则 4.2]
4. $x \in S \cap \varnothing$
   $\Leftrightarrow x \in S \wedge x \in \varnothing$ [规则 4.18]
   $\Leftrightarrow x \in S \wedge$ false [规则 4.2]
   $\Leftrightarrow$ false [规则 3.5]
   $\Leftrightarrow x \in \varnothing$ [规则 4.2]
5. $x \in S \cup \varnothing$

$\Leftrightarrow x \in S \vee x \in \varnothing$ [规则 4.12]

$\Leftrightarrow x \in S \vee \text{false}$ [规则 4.2]

$\Leftrightarrow x \in S$ [规则 3.11]

6. $x \in S \setminus \varnothing$

$\Leftrightarrow x \in S \wedge x \notin \varnothing$ [规则 4.25]

$\Leftrightarrow x \in S \wedge \neg(x \in \varnothing)$ [规则 4.1]

$\Leftrightarrow x \in S \wedge \neg \text{false}$ [规则 4.2]

$\Leftrightarrow x \in S \wedge \text{true}$ [规则 3.1]

$\Leftrightarrow x \in S$ [规则 3.4]

4.25

1. $\#\{h, e, l, o\}=4$
2. $\#\{17, 19, 23\}=3$
3. $\#\{1, 2, 4, 8, 16\}=5$
4. $\#\{1, 2, 3, 4\}=4$
5. $\#\{2, 3\}=2$
6. $\#\{1\}=1$

4.26 2、3 和 6 是无穷的，其余的是有穷的。

4.27

1. $\{\varnothing, \{x\}\}$
2. $\{\varnothing, \{x\}, \{y\}, \{x, y\}\}$
3. $\{\varnothing, \{x\}, \{y\}, \{z\}, \{x, y\}, \{x, z\}, \{y, z\}, \{x, y, z\}\}$

4.28 集合$\{4, 5\}$的幂集有四个元素：$\varnothing$，$\{4\}$，$\{5\}$，$\{4, 5\}$。一般地，如果集合 S 有 n 个元素，那么$\mathcal{P}S$ 有 2^n 个元素。

4.29

1. $\{\varnothing, \{1\}, \{2\}, \{1, 2\}\}$
2. $\{\varnothing, \{\varnothing\}\}$
3. $\{\varnothing, \{\varnothing\}, \{\{1, 2\}\}, \{\varnothing, \{1, 2\}\}\}$
4. $\{\varnothing, \{\varnothing\}, \{\{1\}\}, \{\{2\}\}, \{\{1, 2\}\}, \{\varnothing, \{1\}\}, \{\varnothing, \{2\}\}, \{\varnothing, \{1, 2\}\}, \{\{1\}, \{2\}\}, \{\{1\}, \{1, 2\}\}, \{\{2\}, \{1, 2\}\}, \{\varnothing, \{1\}, \{2\}\}, \{\varnothing, \{1\}, \{1, 2\}\}, \{\varnothing, \{2\}, \{1, 2\}\}, \{\{1\}, \{2\}, \{1, 2\}\}, \{\varnothing, \{1\}, \{2\}, \{1, 2\}\}\}$

4.30

1. {杰克，吉尔，理查德}
2. {杰克，吉尔}
3. {杰克，吉尔，理查德，玛格丽特，艾马}
4. {杰克，吉尔}
5. {杰克，吉尔}

4.31

1. {杰克}
2. {杰克，吉尔}
3. $\varnothing$
4. {杰克，吉尔}
5. $\varnothing$

4.32

1. $\{7, 8, 9, 10, 11, 12, 13\}$
2. $\{7, 9, 11, 13\}$
3. $\{3\}$
4. $\{0, 1, 2\}$

4.33 $A=B=E$ 且 $C=D=F$。

4.34 它们都是互不相等的。

4.35 第二和第三个集合是单集。

4.36

1. $\{\varnothing\}$
2. $\{\varnothing$，{吉恩}，{理查德}，{吉恩，理查德}$\}$
3. $\{\varnothing$，{{吉恩}}，{{理查德}}，{{吉恩}，{理查德}}$\}$
4. $\{\varnothing$，{{吉恩，理查德}}$\}$

4.37 1 和 4 有定义；2 和 3 未定义。

4.38 除了 3 和 4 外都为真。

4.39

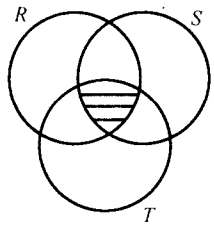
1. 为证$\{1, 2, 3\} \not\subseteq \{2, 3, 4\}$为真，需要找到这样一个元素，它在前一个集合中出现而在后一个集合中不出现。1 就是这样的一个元素。所以，$\{1, 2, 3\} \not\subseteq \{2, 3, 4\}$为真。

2. 为证$\{1, 2, 3\} \subset \{1, 2, 3, 4\}$为真，需要说明：在前一个集合中出现的每一个元素都在后一个集合中出现，而且，至少有一个元素在后一个集合中出现而不在前一个集合中出现。集合$\{1, 2, 3\}$中的每一个元素都在$\{1, 2, 3, 4\}$中出现。另外，元素

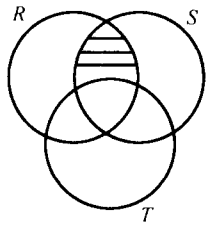
4 在后一个集合中出现，但不在前一个集合中出现。所以，$\{1, 2, 3\} \subset \{1, 2, 3, 4\}$ 为真。

4.40

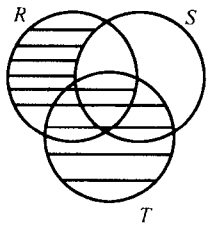
1.



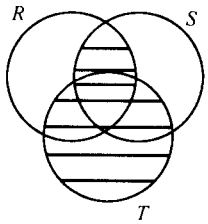
2.



3.



4.



4.41

1. $\{2, 4\}$
2. $\{1\}$
3. $\{3, 5\}$
4. $\{1, 3, 4, 5, 7\}$

4.42

1. $\{1, 2\}$
2. $\{1, 2, 4\}$
3. $\{1\}$
4. $\{1\}$

4.43

$B=\{1, 2, 3, 4, 7\}$

和

$C=\{1, 2, 3, 4, 5, 6\}$

4.44

$x \in (B \cup C) \cap A$

$\Leftrightarrow (x \in B \cup C) \wedge x \in A$ [规则 4.18]

$\Leftrightarrow (x \in B \vee x \in C) \wedge x \in A$ [规则 4.12]

$\Leftrightarrow (x \in B \wedge x \in A) \vee (x \in C \wedge x \in A)$ [规则 3.9]

$\Leftrightarrow (x \in B \cap A) \vee (x \in C \cap A)$ [规则 4.18]

$\Leftrightarrow x \in (B \cap A) \cup (C \cap A)$ [规则 4.12]

4.45

$x \in (B \cap C) \cup A$

$\Leftrightarrow (x \in B \cap C) \vee x \in A$ [规则 4.12]

$\Leftrightarrow (x \in B \wedge x \in C) \vee x \in A$ [规则 4.18]

$\Leftrightarrow (x \in B \vee x \in A) \wedge (x \in C \vee x \in A)$ [规则 3.9]

$\Leftrightarrow (x \in B \cup A) \wedge (x \in C \cup A)$ [规则 4.12]

$\Leftrightarrow x \in (B \cup A) \cap (C \cup A)$ [规则 4.18]

4.46

1. 2
2. $2^2=4$
3. $2^{2^2}=2^4=16$
4. 0
5. $2^0=1$
6. $2^{2^0}=2^1=2$

4.47

1. $\{\varnothing, \{a\}, \{b\}, \{a, b\}\}$
2. $\{\varnothing\}$
3. $\{\varnothing, \{\varnothing\}\}$
4. $\{\varnothing, \{\varnothing\}, \{\{a, b\}\}, \{\varnothing, \{a, b\}\}\}$

4.48 除 4 外都是有穷的。

4.49

1. $\{\varnothing, \{1\}, \{2\}, \{1, 2\}\}$
2. $\{2\}$
3. $\{1, 2, 3\}$
4. $\{1\}$
5. $\{\varnothing\}$

4.50

$x \in (A \setminus B) \cap B$

$\Leftrightarrow (x \in A \setminus B) \wedge x \in B$ [规则 4.18]

$\Leftrightarrow (x \in A \wedge x \notin B) \wedge x \in B$ [规则 4.25]

$\Leftrightarrow x \in A \wedge \neg(x \in B) \wedge x \in B$ [规则 4.1]

$\Leftrightarrow x \in A \wedge \text{false}$ [规则 3.6]

$\Leftrightarrow \text{false}$ [规则 3.5]

$\Leftrightarrow x \in \varnothing$ [规则 4.2]

4.51

$x \in (A \setminus B) \cup B$

$\Leftrightarrow x \in (A \setminus B) \vee x \in B$ [规则 4.12]

$\Leftrightarrow (x \in A \wedge x \notin B) \vee x \in B$ [规则 4.25]

$\Leftrightarrow (x \in A \vee x \in B) \wedge (x \notin B \vee x \in B)$ [规则 3.9]

$\Leftrightarrow (x \in A \vee x \in B) \wedge (\neg(x \in B) \vee x \in B)$ [规则 4.1]

$\Leftrightarrow (x \in A \vee x \in B) \wedge \text{true}$ [规则 3.15]

$\Leftrightarrow x \in A \vee x \in B$ [规则 3.4]

$\Leftrightarrow x \in A \cup B$ [规则 4.12]

4.52

1. $\varnothing$
2. $\varnothing$
3. $\varnothing$
4. $\{1, 3, 5\}$
5. $\{3\}$
6. $\{1, 3, 5\}$

4.53

1. {约翰，格雷厄姆，安迪，理查德}
2. $\{1, 2\}$
3. $\{1\}$
4. $\{\{1\}, \{2\}\}$

4.54

1. {约翰}
2. $\{1\}$
3. $\varnothing$
4. $\varnothing$

4.55 当集合 S 为空或恰好包含一个非空集合时。

第5章 布尔代数

5.1 简介

集合论和命题逻辑具有相似的性质。这些性质可以用于定义被称为布尔代数(Boolean algebra)[○]的数学结构。例如，集合论和命题逻辑中的幂等律极为相似，如下表所示。

集合论	命题逻辑
$S \cup S = S$	$p \vee p = p$
$S \cap S = S$	$p \wedge p = p$

由下表可知，集合论和命题逻辑的交换律也很相似。

集合论	命题逻辑
$S \cup T = T \cup S$	$p \vee q = q \vee p$
$S \cap T = T \cap S$	$p \wedge q = q \wedge p$

练习 5.1 给出集合论和命题逻辑的分配律。 □

实际上，可以统一处理集合论和命题逻辑的相同规则，这样就可以将一个研究领域的研究结果应用于另一个领域。除此之外，布尔代数的概念是计算机科学的一个核心领域(数字电路设计)的基础。我们将在第 14 章讨论数字电路的设计，[Gre98]对此进行了深入的探讨。

本章着重于构成布尔代数的规则。在学习布尔代数之前，我们先回顾命题逻辑和集合论的相关内容。

希望得到布尔代数的完整介绍的读者可以参见[Mon89]。

5.2 命题逻辑回顾

回忆一下，每个命题逻辑都与一个真值(true 或 false)相关联。而且，我们在前面已经看到，可以通过等值关系"$\Leftrightarrow$"把命题分为三类：与 true 逻辑等值的命题(重言式集合)，与 false 逻辑等值的命题(矛盾式集合)，既不与 true 也不与 false 逻辑等值的命题(不定式集合)。可以用 T 表示重言式集合，用 F 表示矛盾式集合。

另外，命题逻辑包含三个基本运算符：$\neg$、$\wedge$ 和 $\vee$(回想一下，可以使用这些运算符定义$\Rightarrow$和$\Leftrightarrow$)，还包含命题逻辑满足的若干规则，如 $\wedge$ 和 $\vee$ 的交换律、分配律等。

总之，命题逻辑由类 T 和 F，运算符 $\neg$、$\wedge$ 和 $\vee$，以及命题逻辑的若干规则组成。

5.3 集合论回顾

考虑集合 S 及其幂集 $\mathcal{P}S$。集合论的运算符 $\cap$ 和 $\cup$ 可以看成是 $\mathcal{P}S$ 上的运算，因为给定 $\mathcal{P}S$ 中的任意两个元素，它们的并和交也是 $\mathcal{P}S$ 的元素。

现在，考虑某个集合 $X \in \mathcal{P}S$，总有 $X \cap S = X$。这一事实可由下面的规则刻画。

规则 5.1 对于任意的集合 $X \in \mathcal{P}S$，有 $X \cap S = X$ □

例 5.1 考虑集合 $S=\{1, 2, 3\}$。这里有 $\{1, 2\} \cap \{1, 2, 3\} = \{1, 2\}$ □

可以如下定义 $\mathcal{P}S$ 上的取补运算符(complement operator)。给定任意的集合

○ 以 George Boole(1815—1864)命名。感兴趣的读者可以参考[GGB97]，该书包括了布尔的一系列研究工作。

$X \in \mathbb{P}S$，X 的补，记作 X^{-}，定义为 $S \setminus X$。

规则 5.2 对于任意的集合 $X \in \mathbb{P}S$，有

$X^{-} = S \setminus X$ □

例 5.2 还是考虑集合 $S = \{1, 2, 3\}$。这里有

$\{1, 2\}^{-} = \{3\}$ □

练习 5.2 设集合 $S = \{a, b, c, d, e\}$，计算下列集合。

1. $\{a, d\}^{-}$
2. $\{a, b, c, d\}^{-}$
3. $\{a\}^{-} \cap \{b\}^{-}$
4. $\varnothing^{-}$
5. $\{a, b, c, d, e\}^{-}$ □

定义了取补运算符之后，可以引入下面两个规则。

规则 5.3 对于任意的集合 $X \in \mathbb{P}S$，有

$X \cup X^{-} = S$ □

规则 5.4 对于任意的集合 $X \in \mathbb{P}S$，有

$X \cap X^{-} = \varnothing$ □

总之，集合论由集合 $\mathbb{P}S$，$\mathbb{P}S$ 上的运算符 $\cap$、$\cup$ 和 $-$，以及若干集合论的规则组成。

因此，可以认为集合论和命题逻辑的抽象结构极其相似。通过学习布尔代数，我们把重点放在决定这个一般结构的规则，即决定命题逻辑和集合论的规则上，而不专门针对命题逻辑或集合论。因此，下面讨论的规则同时适用于命题逻辑和集合论，当然，也适用于满足布尔代数规则的其他结构。

练习 5.3 使用等值推理证明规则 5.1。 □

练习 5.4 使用等值推理证明规则 5.3。 □

练习 5.5 使用等值推理证明规则 5.4。 □

5.4 布尔代数基础

布尔代数描述一个同时包含命题逻辑和集合论的规则的数学结构。这一结构的定义如下。

定义 5.1 考虑集合 B 以及定义在 B 上的二元运算符"+"和"*"及一元运算符"′"。又设 0 和 1 是 B 上两个互不相同的元素。六元组

$\langle B, +, *, ', 0, 1 \rangle$

称为布尔代数，当且仅当对于集合 B 中的任意元素 a、b 和 c，下面四个公理成立。

交换律：$a+b=b+a$

$a*b=b*a$

分配律：$a+(b*c)=(a+b)*(a+c)$

$a*(b+c)=(a*b)+(a*c)$

同一律：$a+0=a$

$a*1=a$

补律：$a+a'=1$

$a*a'=0$ □

也就是说，任何满足上面规则的结构都可看成是布尔代数。

例 5.3 包含两个元素 0 和 1，运算"′"、"+"和"*"的布尔代数 B 的定义如下。

$0'=1$

$1'=0$

$0+0=0$

$0+1=1$

$1+0=1$

$1+1=1$

$0*0=0$

$0*1=0$

$1*0=0$

$1*1=1$

为使这些元素和运算组成布尔代数，必须满足交换律、分配律、同一律和补律。 □

现在，我们考虑布尔代数的一般结构。为此，使用如下术语。

首先，元素 0 称为零元素(zero element)。之所以如此称呼是因为它与算术中的 0 相似。例如，$n*0=0$ 对布尔代数和算术都

成立。

其次，元素 1 称为单位元素（unit element）。同样，之所以如此称呼是因为它与算术中的 1 相似。例如，$n*1=n$ 对布尔代数和算术都成立。

然后，a'称为 a 的补（complement）。例如，$0'=1$，$1'=0$。

最后，运算"$*$"与运算"$+$"所产生的结果分别称为积（product）与和（sum）。

从上面的规则可以推导出其他一些规则；其中有一些是我们所熟悉的。

首先，积与和运算符都满足结合律。

规则 5.5 对于任意的元素 a、b 和 c，有

$a+(b+c)=(a+b)+c$

$a*(b*c)=(a*b)*c$ □

其次，a 的补的补与 a 相等。

规则 5.6 对于任意的元素 a，有

$a''=a$ □

这与第 3 章的等值式$(\neg\neg p)\Leftrightarrow p$ 类似。

下一个规则表明和与积都是幂等的。

规则 5.7 对于任意的元素 a，有

$a+a=a$

$a*a=a$ □

规则 5.8 告诉我们，德·摩根律不仅对命题逻辑和集合论成立，而且对所有布尔代数均成立。

规则 5.8 对于任意的元素 a 和 b，有

$(a+b)'=a'*b'$

$(a*b)'=a'+b'$ □

正如任意命题与 true 的析取都为 true，任意命题与 false 的合取都为 false 那样，任意的布尔代数项与 1 的和都为 1，任意的布尔代数项与 0 的积都为 0。

规则 5.9 对于任意的元素 a，有

$a+1=1$

$a*0=0$ □

下面的规则可由命题逻辑和集合论的相应规则推演出来。

首先，考虑两个原子命题 p 和 q。命题 $p\vee(p\wedge q)$为真，当且仅当其中至少一个析取项为真。然而，两个析取项都需要 p 为真才能为真。由此，

$p\vee(p\wedge q)\Leftrightarrow p$

是命题逻辑定理。

下面考虑两个集合 S 和 T。S 和 T 的并 $S\cup T$ 包含 S 和 T 的所有元素，S 与 $S\cup T$ 的交包含 S 的所有元素。由此，

$S\cap(S\cup T)=S$

对所有集合 S 和 T 成立。

这两个事实对所有布尔代数都成立，正如下面的规则所述。

规则 5.10 对于任意的元素 a 和 b，有

$a+(a*b)=a$

$a*(a+b)=a$ □

最后，由命题逻辑的逻辑等值式$(\neg\ \text{false})\Leftrightarrow\text{true}$ 及$(\neg\ \text{true})\Leftrightarrow\text{false}$ 可以导出下面的规则。

规则 5.11

$0'=1$

$1'=0$ □

练习 5.6 使用布尔代数的公理及规则 5.6 和规则 5.8 证明 $0'=1$ 成立。 □

练习 5.7 使用布尔代数的公理证明，对于任意的元素 a，

$a+0=a*1$

成立。 □

练习 5.8 使用布尔代数的公理证明，对于任意的元素 a，

$a+a=a$

成立。 □

练习 5.9 使用练习 5.8 的结果和规则 5.5 以及布尔代数的公理证明，对任意的元素 a，

$a+1=1$

成立。 □

5.5 简写规定

我们可以扔掉乘法符号，使用并置来进行缩写。即可以用 ab 代替 $a*b$，用 abc 代替 $a*b*c$。

例 5.4 可以用如下形式表示布尔代数的分配律。

$a(b+c)=(ab)+(ac)$

$a+(bc)=(a+b)(a+c)$ □

练习 5.10 利用简写规定缩写下列各式。

1. $a*b$
2. $a*(b+c)$
3. $a+(b*c)$
4. $a*b'$
5. $(a*b)'$ □

5.6 优先级

正如我们对命题逻辑的运算符强加了优先级顺序一样，对布尔代数的运算符也赋予优先级顺序。这里，“′”的优先级高于“*”，“*”的优先级高于“+”。也就是说，“′”的绑定最强，“+”的绑定最弱。

例 5.5 $a+b*c$ 的意思是 $a+(b*c)$ 而非 $(a+b)*c$。 □

例 5.6 $a*b'$ 的意思是 $a*(b')$ 而非 $(a*b)'$。 □

练习 5.11 以完整的括号形式书写下面的表达式。

1. ab'
2. $ab+c$
3. $a+b'$
4. $ab+cd+ef'$ □

5.7 集合的布尔代数

现在，我们考虑关于集合的布尔代数。

设有集合 S 及其幂集 $\mathcal{P}S$，集合 $\mathcal{P}S$ 及其上的并、交和取补运算符构成一个布尔代数。在这一布尔代数中，空集 $\varnothing$ 是零元素，集合 S 是单位元素。

练习 5.12 证明 $\mathcal{P}S$ 是布尔代数。 □

5.8 命题的布尔代数

设 Π 为命题的集合，那么，集合 Π 及其上的命题运算符 $\vee$、$\wedge$ 和 $\neg$ 构成一个布尔代数。在这一布尔代数中，false 是零元素，true 是单位元素。

练习 5.13 证明 Π 是布尔代数。 □

5.9 布尔代数的同构

对于两个布尔代数 B 和 $\overline{B}$，如果存在将 B 的元素转换为 $\overline{B}$ 的元素的运算 f，则称两个布尔代数同构(isomorphic)○。

本质上，这样的运算把 B 的零元素和单位元素分别转换成 $\overline{B}$ 的零元素和单位元素，并且“保持”三个运算，即保证对任意的元素 a、$b\in B$，下面的关系成立。

$f(a+b)=f(a)\overline{+}f(b)$

$f(a*b)=f(a)\overline{*}f(b)$

$f(a')=f(a)'$

其中，$\overline{+}$ 和 $\overline{*}$ 分别表示 $\overline{B}$ 的和运算符与积运算符。

例 5.7 我们可以如下证明集合论的布尔代数与命题逻辑的布尔代数同构：定义运算 g，使得 $g(\text{true})=S$、$g(\text{false})=\varnothing$，而

○ 我们将在第9章给出函数的形式定义，那时就可以给出两个布尔代数同构的明确意义。

因此，本节及相关练习解答并没有给出严格的论述和证明，需要理解论述和证明是示意性的。5.8节的讨论也不够详细。这一部分的内容与后面章节的内容没有关联。——译者注

且对所有的命题 a 和 b，有

$$g(a \vee b) = g(a) \cup g(b)$$
$$g(a \wedge b) = g(a) \cap g(b)$$
$$g(\neg a) = g(a)^{-}$$ □

练习 5.14 将例 5.7 的运算应用于下面各命题。

1. true ∨ false
2. ¬(true ∧ (true ∨ false))
3. (true ∧ (true ∨ false)) ∨ false □

练习 5.15 给定如下定义的运算 h。

$h(\varnothing) = \text{false}$

$h(S) = \text{true}$

$h(a \cup b) = h(a) \vee h(b)$

$h(a \cap b) = h(a) \wedge h(b)$

$h(a^{-}) = \neg h(a)$

将 h 应用于以下各项。

1. $\varnothing^{-}$
2. $\varnothing \cap S$
3. $\varnothing \cap (S \cup \varnothing)$ □

练习 5.16 回想一下，例 5.3 定义的布尔代数是由两个元素 0 和 1 及运算"′"、"+"和"*"构成的，其定义如下。

$0' = 1$

$1' = 0$

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 1$

$0 * 0 = 0$

$0 * 1 = 0$

$1 * 0 = 0$

$1 * 1 = 1$ □

如何证明集合的布尔代数与例 5.3 的布尔代数同构？

练习 5.17 如何证明命题的布尔代数与例 5.3 的布尔代数同构？ □

5.10 对偶性

在布尔代数中，一个陈述 s 的对偶(dual)是交换 s 中的"+"和"*"，并交换常量 0 和 1 而得到的陈述。例如，$0+1$ 的对偶是 $1*0$。

例 5.8 $(a*1)*(0+a')=0$ 的对偶是 $(a+0)-(1*a')=1$。 □

例 5.9 $a+a'b=a+b$ 的对偶是 $a*(a'+b)=a*b$。 □

例 5.10 在命题逻辑的布尔代数中，$p \wedge \text{true}$ 的对偶是 $p \vee \text{false}$。 □

例 5.11 在集合论的布尔代数中，$R \cap T$ 的对偶是 $R \cup T$。 □

对偶性的一个重要结论是，在布尔代数中，任意定理的对偶也是定理。例如，我们已经知道 $p \vee \text{true} \Leftrightarrow \text{true}$ 是命题逻辑的定理，不论 p 取什么值该式总为真。它的对偶 $p \wedge \text{false} \Leftrightarrow \text{false}$ 也是命题逻辑的定理，不论 p 取什么值该式同样总为真。

练习 5.18 给出下列各布尔代数等式的对偶。

1. $(a1)(0+a')=0$
2. $a+(a'b)=a+b$
3. $a(a'+b)=ab$
4. $(a+1)(a+0)=a$
5. $(a+b)(b+c)=ac+b$ □

练习 5.19 给出下列各命题逻辑定理的对偶。

1. $\neg p \wedge p \Leftrightarrow \text{false}$
2. $p \wedge q \Leftrightarrow q \wedge p$
3. $p \vee (q \vee r) \Leftrightarrow (p \vee q) \vee r$
4. $p \wedge (p \vee q) \Leftrightarrow p$ □

5.11 附加练习

练习 5.20 设 p、q 和 r 分别表示布尔代数的项 a、b 和 c，给出下列各布尔代数项

的命题逻辑形式。

1. $a+b$

2. $(a+b)'$

3. $(a'*b')+c'$

练习 5.21 设集合 R、S 和 T 分别表示布尔代数项 a、b 和 c，把练习 5.20 中的项表示成集合论的表达式。

练习 5.22 化简下列布尔表达式。

1. $a*(a*b')'$

2. $b*(a+b)'$

3. $((a*b)+(a'*c))+(b*c')$

练习 5.23 重新考虑练习 5.16 解答中的运算 f，其定义如下：

$f(S)=1$

$f(\varnothing)=0$

$f(a\cup b)=f(a)+f(b)$

$f(a\cap b)=f(a)*f(b)$

$f(a^{-})=f(a)'$

将该运算应用于项 $S\cap(\varnothing\cup S^{-})$。

练习 5.24 重新考虑练习 5.16 解答中的运算 f 和例 5.7 的运算 g，其定义如下：

$f(S)=1$

$f(\varnothing)=0$

$f(a\vee b)=f(a)+f(b)$

$f(a\wedge b)=f(a)*f(b)$

$f(\neg a)=f(a)'$

$g(\text{true})=S$

$g(\text{false})=\varnothing$

$g(a\vee b)=g(a)\cup g(b)$

$g(a\wedge b)=g(a)\cap g(b)$

$g(\neg a)=g(a)^{-}$

把 f 应用于将 g 应用于命题 $\text{true}\wedge(\text{false}\vee\neg\text{true})$的结果上。

练习 5.25 给出下列集合论规则的对偶。

1. $X\cap(Y\cup Z)=(X\cap Y)\cup(X\cap Z)$

2. $X\cap\varnothing=\varnothing$

3. $X\cup X^{-}=S$

练习 5.26 在练习 5.9 中，我们证明了 $a+1=1$ 成立。给出其对偶的证明。

练习 5.27 在练习 5.8 中，我们证明了 $a+a=a$ 成立。给出其对偶的证明。

练习 5.28 重写下列各布尔项，使得其中不含括号和乘号。

1. $(a*b)*(a'*c)$

2. $(a*(b*c))*b$

3. $(a*b)*(c'*(a*(b*a)))$

4. $(a*b)*(c'*(a*(b'*c')))$

练习 5.29 在布尔代数中，文字量定义为变量或变量的补，如 a 或 a'。另外，基本乘积定义为文字量的序列，其中每个变量最多只能出现一次。例如，ab' 是基本乘积，而 abb' 则不是。将下列布尔乘积化简为 0 或基本乘积。

1. $abba'c$

2. $abcbc'$

3. $abc'abc'$

4. $a'b'c'abc$

练习 5.30 当一个基本乘积 P 的所有文字量都是另一个基本乘积 Q 的文字量时，称 P 包含于 Q。例如，ac、a 和 bc 都包含于 abc，而 ac 不包含于 ac'。证明，若 P 和 Q 为基本乘积且 P 包含于 Q，那么有 $P+Q=P$。

5.12 练习解答

5.1 集合论的分配律如下。

$R\cap(S\cup T)=(R\cap S)\cup(R\cap T)$

$R\cup(S\cap T)=(R\cup S)\cap(R\cup T)$

命题逻辑的分配律如下。

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

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

5.2

1. $\{b, c, e\}$

2. $\{e\}$

3. $\{c, d, e\}$

4. $\{a, b, c, d, e\}$

5. $\varnothing$

5.3 对于任意的 $x \in S$，有

$x \in X \cap S$

$\Leftrightarrow x \in X \wedge x \in S$ [规则 4.18]

$\Leftrightarrow x \in X \wedge \text{true}$ [$x \in S$ 的假设]

$\Leftrightarrow x \in X$ [规则 3.4]

5.4 对于任意的 $x \in S$，有

$x \in X \cup X'$

$\Leftrightarrow x \in X \vee x \in X'$ [规则 4.12]

$\Leftrightarrow x \in X \vee x \in S \setminus X$ [规则 5.2]

$\Leftrightarrow x \in X \vee (x \in S \wedge x \notin X)$

[规则 4.25]

$\Leftrightarrow x \in X \vee (\text{true} \wedge x \notin X)$ [$x \in S$ 的假设]

$\Leftrightarrow x \in X \vee (x \notin X \wedge \text{true})$ [规则 3.7]

$\Leftrightarrow x \in X \vee x \notin X$ [规则 3.4]

$\Leftrightarrow \text{true}$ [规则 3.15]

$\Leftrightarrow x \in S$ [$x \in S$ 的假设]

5.5 对于任意的 $x \in S$，有

$x \in X \cap X' \Leftrightarrow x \in X \wedge x \in X'$ [规则 4.18]

$\Leftrightarrow x \in X \wedge x \in S \setminus X$ [规则 5.2]

$\Leftrightarrow x \in X \wedge x \in S \wedge x \notin X$ [规则 4.25]

$\Leftrightarrow x \in X \wedge x \notin X \wedge x \in S$ [规则 3.8]

$\Leftrightarrow \text{false} \wedge x \in S$ [规则 3.6]

$\Leftrightarrow x \in S \wedge \text{false}$ [规则 3.7]

$\Leftrightarrow \text{false}$ [规则 3.5]

$\Leftrightarrow x \in \varnothing$ [规则 4.2]

5.6

$0' = (a * a')'$ [补律]

$= a' + a''$ [规则 5.8]

$= a' + a$ [规则 5.6]

$= 1$ [补律]

5.7

$a + 0 = a$ [同一律]

$= a * 1$ [同一律]

5.8

$a + a = (a + a) * 1$ [同一律]

$= (a + a) * (a + a')$ [补律]

$= a + (a * a')$ [分配律]

$= a + 0$ [补律]

$= a$ [同一律]

5.9

$a + 1 = a + (a + a')$ [补律]

$= (a + a) + a'$ [规则 5.5]

$= a + a'$ [解答 5.8]

$= 1$ [补律]

5.10

1. ab
2. $a(b + c)$
3. $a + (bc)$
4. ab'
5. $(ab)'$

5.11

1. $a * (b')$
2. $(a * b) + c$
3. $a + (b')$
4. $(a * b) + (c * d) + (e * (f'))$

5.12 为证 $\mathcal{P}S$ 构成布尔代数，需要证明交换律、分配律、同一律及补律的公理均成立。由于对任意的集合 X、Y、$Z \in \mathcal{P}S$，下面的公理均成立，因此 $\mathcal{P}S$ 构成布尔代数。

$X \cup Y = Y \cup X$ [规则 4.15]

$X \cap Y = Y \cap X$ [规则 4.21]

$X \cup (Y \cap Z)$

$= (X \cup Y) \cap (X \cup Z)$ [规则 4.24]

$X \cap (Y \cup Z)$

$= (X \cap Y) \cup (X \cap Z)$ [规则 4.24]

$X \cup \varnothing = X$ [规则 4.13]

$X \cap S = X$ [规则 5.1]

$X \cup X' = S$ [规则 5.4]

$X \cap X' = \varnothing$ [规则 5.3]

5.13 为证 Π 构成布尔代数，需要证明交换律、分配律、同一律及补律的公理均成立。由于对任意命题 p、q、$r \in \Pi$，下面的公理均成立，因此 Π 构成布尔代数。

$p \vee q = q \vee p$ [规则 3.14]

$p \wedge q = q \wedge p$ [规则 3.7]

$p \vee (q \wedge r)$

$=(p \vee q) \wedge (p \vee r)$ [规则 3.16]

$p \wedge (q \vee r)$

$=(p \wedge q) \vee (p \wedge r)$ [规则 3.17]

$p \vee \text{false} = p$ [规则 3.11]

$p \wedge \text{true} = p$ [规则 3.4]

$p \vee \neg p = \text{true}$ [规则 3.15]

$p \wedge \neg p = \text{false}$ [规则 3.6]

5.14

1. $g(\text{true} \vee \text{false}) = g(\text{true}) \cup g(\text{false})$

$= S \cup g(\text{false})$

$= S \cup \varnothing$

2. $g(\neg(\text{true} \wedge (\text{true} \vee \text{false})))$

$=(g(\text{true} \wedge (\text{true} \vee \text{false})))^-$

$=(g(\text{true}) \cap g(\text{true} \vee \text{false}))^-$

$=(g(\text{true}) \cap (g(\text{true}) \cup g(\text{false})))^-$

$=(g(\text{true}) \cap (S \cup g(\text{false})))^-$

$=(g(\text{true}) \cap (S \cup \varnothing))^-$

$=(S \cap (S \cup \varnothing))'$

3. $g((\text{true} \wedge (\text{true} \vee \text{false})) \vee \text{false})$

$=g(\text{true} \wedge (\text{true} \vee \text{false})) \cup g(\text{false})$

$=(g(\text{true}) \cap g(\text{true} \vee \text{false}))$

$\cup g(\text{false})$

$=(S \cap g(\text{true} \vee \text{false})) \cup g(\text{false})$

$=(S \cap (g(\text{true}) \cup g(\text{false})))$

$\cup g(\text{false})$

$=(S \cap (S \cup g(\text{false}))) \cup g(\text{false})$

$=(S \cap (S \cup \varnothing)) \cup g(\text{false})$

$=(S \cap (S \cup \varnothing)) \cup \varnothing$

5.15

1. $h(\varnothing^-) = \neg h(\varnothing) = \neg \text{false}$

2. $h(\varnothing \cap S^-) = h(\varnothing) \wedge h(S^-)$

$= \text{false} \wedge h(S^-)$

$= \text{false} \wedge \neg h(S)$

$= \text{false} \wedge \neg \text{true}$

3. $h(\varnothing \cap (S \cup \varnothing))$

$=h(\varnothing) \wedge h(S \cup \varnothing)$

$=\text{false} \wedge h(S \cup \varnothing)$

$=\text{false} \wedge (h(S) \vee h(\varnothing))$

$=\text{false} \wedge (\text{true} \vee h(\varnothing))$

$=\text{false} \wedge (\text{true} \vee \text{false})$

5.16 可以给出如下定义的运算 f。

$f(S)=1$

$f(\varnothing)=0$

$f(R \cup T)=f(R)+f(T)$

$f(R \cap T)=f(R) * f(T)$

$f(R^-)=f(R)'$

5.17 可以给出如下定义的运算 f。

$f(\text{true})=1$

$f(\text{false})=0$

$f(p \vee q)=f(p)+f(q)$

$f(p \wedge q)=f(p) * f(q)$

$f(\neg p)=f(p)'$

5.18

1. $(a+0)+(1a')=1$
2. $a(a'+b)=ab$
3. $a+(a'b)=a+b$
4. $(a0)+(a1)=a$
5. $(ab)+(bc)=(a+c)b$

5.19

1. $\neg p \vee p \Leftrightarrow \text{true}$
2. $p \vee q \Leftrightarrow q \vee p$
3. $p \vee (q \wedge r) \Leftrightarrow (p \wedge q) \wedge r$
4. $p \vee (p \wedge q) \Leftrightarrow p$

5.20

1. $p \vee q$
2. $\neg(p \vee q)$
3. $(\neg p \wedge \neg q) \vee \neg r$

5.21

1. $R \cup S$
2. $(R \cup S)^-$
3. $(R^- \cap S^-) \cup T^-$

5.22

1. $a * (a * b')' = a * (a'+b'')$

$=a * (a'+b)$

$=(a * a')+(a * b)$

$=0+(a * b)$

$=a * b$

2. $b*(a+b)'=b*(a'*b')$
$=b*(b'*a')$
$=(b*b')*a'$
$=0*a'$
$=0$

3. $((a*b)+(a'*c))+(b*c')$
$=((a'*c)+(a*b))+(b*c')$
$=(a'*c)+((a*b)+(b*c'))$
$=(a'*c)+((b*a)+(b*c'))$
$=(a'*c)+(b*(a+c'))$
$=((a'*c)+b)*((a'*c)+(a+c'))$
$=((a'*c)+b)*((a'*c)+(a''+c'))$
$=((a'*c)+b)*((a'*c)+(a'*c)')$
$=((a'*c)+b)*1$
$=(a'*c)+b$

5.23

$f(S\cap(\varnothing\cup S^{-}))=f(S)*f(\varnothing\cup S^{-})$
$=1*f(\varnothing\cup S^{-})$
$=1*(f(\varnothing)+f(S^{-}))$
$=1*(f(\varnothing)+f(S)')$
$=1*(0+f(S)')$
$=1*(0+1')$

5.24 将 g 应用于 true∧(false∨− true)得 $S\cap(\varnothing\cup S^{-})$。由上题的解答，$f(S\cap(\varnothing\cup S^{-}))$给出 $1*(0+1')$。

5.25

1. $X\cup(Y\cap Z)=(X\cup Y)\cap(X\cup Z)$
2. $X\cup S=S$
3. $X\cap X^{-}=\varnothing$

5.26

$a*0=a*(a*a')$ [补律]
$=(a*a)*a'$ [结合律]
$=a*a'$ [幂等律]
$=0$ [补律]

5.27

$a*a=(a*a)+0$ [同一律]
$=(a*a)+(a*a')$ [补律]
$=a*(a+a')$ [分配律]
$=a*1$ [补律]
$=a$ [同一律]

5.28

1. $aba'c$
2. $abcb$
3. $abc'aba$
4. $abc'ab'c'$

5.29

1. $abba'c=aa'bbc$
$=0bbc$
$=0$

2. $abcbc'=abbcc'$
$=abb0$
$=0$

3. $abc'abc'=aabbc'c'$
$=abc'$

4. $a'b'c'abc=aa'bb'cc'$
$=0bb'cc'$
$=0$

5.30 若 P 包含于 Q，则存在 R，使得 $Q=P*R$。由此，有

$P+Q=P+(P*R)$
$=(P*1)+(P*R)$ [同一律]
$=P*(R+1)$ [分配律]
$=P*1$ [规则 5.9]
$=P$ [同一律]

第6章　类型集合论

到目前为止，我们在讨论集合论的运算符时，对陈述的类型都十分小心。在例4.18中，我们说过 *eaten* 是 *fruitbowl* 的子集：二者都是与水果有关的集合。并且，我们尽量避免使用运算符∪、∩或\把数字集合与水果集合这样的集合连接在一起。当使用这些运算符时，一定要确保参与运算的集合必须含有相同类型(type)的元素。

乍看起来，诸如 $1 \in eaten$、$banana \in \mathbb{N}$ 或者 $\mathbb{N} \subseteq People$ 这样的陈述看起来不像是真的。如果对这样的陈述取否而得到陈述 $1 \notin eaten$、$banana \notin \mathbb{N}$、$\mathbb{N} \not\subseteq People$，那么你可能会说，这些否定的陈述是真的。然而，在离散数学中，第二类陈述与第一类陈述一样是没有意义的，这是因为我们关心的是类型集合论(typed set theory)。

类型集合论认为每个集合(事实上，每个元素)都有一个与之相关联的类型。例如，为了讨论陈述 $s \in S$，S 必须是一个这样的集合：它的元素具有与 s 相同的类型。类似地，为了讨论形如 $S \subseteq T$ 或者 $S \cap T$ 这样的陈述，必须确保 S 和 T 具有相同的类型。再例如，如果 S 是一个数的集合，那么为使 $S \cap T$ 有意义，T 必须也是一个数的集合。这与我们把集合论看成是布尔代数的想法是一致的。

这种方法的一个主要优势是，当我们在"现实世界"实现集合时，例如，在关系数据库的上下文中，所有的条目都必须有相关的类型。如果我们要考虑一个体育馆的所有用户的附属信息，并且希望输出所有会员的名字和地址，这时我们可能不希望输出他们的年龄和电话号码；这些对象的类型是完全不同的。另外，我们可能需要合并两个会员编号的集合，而且可能希望当把会员地址集合与会员编号集合合并时产生一个错误信息。正是类型集合的概念为我们提供了这样一种理想的结构。正如我们将要看到的，本章与接下来的两章有着很密切的联系，而且能更方便地应用于关系数据库。

因此，在本章，我们将概述为什么要使用类型集合论，并且描述一些根据这种方法定义集合的方式。关于这一话题的深入内容，有兴趣的读者可以参考[Sup60]。

6.1　类型的需要

在第4章，我们讨论了集合的概念，定义集合为"一些对象的无序汇集"。尽管我们尽量保证集合包含相似类型的对象，把不同类型的对象互相分开，但是在那里并不需要这么做。然而，如果继续采用这样不严格的方式，那么将会出现一系列的悖论；在此，讨论两个最著名的悖论。

Cantor 悖论

考虑所有集合的集合，用 S 来表示这一集合。在此，

$X \in S \Leftrightarrow X$ 是一个集合

S 的每一个子集是一个集合，因此，S 的每一个子集是 S 的元素。因此，有 $\mathcal{P}S \subseteq S$。然而，就像第4章中看到的，事实上，任何一个集合 X 的幂集的势都比 X 的势要大(因为如果 X 有 n 个元素，那么 $\mathcal{P}X$ 的幂集就有 2^n 个元素)。这样，这两个式子是相悖的。这就是著名的 Cantor 悖论。[①]

① 以 Georg Cantor(1845—1918)命名。参见[Dau90]以了解 Cantor 的生平和工作。

Russell 悖论

有些集合可以是它们自己的成员；例如，所有集合的集合 S 就是它自己的成员。另一方面，一些集合不是它们自己的成员：所有狗构成的集合就不是一条狗。假设 R 表示所有不是自己的成员的集合构成的集合，那么有

$$X \in R \Leftrightarrow X \notin X$$

这个定义带来了一个问题，"R 是它自己的成员吗？"如果 $R \in R$ 的话，那么 R 就是所有不是自己成员的集合所构成的集合的成员，因此，$R \notin R$。另一方面，如果 $R \notin R$，那么 R 就不是所有不是自己成员的集合所构成的集合的成员，因此，$R \in R$。这两个陈述是矛盾的。这就是著名的 Russell 悖论[1]。

如果像第 4 章中那样，对集合的概念不给出严格的定义，那么就会导致上述的悖论。为了避免产生这些和其他的悖论，公理集合论(axiomatic set theory)[2]引入了一组公理，我们已经看到了其中的几个公理。因此，只有当一个集合满足这些公理时它才真的有定义[3]。特别地，这些公理保证诸如"所有集合的集合"和"所有不是自己成员的集合构成的集合"等陈述是无效的，我们不允许自己去讨论这样的集合。

在强制地给出一系列规则的同时，公理集合论强调为每一个集合赋予一个类型(type)[4]。因此，集合论中的运算符，如 $\cup$ 和 $\cap$，只能应用于相同类型的两个集合。

例 6.1 我们可以考虑人的类型，记作 *People*，以及汽车的类型，记作 *Cars*。对于任意的集合 X、$Y \subseteq People$，可以自由地讨论它们的交、并和差，但是，当 $Z \subseteq Cars$ 时，不能讨论 X 和 Z 的并或交，因为它们的类型是不兼容的。 □

练习 6.1 假定 *People* 和 *Cars* 是类型，那么 *People* 表示所有人的集合，而 *Cars* 表示所有汽车的集合。假定 A、$B \subseteq People$，而 X、$Y \subseteq Cars$，那么下面哪些交的运用是合法的？

1. $A \cap B$
2. $A \cap X$
3. $A \cap Y$
4. $A \cap People$
5. $A \cap Cars$
6. $A \cap \varnothing$ □

前面介绍过人的集合与人的集合的集合是完全不同的，因为前者的元素类型与后者的元素类型是不同的。例如，前者的一个代表是{乔}而后者的一个代表是{{乔}}。

例 6.2 再次考虑 A、$B \subseteq People$，$A \cap B$ 和 $A \cap People$ 是有定义的。但是，$A \cap \{B\}$ 和 $\{A\} \cap People$ 是未定义的。 □

练习 6.2 下面哪一个 $\cap$ 的应用是合法的？

1. $\{1, 2, 3\} \cap \{\{1\}\}$
2. $\{1, 2, 3\} \cap \varnothing$
3. $\{1, 2, 3\} \cap \{\varnothing\}$
4. $\{1, 2, 3\} \cap \{1\}$ □

在限制集合运算符 $\cap$、$\cup$、$\setminus$ 只能运用于相同类型的集合的同时，类型集合论也限定了对不同级别的对象进行操作的运算符，像 $\in$、$\subseteq$ 的运用。

为了说明这样做的理由，我们考虑一个装有一根香蕉的桶。现在考虑将这一桶香蕉

---

[1] 以 Bertrand Russell(1872—1970)命名。参见[Gar92]以了解 Russell 和 Russell 悖论的影响。[Mon97]记载了 Russell 前 50 年的辉煌生平。

[2] 最初由 Ernst Zermelo(1871—1953)和 Abraham Fraenkel(1891—1965)所创立。

[3] 在此，我们不列出这些公理。读者只要知道大部分公理可以用第 4 章的规则来刻画就可以了。

[4] 可以认为一个类型是包含我们要考虑的所有"类似"对象的一个"大"集合。注意，本书不把 $\varnothing$ 作为类型。——译者注

插入到另一个桶中去。如果有人问：第二个桶中有什么？答案将不是“一根香蕉”，而是“一个装有一根香蕉的桶”。类型集合论规定了这样的原则：{{1}，{2}}的元素是{1}和{2}，而不是1和2。因此，陈述{1}∈{{1}，{2}}是合法的，而陈述1∈{{1}，{2}}是不合法的。按照这种说法，如果x比X“低一个级别”，那么$x\in X$的值是有定义的。

例6.3 假定类型*People*代表了所有人的集合，而且，安迪、邓肯、约翰、理查德都是*People*的元素。

下面∈、⊆的运用是合法的，而且它们的值都为true。

$\varnothing\subseteq People$

$\varnothing\in \mathcal{P}People$

安迪∈*People*

{安迪，约翰}⊆*People*

{安迪，约翰}⊆{安迪，邓肯，约翰，理查德}

下面的∈、∉和⊆的运用都是合法的，并且都有值false。

$People\subseteq\varnothing$

$\varnothing\notin \mathcal{P}People$

安迪∉*People*

{安迪，约翰}⊆{约翰}

{安迪，约翰}⊆{安迪，邓肯，理查德}

下面的∈、⊆的运用是不合法的。

$\varnothing\subseteq$安迪

$\varnothing\in People$

安迪⊆*People*

{安迪，约翰}∈*People*

{安迪，约翰}∈{安迪，邓肯，约翰，理查德} □

练习6.3 假设集合$S=\{1, 2\}$，下面哪些陈述是合法的？

1. $S\in S$
2. $S\subseteq S$
3. $\{1\}\in S$
4. $1\in S$
5. $\{1\}\subseteq S$
6. $1\subseteq S$
7. $S\in \mathcal{P}S$
8. $\{S\}\in \mathcal{P}S$
9. $\{1\}\in \mathcal{P}S$
10. $1\in \mathcal{P}S$ □

6.2 再论空集

对于类型集合论的考虑引发了一些关于空集的重要问题。考虑项$\varnothing\cap\{1\}$，这是∩的合法运用，并且结果为空集。现在考虑{杰克}\{杰克}，同样，这也是集合论运算符的合法运用，其结果等于空集。在这两个前提下，试问

({杰克}\{杰克})∩{1}

是合法的吗？答案当然是否定的。尽管$\varnothing\cap\{1\}$和{杰克}\{杰克}都是集合论运算符的合法运用，但是，陈述({杰克}\{杰克})∩{1}不是合法的。这是因为陈述$\varnothing\cap\{1\}$和{杰克}\{杰克}的结果是不同类型的空集。

这其中的原因就是，类型集合论要求不只有一个唯一的空集，而是有许许多多，事实上有无穷多个空集。特别地，类型集合论要求每一个类型都有一个相应的空集。因此，人的空集和自然数的空集是不同的。前者可以和人的集合进行运算，而后者只能和自然数的集合进行运算。然而，人的空集不能与自然数的任意集合(包括自然数的空集)进行运算。

在上面的例子中，$\varnothing\cap\{1\}$中的空集是自然数的空集，而{杰克}\{杰克}的结果是人的空集。尽管我们从字面上并不能区分二者，但是可以从上下文判断出它们的含义。这样，({杰克}\{杰克})∩{1}是不合法陈述的原因就在于我们试图要建立一个人的空集与一个含有自然数1的集合的交集：很明显类型不匹配。

练习 6.4 假设类型 Car 有{福特，法拉利}⊆Car，下面的哪些项是有定义的？

1. {福特，法拉利}∩{1，2}
2. {福特，法拉利}∩∅
3. {1，2}∩∅
4. {福特，法拉利}∪({1，2}∩{3，4})

□

6.3 集合描述

在第4章，我们看到了如何将集合作为抽象的对象来定义，例如，集合 *People* 代表了世界上所有人的集合，或者，明确地列举出集合的所有成员，例如，

Union ={英格兰，北爱尔兰，苏格兰，威尔士}

本节，我们展示使用已有的集合定义新集合的方法，这被称为集合描述(set comprehension)。

这种集合描述的形式如下。

{声明 | 谓词}

其中，声明(declaration)包括适当类型的对象的枚举，而谓词(predicate)则起了一个过滤器的作用：保留所有满足谓词的对象，而丢掉那些不满足谓词的对象。

例 6.4 所有拥有移动电话的人的集合 *Mobile_owners* 可以定义如下。

Mobile_owners ={ *p*: *People* | *p* 拥有移动电话}

这里，声明 *p*: *People* 表明我们讨论的是所有的人，而谓词"*p* 拥有移动电话"表示只有拥有移动电话的人才能被包括在这个集合中。 □

例 6.5 使用集合描述可以如下定义偶自然数的集合。

$even\_naturals=\{n: \mathbb{N} \mid n \bmod 2 = 0\}$

这一集合描述生成包含元素 0，2，4，6，等等的集合。 □

集合描述可以被看成是一台机器：声明提供了输入，而谓词检查每一个输入是否满足相关的性质。例如，对于上例，机器首先生成0，依照标准对其进行检查：0通过了测试，因此被保留了下来。下一步，机器生成1，依照标准对其进行检查：1没有通过测试，被排除掉。下一步，机器生成2，依照标准对其进行检查：2通过了测试，因此被保留了下来。持续这一过程(对于此例，这一过程无限持续)。

练习 6.5 列举下列集合的元素。

1. $A=\{n: \mathbb{N} \mid n<7\}$
2. $B=\{n: \mathbb{N} \mid n<10 \wedge n \bmod 3=0\}$
3. $C=\{n: \mathbb{N} \mid (4\times n=12) \vee (6\times n=12)\}$
4. $D=\{n: \mathbb{N} \mid n<0\}$ □

练习 6.6 假设类型 *People* 和 *Car* 以及适当的谓词，使用集合描述定义下列集合。

1. 所有红汽车的集合。
2. 所有素食者的集合。
3. 所有有天窗的汽车的集合。
4. 所有富人的集合。 □

诸如"*p* 是富有的"这样的陈述(目前)当然是非形式的：我们没有形式地刻画如何用数学的方式来表示这样的陈述。在随后的章节，我们将看到如何使用谓词形式地表示这样的陈述。然而，这样的半形式的方法在此处已经足够了。

在前面的练习中，我们生成了汽车的集合和人的集合。然而，在某些情况下，需要关于富人的进一步的信息；例如，我们可能不关心这些人本身，而是关心他们的地址。集合描述表记的一个扩展使我们能够获得这些信息。在这样的情况下，集合描述的形式如下。

{声明 | 谓词 · 项目}

这里，对于那些由声明所枚举出的并被谓词所过滤的对象，项目(term)生成一个项。

例 6.6 如果我们希望生成一个包含所有拥有移动电话的人的地址的集合，那么这

个集合的定义如下。

$Mobile\_addresses=\{p: People \mid p$ 拥有移动电话 $\cdot address(p)\}$ □

在上例中，我们假定 *address* 是一个以人为输入、以其地址为输出的运算。

例 6.7 所有偶自然数的平方的集合定义如下。

$$even\_natural\_squares=\{n: \mathbb{N} \mid n \bmod 2=0 \cdot n^2\}$$

这个集合描述生成 0，4，16，36，等等。 □

我们再次把集合描述看作一台机器：第一步生成对象，第二步过滤，第三步对它们实施一些操作，并将操作结果作为输出。因此，在上例中，声明生成数 0，1，2，3，4，5，等等。下一步（谓词阶段）过滤这些数，0，2，4 被保留下来，1，3，5 被拒绝，以此类推。最后，集合描述的项目部分把保留下来的数作为输入，经过操作得到输出：0 的平方等于 0，2 的平方等于 4，4 的平方等于 16，以此类推。

练习 6.7 列举下列集合的元素。

1. $A=\{n: \mathbb{N} \mid n<7 \cdot n^2\}$
2. $B=\{n: \mathbb{N} \mid n<10 \wedge n \bmod 3=0 \cdot n\}$
3. $C=\{n: \mathbb{N} \mid (4\times n=12) \vee (6\times n=12) \cdot n \text{ div } 2\}$
4. $D=\{n: \mathbb{N} \mid n<0 \cdot n+1\}$ □

练习 6.8 假设类型 *People* 和 *Car*，使用集合描述定义下列集合。对于每种情况，必须假设存在相关的操作。

1. 所有红汽车的注册号的集合。
2. 所有素食者的年龄的集合。
3. 所有有天窗的汽车的拥有者的集合。
4. 所有富人的地址的集合。 □

正如希望定义所有红汽车的注册号集合一样，我们也许需要更一般地定义所有汽车的注册号的集合，而不管它们是什么颜色的。在这种情况下，需要使用一个不拒绝声明所生成的任何元素的过滤器；也就是说，一个总是成立的谓词。当然，在数学语言中有这样的谓词：true。因此，下面的集合描述给出我们需要的集合。

$\{c: Car \mid \text{true} \cdot registration(c)\}$

在这样的情形下，我们可以自由地省略这种集合描述的谓词部分，因为它对最终的结果并不起任何作用：在机器刻画中，可以假设第二个步骤被关掉了。因此，在这个例子中，也可以定义

$\{c: Car \cdot registration(c)\}$

来得到同样的结果。

例 6.8 所有自然数的平方的集合可以定义如下。

$squares=\{n: \mathbb{N} \cdot n^2\}$

这一集合描述生成包含 0，1，4，9，16，等等的集合。 □

练习 6.9 列举出下列集合的前九个元素。

1. $A=\{n: \mathbb{N} \cdot n^2\}$
2. $B=\{n: \mathbb{N} \cdot n\}$
3. $C=\{n: \mathbb{N} \cdot n \text{ div } 2\}$
4. $D=\{n: \mathbb{N} \cdot n+1\}$ □

练习 6.10 假设有类型 *People* 和 *Car*，使用集合描述定义下列集合。在每种情况下，假定存在相关的操作。

1. 所有汽车注册号的集合。
2. 人的年龄的集合。
3. 汽车拥有者的集合。
4. 所有人的地址的集合。 □

6.4 特征组

正如可以在集合描述中省略谓词部分一样，也可以省略项目部分。事实上，在集合描述的前几个例子中，就没有使用项目。在这些例子中，集合描述所生成的元素的类型不是由项目部分来决定的，而是由声明决定的。这一信息称为集合描述的特征组（characteristic tuple），相当于在这样的集

合描述法中强加了一个默认的项目。

例如，集合描述

$\{c: Car \mid c$ 是红色的$\}$

和

$\{c: Car \mid c$ 是红色的 $\cdot c\}$

是完全相等的。在这个例子中，特征组是 c。作为另一个例子，集合描述

$\{n: \mathbb{N} \mid n<7\}$

和

$\{n: \mathbb{N} \mid n<7 \cdot n\}$

也是相等的。在这个例子中，特征组是 n。

至此，我们见到的所有特征组都是简单的。然而，当我们在6.6节学习笛卡儿积[○]时，特征组的概念的实用性就会显示出来。

练习6.11 给出下列集合的特征组。

1. $A=\{n: \mathbb{N} \mid n<7\}$
2. $B=\{m: \mathbb{N} \mid m<10 \wedge m \bmod 3=0\}$

□

6.5 缩写

有时候在处理自然数的子集的时候，我们或许不希望完整地写出这个子集的元素；缩写可能会更方便。解决这个问题的一个方法就是使用“..”运算符，意思是“up to”(至)。给定两个自然数 x 和 y 使得 $x \leqslant y$，那么可以用 $x..y$ 来表示包括 x 和 y 在内的出现在 x 和 y 之间的所有自然数的集合。因此，例如，1..3 表示集合{1，2，3}。

例6.9 10到20之间的自然数的集合(包括端点)可以写作10..20。这个集合等于集合

{10，11，12，13，14，15，16，17，18，19，20} □

例6.10 例4.2中的集合 *pool_balls* 也可以如下定义。

$pool\_balls=1..15$ □

练习6.12 完整地写出下列集合。

1. 0..5
2. 1..1
3. 0..5 \ 0..3
4. $\{n: \mathbb{N} \mid n \in 0..3\}$ □

当希望表示无限地持续下去的一系列数字时，不写出缩写的上边界。然而，在这种情况下，明确地给出序列的前几个数字来显示元素的走向。因此，例如3，4，5，…表示从3开始的所有自然数的集合。当然，这也可以写成 $\mathbb{N} \setminus \{0, 1, 2\}$。另一个例子是0，2，4，6，8，…表示偶自然数的集合。

练习6.13 完整地写出下列集合。

1. $\{n: 0..10 \mid n \in 7, 8, 9, \cdots\}$
2. $\{n: 0..10 \mid n \in 2, 4, 6, \cdots\}$
3. $\{n: 0..10 \mid n \in 3, 6, 9, \cdots\}$ □

6.6 笛卡儿积

在第4章和本章的前面，主要考虑了相对简单的集合。另外，我们看到的所有集合论的运算符都是在相同的简单类型的集合之间进行操作的。然而，在现实中，常常需要讨论混合着不同类型的结构。例如，在数据库中存在很多不同类型的对象，数据库结构需要把它们适当地组合在一起。由雇员的名字和薪水组成的记录必须把不同的、简单的信息以及类型信息组合起来，并且在更复杂、更深入的结构中存储相关的信息。如果要形式地讨论这些对象，那么就需要一种使用适当的方式来构造信息的运算符。

把不同类型的集合组合起来的集合论运算符称为笛卡儿积运算符(Cartesian product operator)。给定两个集合 A 和 B，A 和 B 的笛卡儿积记作 $A \times B$，定义如下。

$A \times B=\{a: A; b: B \cdot (a, b)\}$

○ 以Rene Descartes(1596—1650)的名字命名。

这定义了所有序偶(a, b)的集合，其中$a \in A$，$b \in B$。本质上，笛卡儿积考虑的是A中的所有元素和B中的所有元素，并产生一个包含这些元素的所有可能的序偶的集合。

例 6.11 考虑下列集合。

$Name=\{$杰克，吉尔$\}$

$Age=\{65, 60\}$

这里，$Name$有两个元素：杰克和吉尔，Age也有两个元素：65和60。因此，杰克和吉尔都可以和Age中的两个元素配对。所以，笛卡儿积$Name \times Age$是：

{(杰克，65)，(杰克，60)，(吉尔，65)，(吉尔，60)}

注意，每个序偶的第一个元素是第一个集合中的元素，序偶中的第二个元素是第二个集合中的元素。因此，(杰克，杰克)、(65，60)都不是$Name \times Age$的元素。而(杰克，65)和(65，杰克)是不同的对象，因为我们处理的是序偶(ordered pair)。事实上，这些序偶具有不同的类型：前者是$Name \times Age$的元素，后者是$Age \times Name$的元素。 □

例 6.12 考虑下列集合。

$Die=\{1, 2, 3, 4, 5, 6\}$

$Coin=\{$正面，反面$\}$

这时，有

$Die \times Coin=\{$(1，正面)，(2，正面)，(3，正面)，(4，正面)，(5，正面)，(6，正面)，(1，反面)，(2，反面)，(3，反面)，(4，反面)，(5，反面)，(6，反面)$\}$

另外，有

$Coin \times Die=\{$(正面，1)，(正面，2)，(正面，3)，(正面，4)，(正面，5)，(正面，6)，(反面，1)，(反面，2)，(反面，3)，(反面，4)，(反面，5)，(反面，6)$\}$

□

练习 6.14 考虑下列集合。

$A=\{0, 1\}$

$B=\{$yes，no$\}$

下面的序偶属于哪个笛卡儿积？

1. (0，yes)
2. (1，0)
3. (no，yes)
4. (no，1)
5. (no，{yes})
6. (no，{(1，yes)}) □

练习 6.15 考虑集合$A=\{0, 1\}$和$B=\{$yes，no$\}$，列举下列集合的元素。

1. $A \times A$
2. $A \times B$
3. $B \times A$
4. $B \times B$
5. $A \times \mathbb{P}B$
6. $A \times \varnothing$ □

练习 6.16 考虑集合$A=\{0, 1\}$和$B=\{$yes，no$\}$，对于下列集合，给出它们的一个元素。

1. $A \times B$
2. $A \times \mathbb{P}B$
3. $A \times \mathbb{P}\varnothing$
4. $(\mathbb{P}A) \times B$
5. $(\mathbb{P}A) \times (\mathbb{P}B)$
6. $(\mathbb{P}A) \times (\mathbb{P}\varnothing)$
7. $\mathbb{P}(A \times B)$
8. $\mathbb{P}(A \times \mathbb{P}B)$
9. $\mathbb{P}(A \times \mathbb{P}\varnothing)$ □

注意，笛卡儿积的大小是由每个参与运算的集合的大小决定的。例如，在上面的练习中，A和B都有两个元素，因此，$A \times B$的大小为4：有四种可能的序偶。下面的规则刻画了这种关系。

规则 6.1 对于任意的集合X和Y，有

$\#(X \times Y)=\#X \times \#Y$ □

练习 6.17 下面的集合有多少个元素？

1. $\{a, b, c\} \times \{0, 1\}$
2. $\{0, 1\} \times \{a, b, c\}$
3. $\{a, b, c\} \times \varnothing$
4. $\{a, b, c\} \times \mathbb{F}\{0, 1\}$
5. $\{a, b, c\} \times \mathbb{F}\varnothing$
6. $\mathbb{F}(\{0, 1\} \times \{a, b, c\})$
7. $\mathbb{F}(\{a, b, c\} \times \varnothing)$
8. $\mathbb{F}(\mathbb{F}\{0, 1\} \times \mathbb{F}\{a, b, c\})$ □

正如从定义中所看到的那样，笛卡儿积的概念允许我们通过集合描述定义更加复杂的结构。例如，通过集合描述，可以用如下方式定义所有小于3的自然数序偶的集合。

$low\_pairs = \{m, n: \mathbb{N} \mid m < 3 \wedge n < 3 \cdot (m, n)\}$

它生成如下集合。

$\{(0, 0), (0, 1), (0, 2), (1, 0), (1, 1), (1, 2), (2, 0), (2, 1), (2, 2)\}$

这里，集合中的项目部分不是必要的，因为这一集合描述的特征组就是(m, n)。不论何时，当集合描述的声明由两个元素组成且没有项目部分时，特征组就是相应类型的一个序偶。在这个例子中，特征组就是类型$\mathbb{N} \times \mathbb{N}$的一个序偶。

作为第二个例子，下面的集合描述有特征组(n, a)，其中$n \in Name$，$a \in Age$。

$$\{n: Name; a: Age \mid n = Norman \vee a = 31\}$$

这里，声明的顺序是十分重要的：下面集合描述的特征组就不是(n, a)而是(a, n)。

$$\{a: Age; n: Name \mid n = Norman \vee a = 31\}$$

上面的声明给出一个对声明的统一约定。本书中将一直使用这种约定：不同类型的对象的声明之间必须用分号分隔。然而，相同类型的对象的声明之间用逗号分隔即可（就如在 *low_pairs* 的定义中所看到的那样）。事实上，在这样的场合使用逗号是一种简写；也可以等价地如下定义 *low_pairs*。

$low\_pairs = \{m: \mathbb{N}; n: \mathbb{N} \mid m < 3 \wedge n < 3 \cdot (m, n)\}$

练习 6.18 确定下列集合描述的特征组。

1. $\{n: \mathbb{N} \mid p\}$
2. $\{m, n: \mathbb{N} \mid p\}$
3. $\{m: \mathbb{N} \times \mathbb{N}; n: \mathbb{N} \mid p\}$ □

练习 6.19 已知集合 *Girls* 和 *Boys* 定义如下。

Girls={爱米丽，拉切尔，安娜}

Boys={迈克尔，约翰}

列举下列集合的元素。

1. $\{b: Boys; g: Girls\}$
2. $\{g: Girls; b: Boys\}$
3. $\{b, c: Boys\}$
4. $\{b, c: Boys \mid b \neq c\}$
5. $\{b: Boys \cdot (b, b)\}$
6. $\{b: Boys; g: Girls \cdot b\}$
7. $\{b: Boys; g: Girls \mid b =$ 约翰$\}$
8. $\{b: Boys; g: Girls \mid b =$ 约翰 $\cdot g\}$

□

因为笛卡儿积本身就是集合（序偶的集合），所以我们之前见到的所有集合论的运算符都同样适用于笛卡儿积，例如，

{(杰克，65)，(吉尔，60)}⊆{(杰克，65)，(杰克，60)，(吉尔，65)，(吉尔，60)}

是合法的陈述，并且，它是真的。

练习 6.20 下面哪个“\”的运用是合法的？

1. $(\mathbb{N} \times \mathbb{N}) \setminus \{3, 4\}$
2. $(\mathbb{N} \times \mathbb{N}) \setminus \{3, 4\}$
3. $(\mathbb{N} \times \mathbb{N}) \setminus \{(3, 4)\}$
4. $(\mathbb{F}(\mathbb{N} \times \mathbb{N})) \setminus \{(3, 4)\}$
5. $(\mathbb{F}(\mathbb{N} \times \mathbb{N})) \setminus \{\{(3, 4)\}\}$ □

练习 6.21 给出下列∪、∩、×的结果。

1. $(\{0, 1, 2\} \cap \{1, 2, 3\}) \times \{2, 3\}$
2. $(\{0, 1, 2\} \cup \{1, 2, 3\}) \times \{2, 3\}$
3. $(\{0, 1, 2\} \setminus \{1, 2, 3\}) \times \{2, 3\}$ □

到目前为止，我们把笛卡儿积局限于序偶的生成。然而，笛卡儿积的应用更加广泛　我们可以用它来定义三元组、四元组、五元组。事实上，对于任意的 $n \in \mathbb{N}$，可以用笛卡儿积生成 n 元组。另外，可以考虑序偶的序偶、序偶的三元组、三元组的序偶，等等。

例 6.13　考虑一个存储姓名、年龄和电子邮件地址的数据库，其中姓名、年龄和电子邮件地址分别为类型 *Name*、*Age*、*Email*。下面的笛卡儿积生成三元组。

$Name \times Age \times Email$

假设，*Name*、*Age*、*Email* 的定义如下。

$Name = \{$杰克，吉尔$\}$

$Age = \{60, 65\}$

$Email = \{$jack@the_hill.com, jill@the_hill.com$\}$

那么，这个笛卡儿积的一个代表是(杰克，65，jack@the_hill.com)。 □

正如笛卡儿积中的顺序非常重要一样，括号也同样重要。笛卡儿积 $Name \times Age \times Email$ 产生一个三元组；而笛卡儿积 $Name \times (Age \times Email)$ 产生一个序偶，其中，每个序偶的第一个成分是 *Name* 的成员，而第二个成分是 $Age \times Email$ 类型的序偶。例如，(杰克，(65，jack@the_hill.com))是 $Name \times (Age \times Email)$ 的元素。

练习 6.22　假设有集合 *Name*、*Age* 和 *Email* 同上，列举出下列集合的元素。

1. $Name \times Age \times Email$
2. $Name \times (Age \times Email)$
3. $(Name \times Age) \times Email$ □

练习 6.23　假设有集合 *Name*、*Age* 和 *Email* 同上，列举出下列集合的元素。

1. $\{a, b, c: Age\}$
2. $\{n: Name \bullet (n, n, n)\}$
3. $\{m, n: Name \mid m = $杰克$ \bullet (m, (m, n))\}$
4. $\{m, n: Name \bullet ((m, n), (m, n))\}$
5. $\{a: Age \times Email\}$
6. $\{a: Age; n: Name; e: Email \mid a = 65 \wedge n = $杰克$ \bullet (n, a, e)\}$ □

至此，我们学习了如何利用已有的集合通过笛卡儿积来构造新的集合。然而，有时候，我们希望做相反的工作，也就是说，给定一个序偶的集合，只考虑每个序偶的第一个成分。可以通过成分筛选(component selection)来达到这一目的。

例 6.14　假设(杰克，65，jack@the_hill.com)是笛卡儿积 $Name \times Age \times Email$ 的一个元素，并且希望得到杰克的年龄，即三元组的第二个成分。为了得到多元组的一个成分，用一个点号来标识我们考虑的成分的位置。例如，成分筛选(杰克，65，jack@the_hill.com).2 表示我们关心的是这个三元组的第二个成分。因此，有

(杰克，65，jack@the_hill.com).2＝65

另外，也可以得到如下结果。

(杰克，65，jack@the_hill.com).1＝杰克

(杰克，65，jack@the_hill.com).3＝jack@the_hill.com □

例 6.15　考虑下列集合描述。

$\{x: Name \times Age \times Email \bullet x.1\}$

这里，集合描述的声明部分生成了相应类型的所有三元组，而项目部分把每个三元组的第一个成分提取出来。因此，这个集合等于{杰克，吉尔}。 □

练习 6.24　计算下列集合。

1. (爱米丽，金鱼).1
2. (爱米丽，金鱼).2
3. (邓肯，(格比尔，红色)).1
4. (邓肯，(格比尔，红色)).2
5. ((爱米丽，金鱼)，(邓肯，格比尔)).1
6. ((爱米丽，金鱼)，(邓肯，格比尔)).2

7. ((爱米丽，金鱼)，(邓肯，格比尔)).1.2 □

练习 6.25 假设集合 *Girls* 和 *Boys* 定义如下。

Girls={爱米丽，米切尔，安娜}

Boys={迈克尔，约翰}

列举下列集合的元素。

1. $\{x: Boys \times Girls \cdot x.1\}$

2. $\{x: Girls \times Boys \cdot x.1\}$

3. $\{x: Girls \times Boys \mid x.2 = 迈克尔\}$

4. $\{x: Girls \times Boys \mid x.2 = 迈克尔 \cdot x.1\}$

5. $\{x: Girls \times Boys \mid x.2 = 迈克尔 \cdot x.2\}$ □

6.7 公理定义

到此为止，我们已经讨论了定义集合的许多方法：有时仅仅使用名字作为抽象的对象就可以了；有时通过列举它们的所有元素用外延法定义一个集合；而有时，使用集合描述就足够了。在后面的关系、函数和序列各章所遇到的结构也都能够通过集合描述来定义，正如我们将会看到的那样，这些结构都是以集合的概念为基础的。然而，通常，特别是在处理复杂结构时，当必须提供复杂系统的数学公式时，集合描述是把对象转化成描述的一种特别繁杂的方法。另外，在某些情况下，声明是不能够使用集合描述来完成的。

为此，我们介绍公理定义(axiomatic definition)的概念。公理定义的形式与集合描述的形式一样，具有一个声明部分和一个谓词部分，但是没有项目部分。事实上，可以在公理定义的谓词部分使用集合描述，但是，反过来是不行的。

公理定义的结构如下所示。

```
| 声明
|------
| 谓词
```

声明部分提供被声明的对象的名字和类型，而谓词部分说明对象必须满足的性质。

例 6.16 下面的公理定义引入了自然数的一个集合 X，使得 X 至少含有 3 个元素。

```
| X : ℙ ℕ
|---------
| #X ≥ 3
```

□

公理定义的目的就是把对象带入(转化成)数学描述；我们将在第 14 章看到它的重要性，在那里将描述一些实例，并对它们进行推理。

注意，公理定义比集合描述灵活了许多：后者明确地指出一个元素是所讨论集合的成员的条件，而公理定义只是简单地说明 X 是自然数的一个集，并且在 X 中至少有三个元素，它并没有定义这些元素的实际值。

当然，公理定义并不一定如此松散：可以明确地指出是什么构成了 X；下面是 X 的更严格的定义。

```
| X : ℙ ℕ
|---------
| X = {17, 19, 23}
```

我们再来看看另一个极端的例子，可以将谓词部分完全省略，从而简单地引入自然数的一个集合 X。

```
| X : ℙ ℕ
```

这里，和集合描述的情况一样，默认的谓词是 true。

练习 6.26 对于下面的公理定义，给出被声明的对象的值。

1.

```
| x : ℕ
|-------
| x < 1
```

2.

```
| x : ℙ ℕ
|---------
| # x = 0
```

3.

$$\frac{x: \mathbb{N} \times \mathbb{N}}{(x.1 = x.2) \wedge (x.1 = 7 \text{ div } 2)}$$

□

练习 6.27 使用公理定义，定义下列集合。

1. x 的类型为 $\mathbb{N}$，值为 42。

2. X 的类型为 $\mathbb{P}\,\mathbb{N}$，X 至少包含一个元素而且它不包含 0。

3. X 的类型为 $\mathbb{N} \times \mathbb{N}$，$X$ 中的每一个序偶的第一个成分不等于第二个成分。 □

6.8 附加练习

练习 6.28 假设有类型 *Dog* 和适当的谓词，使用集合描述定义下列集合。

1. 嗅觉好的狗的集合。

2. 受过良好训练的狗的集合。

3. 小狗的集合。

4. 嗅觉好且受过良好训练的狗的集合。

5. 嗅觉好、受过良好训练的小狗的集合。

练习 6.29 假设有类型 *Dog* 和 *Person*，以及适当的谓词和运算，使用集合描述定义下列集合。

1. 嗅觉好且受过良好训练的狗的年龄的集合。

2. 嗅觉好且受过良好训练的狗的主人的集合。

3. 嗅觉好且受过良好训练的狗的主人的地址的集合。

练习 6.30 假设有类型 *Dog*、*Person* 和 *Age*，以及适当的谓词和运算，使用集合描述定义下列集合。

1. 类型 $Dog \times Person$ 的所有序偶的集合。

2. 类型 $Dog \times Person$ 的满足 d 是小狗且 p 拥有 d 的所有序偶(d, p)的集合。

3. 类型 $Dog \times Person$ 的满足 d 是小狗、p 是小孩且 p 拥有 d 的所有序偶(d, p)的集合。

4. 类型 $Dog \times Age$ 的满足 d 是嗅觉好的狗且 a 是 d 的主人的年龄的所有序偶(d, a)的集合。

5. 类型$(Dog \times Age) \times (Person \times Age)$的满足 a_1 是 d 的年龄，a_2 是 p 的年龄，p 拥有 d 的所有序偶$((d, a_1), (p, a_2))$的集合。

练习 6.31 给定下列集合。

$D = \{(\text{fido}, \text{ken}), (\text{spot}, \text{barry}), (\text{barker}, \text{nigel}), (\text{lassie}, \text{david})\}$

使用 D、集合描述和成分筛选来定义下列集合。

1. $\{\text{fido}, \text{spot}, \text{barker}, \text{lassie}\}$

2. $\{\text{ken}, \text{barry}, \text{nigel}, \text{david}\}$

3. $\{((\text{fido}, \text{ken}), \text{ken}), ((\text{spot}, \text{barry}), \text{barry}), ((\text{barker}, \text{nigel}), \text{nigel}), ((\text{lassie}, \text{david}), (\text{david})\}$

4. $\{(\text{ken}, \text{fido}), (\text{barry}, \text{spot}), (\text{nigel}, \text{barker}), (\text{david}, \text{lassie})\}$

练习 6.32 列举下列集合的元素。

1. $\{n: \mathbb{N} \mid n > 1 \wedge n < 4\}$

2. $\{n: \mathbb{N} \mid n > 1 \wedge n < 4 \cdot n^2\}$

3. $\{n: \mathbb{N} \mid n > 1 \wedge n < 4 \cdot (n, n)\}$

4. $\{m, n: \mathbb{N} \mid m + n = 4\}$

5. $\{m, n: \mathbb{N} \mid m + n = 4 \cdot n\}$

练习 6.33 使用集合描述定义下列集合。

1. 0 与 100 间(包括端点)的自然数的集合。

2. 偶整数的集合。

3. 联合国成员国的集合。

练习 6.34

1. 使用集合描述定义集合 $A = \{1, 2\}$。

2. 使用集合描述定义集合 $B = \{2, 3\}$。

3. 使用集合描述定义集合 $id = \{(1, 1), (2, 2), (3, 3), (4, 4)\}$。

4. 使用集合描述定义集合 $double=\{(1,2),(2,4),(3,6),(4,8)\}$。

练习 6.35 ×满足结合律吗?

练习 6.36 ×满足交换律吗?

练习 6.37 ×是幂等的吗?

练习 6.38 证明下式成立。

$A\times(B\cup C)=(A\times B)\cup(A\times C)$

练习 6.39 证明下式成立。

$(A\cap B)\times C=(A\times C)\cap(B\times C)$

练习 6.40 给定集合 C 如下所示。

$C=\{$(琼, no, 14), (戴夫, yes, 64), (吉姆, yes, 29)$\}$

计算下列集合。

1. $\{c: C \mid c.2=\text{yes}\}$
2. $\{c: C \mid c.2=\text{yes} \wedge c.3=14\}$
3. $\{c: C \mid c.2=\text{no} \cdot c.3\}$

练习 6.41 下列集合中哪些等于空集?

1. $\{n: \mathbb{N} \mid n<0\}$
2. $\{n: \mathbb{N} \mid n\neq n\}$
3. $\{n: \mathbb{N} \mid n=n+1\}$
4. $\{n: \mathbb{N} \mid n\leqslant n-1\}$

练习 6.42 给定集合 $People=\{$贝基, 劳拉, 简$\}$, 使用公理定义来定义下列 $People$ 的子集。

1. 集合 A: 只包含贝基和劳拉的集合。
2. 集合 B: 若 B 非空则它一定包含简。
3. 集合 C: 若简在该集合出现, 那么劳拉也一定出现。
4. 集合 D: 序偶的非空集合。
5. 集合 E: 包含元素(劳拉, 简)的序偶的集合。
6. 集合 F: 只有贝基可能出现在第一个成分的序偶的集合。

6.9 练习解答

6.1 1, 4, 6 是∩的合法运用。

6.2 2, 4 是∩的合法运用。

6.3 2, 4, 5, 7, 9 是∈或⊆的合法运用。

6.4 2 和 3 有定义, 1 和 4 未定义。

6.5

1. $A=\{0, 1, 2, 3, 4, 5, 6\}$
2. $B=\{0, 3, 6, 9\}$
3. $C=\{2, 3\}$
4. $D=\varnothing$

6.6

1. $\{c: Car \mid c$ 是红色的$\}$
2. $\{p: People \mid p$ 是素食者$\}$
3. $\{c: Car \mid c$ 有天窗$\}$
4. $\{p: People \mid p$ 是富人$\}$

6.7

1. $A=\{0, 1, 4, 9, 16, 25, 36\}$
2. $B=\{0, 3, 6, 9\}$
3. $C=\{1\}$
4. $D=\varnothing$

6.8

1. $\{c: Car \mid c$ 是红色的 $\cdot registration(c)\}$
2. $\{p: People \mid p$ 是素食者 $\cdot age(p)\}$
3. $\{c: Car \mid c$ 有天窗 $\cdot owner(c)\}$
4. $\{p: People \mid p$ 是富人 $\cdot address(p)\}$

6.9

1. $A=\{0, 1, 4, 9, 16, 25, 36, 49, 64, \cdots$
2. $B=\{0, 1, 2, 3, 4, 5, 6, 7, 8, \cdots$
3. $C=\{0, 0, 1, 1, 2, 2, 3, 3, 4, \cdots$
4. $D=\{1, 2, 3, 4, 5, 6, 7, 8, 9, \cdots$

6.10

1. $\{c: Car \cdot registration(c)\}$
2. $\{p: People \cdot age(p)\}$
3. $\{c: Car \cdot owner(c)\}$
4. $\{p: People \cdot address(p)\}$

6.11

1. n
2. m

6.12

1. {0, 1, 2, 3, 4, 5}
2. {1}
3. {4, 5}
4. {0, 1, 2, 3}

6.13

1. {7, 8, 9, 10}
2. {2, 4, 6, 8, 10}
3. {3, 6, 9}

6.14

1. $A\times B$
2. $A\times A$
3. $B\times B$
4. $B\times A$
5. $B\times B$
6. $B\times(A\times B)$

6.15

1. {(0, 0), (0, 1), (1, 0), (1, 1)}
2. {(0, yes), (0, no), (1, yes), (1, no)}
3. {(yes, 0), (yes, 1), (no, 0), (no, 1)}
4. {(yes, yes), (yes, no), (no, yes), (no, no)}
5. {(0, ∅), (0, {yes}), (0, {no}), (0, {yes, no}), (1, ∅), (1, {yes}), (1, {no}), (1, {yes, no})}
6. ∅

6.16

1. (0, yes)
2. (0, {yes})
3. (0, ∅)
4. ({0}, yes)
5. ({0}, {yes})
6. ({0}, ∅)
7. {(0, yes)}
8. {(0, {yes})}
9. {(0, ∅)}

6.17

1. $3\times2=6$
2. $2\times3=6$
3. $3\times0=0$
4. $3\times2^2=12$
5. $3\times2^0=3$
6. $2^6=64$
7. $2^0=1$
8. $2^{2^2\times2^3}=2^{32}$

6.18

1. n
2. (m, n)
3. (m, n)

6.19

1. {(迈克尔，爱米丽)，(迈克尔，拉切尔)，(迈克尔，安娜)，(约翰，爱米丽)，(约翰，拉切尔)，(约翰，安娜)}
2. {(爱米丽，迈克尔)，(爱米丽，约翰)，(拉切尔，迈克尔)，(拉切尔，约翰)，(安娜，迈克尔)，(安娜，约翰)}
3. {(迈克尔，迈克尔)，(迈克尔，约翰)，(约翰，迈克尔)，(约翰，约翰)}
4. {(迈克尔，约翰)，(约翰，迈克尔)}
5. {(迈克尔，迈克尔)，(约翰，约翰)}
6. {迈克尔，约翰}
7. {(约翰，爱米丽)，(约翰，拉切尔)，(约翰，安娜)}
8. {爱米丽，拉切尔，安娜}

6.20 3和5的“\”的运用是合法的，其他都是不合法的。

6.21

1. {(1, 2), (1, 3), (2, 2), (2, 3)}
2. {(0, 2), (0, 3), (1, 2), (1, 3), (2, 2), (2, 3), (3, 2), (3, 3)}
3. {(0, 2), (0, 3)}

6.22

1. {(杰克，60，jack@the_hill.com)，(杰克，60，jill@the_hill.com)，(杰克，65，jack@the_hill.com)，(杰克，65，jill@the_

hill. com)，(吉尔，60，jack@the_ hill. com)，(吉尔，60，jill@the_ hill. com)，(吉尔，65，jack@the_ hill. com)，(吉尔，65，jill@the_ hill. com)}

2. {(杰克，(60，jack@ the_hill. com))，(杰克，(60，jill @ the_hill. com))，(杰克，(65，jack@ the_hill. com))，(杰克，(65，jill@the_hill. com))，(吉尔，(60，jack@ the_ hill. com))，(吉尔，(60，jill @ the_ hill. com))，(吉尔，(65，jack @ the_ hill. com))，(吉尔，(65，jill@the_hill. com))}

3. {((杰克，60)，jack@the_hill. com)，((杰克，60)，jill@the_hill. com)，((杰克，65)，jack@the_hill. com)，((杰克，65)，jill @the_hill. com)，((吉尔，60)，jack@the_ hill. com)，((吉尔，60)，jill@the_hill. com)，((吉尔，65)，jack@the_hill. com)，((吉尔，65)，jill@the_hill. com)}

6.23

1. {(60，60，60)，(60，60，65)，(60，65，60)，(60，65，65)，(65，60，60)，(65，60，65)，(65，65，60)，(65，65，65)}

2. {(杰克，杰克，杰克)，(吉尔，吉尔，吉尔)}

3. {(杰克，(杰克，杰克))，(杰克，(杰克，吉尔))}

4. {((杰克，杰克)，(杰克，杰克))，((杰克，吉尔)，(杰克，吉尔))，((吉尔，杰克)，(吉尔，杰克))，((吉尔，吉尔)，(吉尔，吉尔))}

5. {(60，jack@ the_hill. com)，(60，jill @ the_hill. com)，(65，jack @ the_ hill. com)，(65，jill@the_hill. com)}

6. {(杰克，65，jack@the_hill. com)，(杰克，65，jill@the_hill. com)}

6.24

1. 爱米丽
2. 金鱼
3. 邓肯
4. (格比尔，红色)
5. (爱米丽，金鱼)
6. (邓肯，格比尔)
7. 金鱼

6.25

1. {迈克尔，约翰}
2. {爱米丽，拉切尔，安娜}
3. {(爱米丽，迈克尔)，(拉切尔，迈克尔)，(安娜，迈克尔)}
4. {爱米丽，拉切尔，安娜}
5. {迈克尔}

6.26

1. $x=0$
2. $x=\varnothing$
3. $x=(3, 3)$

6.27

1.

$$\begin{array}{|l} x:\mathbb{N} \\ \hline x=42 \end{array}$$

2.

$$\begin{array}{|l} X:\mathbb{PN} \\ \hline X\neq\varnothing\wedge 0\notin X \end{array}$$

3.

$$\begin{array}{|l} X:\mathbb{N}\times\mathbb{N} \\ \hline X\subseteq\{x, y:\mathbb{N} \mid x\neq y\} \end{array}$$

6.28

1. $\{d: Dog \mid d$ 嗅觉好$\}$
2. $\{d: Dog \mid d$ 受过良好的训练$\}$
3. $\{d: Dog \mid d$ 是小狗$\}$
4. $\{d: Dog \mid d$ 嗅觉好$\wedge d$ 受过良好的训练$\}$
5. $\{d: Dog \mid d$ 嗅觉好$\wedge d$ 受过良好的训练$\wedge d$ 是小狗$\}$

6.29

1. $\{d: Dog \mid d$ 嗅觉好 $\wedge d$ 受过良好的训练 $\cdot age(d)\}$

2. $\{d: Dog; p: Person \mid d$ 嗅觉好 $\wedge d$ 受过良好的训练 $\wedge p$ 拥有 $d \cdot p\}$

3. $\{d: Dog; p: Person \mid d$ 嗅觉好 $\wedge d$ 受过良好的训练 $\wedge p$ 拥有 $d \cdot address(p)\}$

6.30

1. $\{d: Dog; p: Person\}$

2. $\{d: Dog; p: Person \mid d$ 是小狗 $\wedge p$ 拥有 $d\}$

3. $\{d: Dog; p: Person \mid d$ 是小狗 $\wedge p$ 是小孩 $\wedge p$ 拥有 $d\}$

4. $\{d: Dog; p: Person \mid d$ 嗅觉好 $\wedge p$ 拥有 $d \cdot (d, age(p))\}$

5. $\{d: Dog; p: Person \mid p$ 拥有 $d \cdot ((d, age(d)), (p, age(p)))\}$

6.31

1. $\{d: D \cdot d.1\}$

2. $\{d: D \cdot d.2\}$

3. $\{d: D \cdot (d, d.2)\}$

4. $\{d: D \cdot (d.2, d.1)\}$

6.32

1. $\{2, 3\}$

2. $\{4, 9\}$

3. $\{(2, 2), (3, 3)\}$

4. $\{(0, 4), (1, 3), (2, 2), (3, 1), (4, 0)\}$

5. $\{0, 1, 2, 3, 4\}$

6.33

1. $\{n: \mathbb{N} \mid n \leqslant 100\}$

2. $\{n: \mathbb{Z} \mid n \bmod 2 = 0\}$

3. $\{c: Country \mid c$ 是联合国成员国$\}$

6.34

1. $A = \{n: \mathbb{N} \mid n > 0 \wedge n < 3\}$

2. $B = \{n: \mathbb{N} \mid n > 1 \wedge n < 4\}$

3. $id = \{n: \mathbb{N} \mid n > 0 \wedge n < 5 \cdot (n, n)\}$

4. $double = \{n: \mathbb{N} \mid n > 0 \wedge n < 5 \cdot (n, n \times 2)\}$

6.35 不满足结合律。考虑集合 $A = \{1, 2\}$、$B = \{a\}$ 和 $C = \{$琼$\}$。在此，有

$$A \times (B \times C) = \{a: A; d: B \times C \cdot (a, d)\}$$
$$= \{a: A; b: B; c: C \cdot (a, (b, c))\}$$
$$= \{(1, (a, \text{琼})), (2, (a, \text{琼}))\}$$
$$(A \times B) \times C = \{d: A \times B; c: C \cdot (d, c)\}$$
$$= \{a: A; b: B; c: C \cdot ((a, b), c)\}$$
$$= \{((1, a), \text{琼}), ((2, a), \text{琼})\}$$

这里，集合 $A \times (B \times C)$ 与 $(A \times B) \times C$ 不相等。因此，$\times$ 不满足结合律。

6.36 不满足交换律。考虑集合 $A = \{1, 2\}$ 和 $B = \{a\}$。在此，有

$$A \times B = \{a: A; b: B \cdot (a, b)\}$$
$$= \{(1, a), (2, a)\}$$
$$B \times A = \{b: B; a: A \cdot (b, a)\}$$
$$= \{(a, 1), (a, 2)\}$$

这里，集合 $A \times B$ 与 $B \times A$ 不相等。因此，$\times$ 不满足交换律。

6.37 不是幂等的。考虑集合 $A = \{1, 2\}$。在此，有

$$A \times A = \{a: A; b: A \cdot (a, b)\}$$
$$= \{(1, 1), (1, 2), (2, 1), (2, 2)\}$$

这里，集合 A 与 $A \times A$ 不相等。因此，$\times$ 不是幂等的。

6.38 设有相应类型的序偶(x, y)。有

$$(x, y) \in A \times (B \cup C)$$
$$\Leftrightarrow x \in A \wedge y \in B \cup C$$
$$\Leftrightarrow x \in A \wedge (y \in B \vee y \in C)$$
$$\Leftrightarrow (x \in A \wedge y \in B) \vee (x \in A \wedge y \in C)$$
$$\Leftrightarrow (x, y) \in A \times B \vee (x, y) \in A \times C$$
$$\Leftrightarrow (x, y) \in (A \times B) \cup (A \times C)$$

6.39 设有相应类型的序偶(x, y)。有

$$(x, y) \in (A \cap B) \times C$$
$$\Leftrightarrow (x \in A \cap B) \wedge y \in C$$
$$\Leftrightarrow x \in A \wedge x \in B \wedge y \in C$$
$$\Leftrightarrow x \in A \wedge x \in B \wedge y \in C \wedge y \in C$$
$$\Leftrightarrow x \in A \wedge y \in C \wedge x \in B \wedge y \in C$$
$$\Leftrightarrow (x, y) \in A \times C \wedge (x, y) \in B \times C$$
$$\Leftrightarrow (x, y) \in (A \times C) \cap (B \times C)$$

6.40

1. {(戴夫，yes，64)，(吉姆，yes，29)}
2. $\varnothing$
3. {14}

6.41 它们全与空集相等。

6.42

1.

$$\begin{array}{|l} A : \mathbb{P}People \\ \hline A = \{贝基，劳拉\} \end{array}$$

2.

$$\begin{array}{|l} B : \mathbb{P}People \\ \hline B \neq \varnothing \Rightarrow 简 \in B \end{array}$$

3.

$$\begin{array}{|l} C : \mathbb{P}People \\ \hline 简 \in C \Rightarrow 劳拉 \in C \end{array}$$

4.

$$\begin{array}{|l} D : People \times People \\ \hline D \neq \varnothing \end{array}$$

5.

$$\begin{array}{|l} E : People \times People \\ \hline (劳拉，简) \in E \end{array}$$

6.

$$\begin{array}{|l} F : People \times People \\ \hline F = \varnothing \\ \vee \\ F \subseteq \{p : People \times People \mid p.1 = 贝基\} \end{array}$$

第7章 谓词逻辑

7.1 量词的需要

真值表是一种相当令人满意的方法，它能够确切地决定由少量原子命题组成的命题的真假性。然而，正如我们看到的那样，当命题中原子命题的数目超过4的时候，命题的真值表的构建过程就会变得相当复杂。例如，命题$(p \vee \neg q) \wedge (r \vee s)$的真值表有16行，而命题$(p \vee \neg q) \wedge (r \vee s) \wedge \neg t$的真值表有32行(回顾一下，由$n$个原子命题组成的命题的真值表有$2^n$行)。

另外，假设我们想要使用命题逻辑对陈述"45是素数，或者46是素数，或者47是素数，或者48是素数，等等"公式化。

假设存在关系*prime*使得$prime(x)$为真当且仅当$x \in \mathbb{N}$是一个素数。这时，可以通过命题逻辑如下表示这一陈述。

$prime(45) \vee prime(46) \vee prime(47) \vee prime(48) \vee \cdots$

我们可能需要一个非常大的真值表来决定这个陈述的真假性。这里，问题不在于真值表，而在于命题逻辑本身：无法使用命题逻辑来形式地写出这样的命题，更何况讨论它的真假性(即便我们知道这个陈述是真的)。因此，需要一种表示这种陈述的更简洁的方法；谓词逻辑(predicate logic)为我们提供了这样的方法。

7.2 全称量词

正如我们在第4章看到的那样，$\bigcap$是$\cap$的一个广义形式。类似地，这里的第一个谓词逻辑运算符——全称量词$\forall$是合取($\wedge$)的一种广义形式。全称量词使得我们可以刻画"对于所有"或者"对于每一个"这种形式的命题。例如，陈述"每一个自然数都大于或者等于零"可以如下形式地表示。

$\forall n: \mathbb{N} \cdot n \geqslant 0$

这一形式的一个量化(quantified)陈述由三个部分组成：量词(quantifier)(在这里是全称量词$\forall$)；量化变元(quantification)，它描述陈述所关心的变元和变元类型；谓词(predicate)，它通常是关于量化变元的陈述。因此，至少在目前可以认为每个谓词逻辑的陈述有如下形式。

quantifier quantification $\cdot$ predicate

当然，上面的陈述逻辑等值于下面的合取式(而且更加简洁)。

$(0 \geqslant 0) \wedge (1 \geqslant 0) \wedge (2 \geqslant 0) \wedge (3 \geqslant 0) \wedge \cdots$

与集合论中的声明一样，我们不局限于只声明一个变元。例如，下面的谓词表示，给定两个自然数m和n，总是有或者$m<n$，或者$m=n$，或者$m>n$。

$\forall m, n: \mathbb{N} \cdot m<n \vee m=n \vee m>n$

注意，这里对声明的约定与第6章的约定是一致的；上面的谓词命题可以等价地写成如下形式。

$\forall m: \mathbb{N}; n: \mathbb{N} \cdot m<n \vee m=n \vee m>n$

下面的写法也是正确的。

$\forall m: \mathbb{N} \cdot \forall n: \mathbb{N} \cdot m<n \vee m=n \vee m>n$

另外，下面的写法也是正确的。

$\forall n: \mathbb{N} \cdot \forall m: \mathbb{N} \cdot m<n \vee m=n \vee m>n$

此外，我们不局限于量化一种类型的变元。下面的全称量化的陈述表明小狗和小孩间的某种关系。

$\forall p: People; d: Dog \cdot p$是小孩$\wedge p$拥有$d \Rightarrow d$是小狗

练习7.1 用自然语言说明下列陈述的含义。

1. $\forall d$: $Dog \cdot d$ 嗅觉好

2. $\forall d$: $Dog \cdot$ (d 受过良好的训练 $\vee$ d 是小狗)

3. ($\forall d$: $Dog \cdot d$ 受过良好的训练) $\vee$ ($\forall d$: $Dog \cdot d$ 是小狗)

4. $\forall d$: $Dog \cdot$ (d 是小狗 $\Rightarrow$ d 受过良好的训练)

5. ($\forall d$: $Dog \cdot d$ 是小狗) $\Rightarrow$ ($\forall d$: $Dog \cdot d$ 受过良好的训练) □

练习 7.2 考虑练习 7.1 中的陈述 4 和 5，它们是否逻辑等值？证明你的结论。 □

练习 7.3 假设有集合 *People*，使用这一集合与谓词"喜欢佳发蛋糕"和"是素食者"，运用谓词逻辑公式化下列陈述。

1. "每个人都喜欢佳发蛋糕。"

2. "所有的素食者都不喜欢佳发蛋糕。"

3. "每个人都或者喜欢佳发蛋糕或者是素食者。"

4. "或者每个人都喜欢佳发蛋糕或者每个人都是素食者。"

5. "每个人都喜欢佳发蛋糕且是素食者。"

6. "每个人都喜欢佳发蛋糕且每个人都是素食者。" □

练习 7.4 考虑练习 7.3 的陈述 3 和 4，它们是否逻辑等值？证明你的结论。 □

注意，在上面的练习中，尽管陈述 3 并不逻辑等值于陈述 4，但是陈述 4 的确蕴含陈述 3。这一事实可以一般地表述如下。

规则 7.1

$((\forall x: X \cdot p) \vee (\forall x: X \cdot q)) \Rightarrow (\forall x: X \cdot p \vee q)$ □

例 7.1 下面的陈述对人的给定集合成立。

($\forall p$: $Person \cdot p$ 是高个子) $\vee$ ($\forall p$: $Person \cdot p$ 强壮)) $\Rightarrow$ ($\forall p$: $Person \cdot p$ 是高个子 $\vee$ p 强壮) □

练习 7.5 考虑练习 7.3 中的陈述 5 和 6，它们是否逻辑等值？证明你的结论。 □

上面的逻辑等值可以一般地表述如下。

规则 7.2

$(\forall x: X \cdot p \wedge q) \Leftrightarrow ((\forall x: X \cdot p) \wedge (\forall x: X \cdot q))$ □

例 7.2 下面的陈述对人的给定集合成立。

($\forall p$: $Person \cdot p$ 是高个子) $\wedge$ ($\forall p$: $Person \cdot p$ 强壮)) $\Leftrightarrow$ ($\forall p$: $Person \cdot p$ 是高个子 $\wedge$ p 强壮) □

这一等值式成立的原因是，全称量词充当着"超级合取"(big conjunction)的角色。回想一下，当我们通过合取运算符把命题连接起来的时候，组成合取式的各命题的顺序是无关紧要的。同样地，全称量词对于合取运算符满足分配律：这与运用"并且"(and)的顺序无关。

练习 7.6 设有集合 *People* 和序偶 $likes \in People \times People$ 的集合，两个集合的定义如下。

People={吉姆，琼，瑞克}

likes={(吉姆，琼)，(琼，吉姆)，
(瑞克，琼)，(琼，瑞克)}

确定下列陈述的真假性。

1. $\forall l$: $likes \cdot l.1$=瑞克

2. $\forall l$: $likes \cdot \neg(l.1$=瑞克)

3. $\forall p, q$: $People \cdot (p, q) \in likes \Rightarrow (q, p) \in likes$

4. $\forall l$: $likes \cdot l.1$=瑞克 $\Rightarrow l.2$=琼 □

7.3 存在量词

第二个量词是存在量词(existential quantifier)，记作 $\exists$。全称量化变元用于声明一个特定的性质对集合的每一个元素都成立。而存在量词用于声明一个特定的性质对集合的某些元素(至少有一个)成立。另外，就像 $\forall$ 可以看成是 $\wedge$ 的广义形式一样，$\exists$ 可以看成是 $\vee$ 的广义形式。

例 7.3 陈述"有些自然数能被 3 整除"

可以表述如下。

$\exists n: \quad \cdot n \bmod 3=0$ □

注意，带存在量词的陈述的形式与全称量词陈述的形式类似，其形式如下。

quantifier quantification · predicate

练习 7.7 使用自然语言说明下列陈述的含义。

1. $\exists d$: $Dog \cdot d$ 嗅觉好

2. $\exists d$: $Dog \cdot$ (d 受过良好的训练 $\vee$ d 是小狗)

3. $\exists d$: $Dog \cdot$ (d 是小狗 $\Rightarrow$ d 受过良好的训练)

4. ($\exists d$: $Dog \cdot d$ 是小狗) $\Rightarrow$ ($\exists d$: $Dog \cdot d$ 受过良好的训练) □

练习 7.8 假设有集合 $People$，使用这一集合与谓词"喜欢佳发蛋糕"和"是素食者"，运用谓词逻辑公式化下列陈述。

1. "有人喜欢佳发蛋糕。"

2. "有些素食者不喜欢佳发蛋糕。"

3. "有人或者喜欢佳发蛋糕或者是素食者。"

4. "或者有人喜欢佳发蛋糕或者有人是素食者。"

5. "有人喜欢佳发蛋糕且是素食者。"

6. "有人喜欢佳发蛋糕且有人是素食者。" □

练习 7.9 考虑练习 7.8 的陈述 3 和 4，它们是否逻辑等值？证明你的结论。 □

上面的等值式可以一般地表述如下。

规则 7.3

$(\exists x: X \cdot p \vee q) \Leftrightarrow ((\exists x: X \cdot p) \vee (\exists x: X \cdot q))$ □

例 7.4 下面的陈述对汽车的给定集合 Car 成立。

$(\exists c: Car \cdot fast(c) \vee small(c)) \Leftrightarrow ((\exists c: Car \cdot fast(c)) \vee (\exists c: Car \cdot small(c)))$ □

这一规则成立的理由与规则 7.2 成立的理由类似。回想一下，存在量词充当着"超级析取"的角色。与合取的情况一样，当我们通过析取运算符把命题连接起来的时候，组成析取式的各命题的顺序是无关紧要的。因此，存在量词对于析取运算符满足分配律。

练习 7.10 考虑练习 7.8 的陈述 5 和 6，它们是否逻辑等值？证明你的结论。 □

注意，在上面的练习中，陈述 5 蕴含陈述 6。这一事实可以一般地表述如下。

规则 7.4

$(\exists x: X \cdot p \wedge q) \Rightarrow ((\exists x: X \cdot p) \wedge (\exists x: X \cdot q))$ □

例 7.5 下面的陈述对汽车的给定集合 Car 成立。

$(\exists c: Car \cdot fast(c) \wedge small(c)) \Rightarrow ((\exists c: Car \cdot fast(c)) \wedge (\exists c: Car \cdot small(c)))$ □

练习 7.11 设有集合 $People$ 和序偶 $likes \in People \times People$ 的集合，两个集合的定义如下。

$People=\{$吉姆，琼，瑞克$\}$

$likes=\{$(吉姆，琼)，(琼，吉姆)，(瑞克，琼)，(琼，瑞克)$\}$

确定下列陈述的真假性。

1. $\exists l$: $likes \cdot l.1=$瑞克

2. $\exists l$: $likes \cdot \neg(l.1=$瑞克$)$

3. $\exists p, q$: $People \cdot (p, q) \in likes \Rightarrow (q, p) \in likes$

4. $\exists l$: $likes \cdot l.1=$瑞克$\Rightarrow l.2=$琼 □

练习 7.12 下面的命题总为真吗？

$(\forall x: X \cdot p) \Rightarrow (\exists x: X \cdot p)$ □

练习 7.13 下面的命题总为真吗？

$(\exists x: X \cdot p) \Rightarrow (\forall x: X \cdot p)$ □

练习 7.14

1. 找出一个使下面的陈述为真的谓词 p。

$(\exists n: \mathbb{N} \cdot p) \Rightarrow (\forall n: \mathbb{N} \cdot p)$

2. 找出一个使上面的陈述为假的谓词 p。 □

7.4 可满足性和有效性

可以认为谓词 $n>3$ 既不为真也不为假，除非我们知道 n 的值：如果 n 的值小于4，那么这个陈述等值于 false，否则等值于 true。

称谓词 p 是有效的(valid)，当且仅当它对相应类型的所有可能值都为真。也就是说，如果谓词 p 与类型 X 的变元 x 有关，那么 p 是有效的，当且仅当 $\forall x: X \cdot p$ 为真。

例 7.6 谓词 $n \geqslant 0$ 是有效的，因为 $\forall n: \mathbb{N} \cdot n \geqslant 0$ 等值于 true。 □

称谓词 p 是可满足的(satisfiable)，当且仅当它对相应类型的某些值为真。也就是说，如果谓词 p 与类型 X 的变元 x 有关，那么 p 是可满足的，当且仅当 $\exists x: X \cdot p$ 为真。

例 7.7 谓词 $n>0$ 是可满足的，因为 $\exists n: \mathbb{N} \cdot n>0$ 等值于 true。 □

最后，称谓词 p 是不可满足的(unsatisfiable)，当且仅当它对相应类型的所有可能值都为假。也就是说，如果谓词 p 与类型 X 的变元 x 有关，那么 p 是不可满足的，当且仅当 $\exists x: X \cdot p$ 为假。

例 7.8 谓词 $n \neq n$ 是不可满足的，因为 $\exists n: \mathbb{N} \cdot n \neq n$ 等值于 false。 □

注意有效、可满足和不可满足谓词与重言式、不定式和矛盾式之间的相似之处：有效谓词和重言式总为真；可满足谓词和不定式有时为真，有时为假；不可满足谓词和矛盾式从不为真。

练习 7.15 假设有集合 $People=\{$亚历克斯，安妮，亨利$\}$。对于下列谓词，确定哪些是有效的，哪些是可满足的，哪些是不可满足的？可以认为 x 和 y 是属于类型 $People$ 的。

1. $x=$亚历克斯
2. $x=$亚历克斯 $\wedge x=$安妮
3. $x=$亚历克斯 $\wedge y=$安妮
4. $x \in People$
5. $x \in \varnothing$ □

7.5 量词的否定

就像我们已经看到的，通过命题逻辑运算符 $\wedge$、$\vee$、$\Rightarrow$ 和 $\Leftrightarrow$，可以自由地将量化(quantified)的陈述组合起来。还可以将否定运算符 $\neg$ 运用于量化的表达式，这就是本节的内容。

陈述“有人喜欢布莱恩”可以使用谓词逻辑表示如下。

$\exists p: Person \cdot p$ 喜欢布莱恩

如果想否定该表达式，可以写成如下的表达式。

$\neg \exists p: Person \cdot p$ 喜欢布莱恩

用自然语言，这一表达式可以叙述为“没有人喜欢布莱恩”。

从逻辑上来说，说“没有人喜欢布莱恩”与说“每个人都不喜欢布莱恩”是等值的。量词的否定的行为方式正是这样：就如在自然语言中“没有人喜欢布莱恩”与“每个人都不喜欢布莱恩”等值一样，在谓词逻辑中，

$\neg(\exists p: Person \cdot p$ 喜欢布莱恩$)$

和

$\forall p: Person \cdot \neg(p$ 喜欢布莱恩$)$

是等值的。

再举一个例子，陈述“每个人都喜欢布莱恩”可以使用谓词逻辑表示如下。

$\forall p: Person \cdot p$ 喜欢布莱恩

如果想否定这个表达式，可以书写如下。

$\neg \forall p: Person \cdot p$ 喜欢布莱恩

这在自然语言中可以陈述为“不是每个人都喜欢布莱恩”。

从逻辑上来说，说“不是每个人都喜欢布莱恩”与说“有人不喜欢布莱恩”是等值的。

就如在自然语言中"不是每个人都喜欢布莱恩"与"有人不喜欢布莱恩"等值一样，在谓词逻辑中，

$\neg(\forall p: Person \cdot p$ 喜欢布莱恩)

和

$\exists p: Person \cdot \neg(p$ 喜欢布莱恩)

等值。这些等值关系可以一般地表述如下。

规则 7.5

$\neg(\exists x: X \cdot p) \Leftrightarrow \forall x: X \cdot \neg p$

$\neg(\forall x: X \cdot p) \Leftrightarrow \exists x: X \cdot \neg p$ □

从结果上看，当否定被运用于量化表达式时，随着它向内部移动，它会"翻转"(flip)量词(即，否定把所有的全称量词改成存在量词，反之亦然，并且否定所有的谓词)。当表达式中有多个量词时，这一行为方式就会更加突显出来。例如，考虑下面的陈述。

$\forall x: X \cdot \exists y: Y \cdot \forall z: Z \cdot p(x, y, z)$

在此，可以如下确定这一表达式的否定的结果。

$\neg \forall x: X \cdot \exists y: Y \cdot \forall z: Z \cdot p(x, y, z)$

$\Leftrightarrow \exists x: X \cdot \neg \exists y: Y \cdot \forall z: Z \cdot p(x, y, z)$

$\Leftrightarrow \exists x: X \cdot \forall y: Y \cdot \neg \forall z: Z \cdot p(x, y, z)$

$\Leftrightarrow \exists x: X \cdot \forall y: Y \cdot \exists z: Z \cdot \neg p(x, y, z)$

随着否定"扫过"整个表达式，它反转路途中出现的每一个量词。

练习 7.16 给出下列陈述的否定。

1. "每个人都喜欢每个人。"
2. "每个人都喜欢某个人。"
3. "某个人喜欢每个人。"
4. "某个人喜欢某个人。" □

练习 7.17 假设有类型 *Person* 和谓词"喜欢"使得 x 喜欢 y 当且仅当 *Person* x 喜欢 *Person* y，使用谓词逻辑写出练习 7.16 的答案的陈述。 □

练习 7.18 不使用¬，给出下列谓词的否定。

1. $\forall x: \mathbb{N} \cdot x = x$
2. $\forall x: \mathbb{N} \cdot \exists y: \mathbb{N} \cdot x \geqslant y$
3. $\forall x: \mathbb{N} \cdot \forall y: \mathbb{N} \cdot x > y \vee x = y \vee x < y$ □

7.6 自由变元和约束变元

谓词 $n>5$ 指出了有关自然数 n 的一些属性：它大于 5。可是，我们不知道 n 的值：n 只是一个位置标志符(placeholder)，或者说是一个变元(variable)，它的值不能仅通过这个谓词来决定。因此，正如我们已经看到的那样，不能确定谓词 $n>5$ 的真假性。

另一方面，考虑陈述 $\exists n: \mathbb{N} \cdot n>5$，它等值于下面的表达式。

$(0>5) \vee (1>5) \vee (2>5) \vee (3>5) \vee (4>5) \vee (5>5) \vee (6>5) \cdots$

这个谓词当然是真的：只要给出至少一个比 5 大的自然数，这个表达式就成立。因此，这个量化了的陈述与没有量化的谓词 $n>5$ 意义完全不同。

在陈述 $\exists n: \mathbb{N} \cdot n>5$ 中，我们称变元 n 是约束的(bound)。这是因为我们知道 n 所扮演的角色：它是一个存在量化变元。另一方面，在陈述 $n>5$ 中，我们只知道 n 是一个整数。在这种情况下，称变元 n 是自由的(free)：我们无法确定甚至是限定它的值。

例 7.9 假设有集合 *People*，其定义如下。

People={罗宾，理查德，罗伯特}

又假设有下面的表达式。

($\forall y$: *People* · y 喜欢罗宾) $\vee$ x 喜欢理查德

在这一表达式中出现了两个变元 y 和 x。前者是全称量化变元，所以它是约束的，而后者是自由的。 □

通常，给定陈述 $\forall x: X \cdot p$，我们说在声明中出现的 x 是一个约束。x 在 p 中的任何出现都是约束出现，而除了 x 之外(并且

不受其他量词约束)的任何变元的任何出现都是自由出现。参看下面的说明示例。

$$(\forall \overbrace{x}^{\text{约束}}: \mathbb{N} \cdot \overbrace{x}^{\text{约束出现}} \geqslant 0 \wedge \overbrace{y}^{\text{自由出现}} \geqslant \overbrace{x}^{\text{约束出现}}) \wedge \overbrace{x}^{\text{自由出现}} = 3$$

注意，在这个表达式中，全称量词的辖域(scope)只能到达右括号：最右边的 x 的出现在全称量词的辖域之外，所以，最后一个 x 的出现是自由的。参看下面的说明示例。

$$(\forall x: \mathbb{N} \cdot \underbrace{x \geqslant 0 \wedge y \geqslant x}_{\text{辖域}}) \wedge x = 3$$

练习 7.19 给出下面谓词的全称量词和存在量词的辖域。

$(\forall x: X \cdot p \wedge (\exists y: Y \cdot q) \wedge r)$ □

练习 7.20 以下表达式中，哪些变元的出现是约束出现，哪些是自由出现？

1. $x \geqslant 0$
2. $\forall x: \mathbb{N} \cdot x \geqslant 0$
3. $\forall x: \mathbb{N} \cdot x \geqslant x$
4. $\forall x: \mathbb{N} \cdot x \geqslant y$
5. $y \geqslant 0 \wedge \forall x: \mathbb{N} \cdot x \geqslant y$
6. $x \geqslant 0 \wedge \forall x: \mathbb{N} \cdot x \geqslant y$
7. $\forall x: \mathbb{N} \cdot \exists y: \mathbb{N} \cdot x \geqslant y$
8. $\forall x: \mathbb{N} \cdot (y = 0 \wedge \exists y: \mathbb{N} \cdot x \geqslant y)$

□

练习 7.21 我们能对练习 7.20 的哪些谓词逻辑表达式赋真值？ □

在讨论完自由变元和约束变元的概念之后，现在我们来形式地刻画谓词与命题之间的微妙的差异。它们之间的差异中最简单的一点在于，谓词是含有自由变元(或"逻辑缺口")的逻辑陈述，而命题则是不含这样的缺口的逻辑陈述。

例 7.10 表达式 $n>7$ 是一个谓词。而 $\exists n: \mathbb{N} \cdot n>7$ 和 $8>7$ 都是命题。 □

练习 7.22 假设理查德和邓肯都是 Person 的元素，下列哪些陈述是谓词，哪些是命题？

1. 理查德比邓肯高。
2. 理查德比 x 高。
3. $\exists x: X \cdot$ 理查德比 x 高。 □

7.7 替换

再次考虑陈述 $\exists n: \mathbb{N} \cdot n>5$。在上一节中介绍过，在这一表达式中，变元 n 受到存在量词的约束：我们可以确定它在整个陈述中所扮演的角色。

另一方面，在无量化的谓词 $n>5$ 中，我们说过变元 n 是自由的：可以假定它取任意整数值，如果这个值小于 6，那么这一陈述的真值将是 false；否则它将是 true。例如，如果 n 取 3，那么这一陈述是 false；另一方面，如果 n 被 $3+4$ 替换，那么这一陈述就等值于 true。

从这个意义上讲，变元 n 是自由的这一事实使我们得出结论：可以用任意的项(term)t 来替换名字(name)n，前提是 n 和 t 是同一类型。这一过程就叫做替换(substitution)：表达式 $p[t/n]$ 表示谓词 p 中的变元名 n 被项 t 所替换。例如，可以用下式来表示，把谓词 $n>5$ 中的 n 替换成 3。

$n>5[3/n]$

这给出下面的结果。

$3>5$

另外，可以用下式表示用 $3+4$ 替换 $n>5$ 中的 n。

$n>5[3+4/n]$

这给出下面的结果。

$3+4>5$

注意，在上面两个例子中，结果表达式已不再是谓词，它们是命题(回忆一下，谓词是至少含有一个自由变元的逻辑表达式，而命题是不含自由变元的逻辑表达式)。

注意，只有自由变元才能进行替换：替换约束变元是无效的。例如，

$$((\exists n: \mathbb{N} \cdot n>5)[3/n]) \Leftrightarrow \exists n: \mathbb{N} \cdot n>5$$

这正是我们所希望的：如果这一替换是

有效的，其结果将是 $\exists n$：$\mathbb{N}\cdot 3>5$，逻辑上说，这与 $\exists n$：$\mathbb{N}\cdot n>5$ 的意义完全不同。

练习 7.23 下列表达式中的哪些变元可以进行替换？

1. $4+3=7-x$
2. $\exists x$：$\mathbb{N}\cdot 4+3=7-x$
3. $\exists x$：$\mathbb{N}\cdot 4+3=7-y$
4. $(\exists x$：$\mathbb{N}\cdot 4+3=7-x\wedge x=y)$
5. $(\exists x$：$\mathbb{N}\cdot 4+3=7-x)\wedge x=y$ □

练习 7.24 确定下列替换的效果。

1. x 高兴[奈杰尔/x]
2. (x 高兴 $\wedge$ y 悲伤)[奈杰尔/x]
3. (x 高兴 $\vee$ x 悲伤)[奈杰尔/x]
4. x 高兴 $\vee$ (x 悲伤[奈杰尔/x])
5. ($\forall x$：*People*·x 高兴)[奈杰尔/x]
6. ($\forall x$：*People*·x 高兴 $\vee$ x 悲伤)[奈杰尔/x]
7. ($\forall x$：*People*·x 高兴) $\vee$ x 悲伤[奈杰尔/x] □

练习 7.25 确定下列替换的效果。

1. $(\forall x$：$\mathbb{N}\cdot x\geqslant 0)[3/x]$
2. $(\forall x$：$\mathbb{N}\cdot x\geqslant y)[3/y]$
3. $(y>0\wedge\forall x$：$\mathbb{N}\cdot x\geqslant y)[3/y]$
4. $(x>0\wedge\forall x$：$\mathbb{N}\cdot x\geqslant y)[3/x]$ □

到此为止，我们只考虑了在表达式中替换一个变元的情况，当然，也可以在表达式中替换多个自由变元；在这样的情况下，我们希望实行多次替换。

考虑谓词“x 高兴 $\wedge$ y 悲伤”，其中，x 和 y 都是自由变元。我们可以如下所示地用奈杰尔替换 x。

(x 高兴 $\wedge$ y 悲伤)[奈杰尔/x]$\Leftrightarrow$奈杰尔高兴 $\wedge$ y 悲伤

现在，再用肯替换 y：

(奈杰尔高兴 $\wedge$ y 悲伤)[肯/y]$\Leftrightarrow$ 奈杰尔高兴 $\wedge$ 肯悲伤

与其分两步完成这些替换，可以如下所示地表示整个替换。

(x 高兴 $\wedge$ y 悲伤)[奈杰尔/x][肯/y]

这表明两个替换，奈杰尔替换 x 和肯替换 y，将依次完成。首先，奈杰尔替换了 x，然后，肯替换了 y。这与上面的两个替换是等效的。

另一方面，如果我们希望这两个替换同时进行，就可以写作：

(x 高兴 $\wedge$ y 悲伤)[奈杰尔/x，肯/y]

在此，这两个替换同时发生且给出相同的结果。

例 7.11 为了说明依次替换和同时替换的区别，考虑下面两个替换的效果。

$(x>3\wedge y>7)[y/x,\ 8/y]$

$\Leftrightarrow y>3\wedge 8>7(x>3\wedge y>7)[y/x][8/y]$

$\Leftrightarrow y>3\wedge y>7[8/y]$

$\Leftrightarrow 8>3\wedge 8>7$

$\Leftrightarrow$true

对于后者，替换的依次性表明被 8 替换的 y 出现两次，而对于前者，被 8 替换的 y 只出现一次。 □

练习 7.26 确定下列替换的效果。

1. $x^2=49[y+3/x][4/y]$
2. $x^2=49[y+3/x,\ 4/y]$
3. $x^2>y[y+3/x][4/y]$
4. $x^2>y[y+3/x,\ 4/y]$ □

替换时必须小心。尤其是当我们用一些变元或含有变元的项来替换其他变元时更要小心。举例说明如下。

$\exists n$：$\mathbb{N}\cdot(n>5\wedge m<5)$

这一陈述的真假性依赖于 m 的值：如果 m 小于 5，那么这一陈述为真，否则为假。

如果在这一陈述中用 n 替换 m，那么结果如下。

$\exists n$：$\mathbb{N}\cdot(n>5\wedge n<5)$

而这一陈述的真假性不再依赖于 m 的值：替换之后，这一陈述的值等值于 false。

上面所发生的是变元俘获(variable capture)现象：在替换中使用了一个包含变

元名的项，而该变元名在被替换的环境中是约束的。在上例中，m 在这一陈述中是自由变元；而用 n 代替 m 意味着使用一个约束变元替换一个自由变元。这显然是我们不希望的。

因此，当把包含变元名的项替换到一个量化表达式中时，如果这个变元名与量化表达式中的约束变元同名，那么就必须为这个约束变元重新命名(rename the bound variable)，这一改变对表达式的真假性不产生影响。为了说明这一点，只要考虑表达式 $\forall x: \mathbb{N} \cdot x=x$ 和 $\forall y: \mathbb{N} \cdot y=y$。在此，使用不同的变元名来表示谓词○"所有的自然数都和它们自身相等"，因而，从逻辑上说，这两个表达式是等值的。

例 7.12 谓词 $\exists n: \mathbb{N} \cdot (n>5 \wedge m<5)$ 等值于

$$\exists l: \mathbb{N} \cdot (l>5 \wedge m<5)$$

因此，在原来的表达式中用 n 替换 m 的效果如下。

$$\exists n: \mathbb{N} \cdot (n>5 \wedge m<5)[n/m]$$
$$\Leftrightarrow \exists l: \mathbb{N} \cdot (l>5 \wedge n<5)$$

当然，这是我们所期望的结果。 □

练习 7.27 下面哪些替换会导致变元俘获？

1. $\forall x: \mathbb{N} \cdot x \geqslant y[z/y]$
2. $\forall x: \mathbb{N} \cdot x \geqslant y[x/y]$
3. $\forall x: \mathbb{N} \cdot x \geqslant y[x+y/y]$
4. $\forall x, y: \mathbb{N} \cdot x \geqslant y[x/y, y/x]$
5. $\forall x: \mathbb{N} \cdot x \geqslant y[z/y][x/z]$
6. $\forall x: \mathbb{N} \cdot x \geqslant y[z/x][x/z]$
7. $(y>0 \wedge \forall x: \mathbb{N} \cdot x \geqslant y)[x+3/y]$ □

练习 7.28 确定上面练习的替换的结果。 □

7.8 限制

在前面的章节中，我们看到了可以通过集合描述定义素数的平方的集合。这可以按如下方式来实现。

$$square\_primes = \{n: \mathbb{N} \mid prime(n) \cdot n^2\}$$

这里，集合描述的谓词部分充当着"过滤器"的角色，它检查自然数，保留满足谓词的自然数，而扔掉不满足谓词的自然数。

过滤器在谓词逻辑中也产生作用。例如，陈述"任何比2大的素数都是奇数"可以表示如下。

$$\forall n: \mathbb{N} \cdot (n \text{ 是素数} \wedge n>2)$$
$$\Rightarrow n \text{ 是奇数}$$

使用过滤器，可以把这一陈述写成如下形式。

$$\forall n: \mathbb{N} \mid n \text{ 是素数} \wedge n>2 \cdot n \text{ 是奇数}$$

在此，这一集合的谓词的范围受到下面的条件的限制(restriction)。

$$n \text{ 是素数} \wedge n>2$$

这种受限谓词的形式与集合描述的形式类似，即受限谓词的形式如下。

quantifier quantification | restriction · predicate

同时，与集合描述的情况一样，缺少限制完全等值于具有限制 true。例如，下面两个陈述是等值的。

$$\forall p: People \mid true \cdot p \text{ 喜欢猪肉罐头}$$
$$\Leftrightarrow \forall p: People \cdot p \text{ 喜欢猪肉罐头}$$

当然，这样的限制也可以用于存在量化变元。例如，陈述"曾经有过女性的英国首相"可以如下表示。

$$\exists p: People \mid p \text{ 是女性}$$
$$\cdot\ p \text{ 曾是英国首相}$$

这种形式的任何陈述都可以改写成不带

○ 原书在说明中并没有严格区别谓词和命题，特别是对如此例的非自明的情况。另外，有时还使用"表达式"和"陈述"。——译者注

有过滤器的形式。在这样做的时候，记住量词间的差异是非常重要的。

再看如下表达式。

$\forall n: \mathbb{N} \mid n$ 是素数 $\wedge n>2 \cdot n$ 是奇数

我们曾认为它等值于下面的表达式。

$\forall n: \mathbb{N} \cdot (n$ 是素数 $\wedge n>2)$

$\Rightarrow n$ 是奇数

也就是说，它表示谓词"给定任意自然数 n，如果 n 是素数且大于 2，那么它是奇数"。在这里，形如

universal quantification | restriction · predicate

的表达式被转化为如下形式的表达式。

universal quantification · restriction ⇒predicate

现在，如果把这一变换运用于下面的表达式。

$\exists p: People \mid p$ 是女性

$\cdot p$ 曾是英国首相

那么，可以得到下面的表达式。

$\exists p: People \cdot p$ 是女性

$\Rightarrow p$ 曾是英国首相

第一个陈述刚好对一个人为真。然而，第二个经过转换的陈述在对一位女性为真的同时，对每个男性都为真(回顾 false⇒true 逻辑等值于 true)[1]。因此，这两个陈述不等值。问题在于，对于存在量词量化的陈述的限制与对于全称量词量化的陈述的限制是不同的：对于后者，限制充当着蕴含的角色，而对于前者，限制充当着合取的角色。因此，限制的正确消去给出如下结果。

$\exists p: People \cdot p$ 是女性

$\wedge p$ 曾是英国首相

显然，这一陈述逻辑等值于原来的陈述。

例 7.13 陈述

$\exists n: \mathbb{N} \mid odd(n) \cdot n=5$

等值于陈述

$\exists n: \mathbb{N} \cdot odd(n) \wedge n=5$ □

例 7.14 陈述

$\forall n: \mathbb{N} \mid odd(n) \cdot n=5$

等值于陈述

$\forall n: \mathbb{N} \cdot odd(n) \Rightarrow n=5$ □

这些等值关系通常表述如下。

规则 7.6

$(\forall x: X \mid p \cdot t) \Leftrightarrow (\forall x: X \cdot p \Rightarrow t)$ □

规则 7.7

$(\exists x: X \mid p \cdot t) \Leftrightarrow (\exists x: X \cdot p \wedge t)$ □

练习 7.29 用下面形式的陈述表示下列各陈述。可以假定有类型 *People* 以及谓词"喜欢猪肉罐头"、"喜欢吉士"和"喜欢香蕉"。

quantifier quantification | restriction · predicate

1. "每个喜欢吉士的人都喜欢猪肉罐头。"

2. "有一个喜欢吉士的人喜欢猪肉罐头。"

3. "每个喜欢吉士的人也喜欢猪肉罐头且喜欢香蕉。"

4. "每个喜欢吉士和猪肉罐头的人也喜欢香蕉。"

5. "有一个喜欢吉士和猪肉罐头的人喜欢香蕉。"

6. "有一个喜欢吉士的人喜欢猪肉罐头和香蕉。" □

练习 7.30 用下面的形式改写练习 7.29 的答案的谓词。

quantifier quantification · predicate □

练习 7.31 确定下列陈述的真假性。

1. $\exists n: \mathbb{N} \mid p(n) \cdot \text{true} \Rightarrow \neg p(n)$

2. $\forall n: \mathbb{N} \mid p(n) \cdot \text{false} \Rightarrow \neg p(n)$

3. $\exists n: \mathbb{N} \mid n \in \varnothing \cdot n$ 是素数

4. $\forall n: \mathbb{N} \mid n \in \varnothing \cdot n$ 是素数 □

---

[1] 此例的说明不是很严密，也不十分清晰。可以考虑表达式：$\exists n: \mathbb{N} \mid n=0$，$n$ 是素数。这一命题显然是假的，但经过转换的陈述 $\exists n: \mathbb{N} \cdot n=0 \Rightarrow n$ 是素数则是真的，$n=1$ 满足谓词。——译者注

7.9 唯一存在量词

存在量词∃允许我们表示形如"至少有一个 x，使得……"之类的陈述。有时我们也许希望考虑形如"恰好存在一个 x，使得……"这样的更加特殊的陈述。

例如，陈述 $\exists x$：$\mathbb{N}$ • $x+1=1$ 等值于 true。另外，谓词 $x+1=1$ 恰好对自然数 0 为真。我们希望存在一个运算符，使得可以表示"恰好存在一个自然数 x，使得 $x+1=1$"或"恰好存在一个超级强权"这样的陈述。幸运的是，我们可以定义这样一个运算符：$\exists_1$。

使用这一新运算符量化的表达式的结构与其他量词量化的表达式结构完全相同。例如，通过 $\exists_1$ 运算符，可以将陈述"恰好存在一个自然数 x，使得 $x+1=1$"表示如下。

$\exists_1 x$：$\mathbb{N}$ • $x+1=1$

如果没有这一运算符，那么通过谓词逻辑只能如下表示这一陈述。

$\exists x$：$\mathbb{N}$ •（$x+1=1 \wedge$

$\forall y$：$\mathbb{N}$ | $x \neq y$ • $y+1 \neq 1$）

这一陈述指出，存在一个使这一性质成立的自然数 x，而且对于所有满足 $y \neq x$ 的自然数 y，这一性质不成立。当然，前面的陈述恰好就是原来的表达式所刻画的，只是更加简洁。

例 7.15 谓词"恰好存在一个超级强权"可以如下表示。

$\exists_1 c$：*Country* • c 是超级强权 □

进一步说，如果我们没有运算符 $\exists_1$，那么只能用更复杂的方式来表示上面这个谓词，即

$\exists c$：*Country* •（c 是超级强权 $\wedge$ $\forall d$：*Country* | $c \neq d$ • $\neg$ d 是超级强权）

练习 7.32 下列陈述哪些为真，哪些为假？

1. $\exists_1 n$：$\mathbb{N}$ • n 是 7 的因子 $\wedge n \neq 1$
2. $\exists_1 n$：$\mathbb{N}$ • $\forall m$：$\mathbb{N}$ • $n \leqslant m$
3. $\exists_1 n$：$\mathbb{N}$ • $\forall m$：$\mathbb{N}$ • $n \geqslant m$
4. $\forall n$：$\mathbb{N}$ • $\exists_1 m$：$\mathbb{N}$ • $n \geqslant m$ □

练习 7.33 使用∃和∀重写练习 7.32 的表达式。 □

陈述 $\exists_1 x$：$\mathbb{N}$ • $x+1=1$ 表明恰好存在一个小于 1 的自然数：当然，我们知道它的值为 0。另外，陈述 $\exists_1 c$：*Country* • c 是超级强权，表明恰好存在一个超级强权：我们可以准确地确定相应类型的哪一个元素是超级强权。

假设陈述 $\exists_1 x$：X • p 等值于 true，有时确切地指出 X 的哪一个元素满足谓词 p 是很有用的。另外，如果可以对这个值实施某个运算会给我们带来方便。运算符 μ 使得我们可以做到这一点：陈述（μx：X | p）读作"使得 p 成立的集合 X 的唯一元素 x"。例如，下面的 μ 表达式给出值 0。

（μx：$\mathbb{N}$ | $x+1=1$）

而下面的 μ 表达式给出值 USA。

（μc：*Country* | c 是超级强权）

当相关类型的一个唯一元素的确拥有相关性质时，我们就可以如上所示地构造陈述。另一方面，不与唯一的元素相关联的 μ 表达式生成一个未定义值。例如，下面两个 μ 表达式都是未定义的。

（μn：$\mathbb{N}$ | n 是偶数）

（μc：*Country* | *population*（c）> 10 000 000）

正如可以把项结合到集合描述中一样，也可以把项运用于 μ 表达式。下面这种形式的表达式将项 t 运用于满足谓词 p 的集合 X 的唯一元素，并返回运用的结果。

（μx：X | p • t）

例如，下面的 μ 表达式给出值 1[①]。

（μx：$\mathbb{N}$ | $x+1=1$ • $x+1$）

而下面的 μ 表达式给出值"华盛顿"。

① 原书中的表达式为（μn：$\mathbb{N}$ | $x+1=1$ • $x+1$），是错误的。——译者注

$(\mu c: Country \mid c$ 是超级强权 $\cdot capital(c))$

与前面一样，如果不存在满足相关性质的唯一元素，那么结果是未定义的。因此，相面两个 μ 表达式未定义。

$(\mu n: \mathbb{N} \mid n$ 是偶数 $\cdot n^2)$

$(\mu c: Country \mid population(c) > 10\ 000\ 000 \cdot capital(c))$

另外，认识到结果项也有相应的类型很重要。因此，命题

$(\mu c: Country \mid c$ 是超级强权 $\cdot capital(c)) = $ 巴黎

等值于 false，而命题

$(\mu c: Country \mid c$ 是超级强权 $\cdot capital(c)) = 1$

未定义：等式左边的类型是城市，而右边的类型是 $\mathbb{N}$。

练习 7.34 假设有相应的类型、谓词和函数，用 μ 表达式写出下列描述。

1. 世界上最高的山。
2. 世界上最高的山的高度。
3. 世界上最老的人。
4. 世界上最老的人的国籍。
5. 最小的自然数。 □

练习 7.35 对下列陈述，说明它们是真的、假的还是未定义的。

1. $(\mu n: \mathbb{N} \mid \forall m: \mathbb{N} \cdot n \leqslant m) = 0$
2. $(\mu n: \mathbb{N} \mid \forall m: \mathbb{N} \cdot n \geqslant m) \neq 0$
3. $(\mu n: \mathbb{N} \mid n^2 = 1) \neq 0$
4. $(\mu n: \mathbb{N} \mid n^2 = n) = 0$
5. $(\mu n: \mathbb{N} \mid n^2 = n \wedge n \neq 1) = 0$ □

7.10 等值推理

在第 3 章中，我们看到了一系列有关命题有效性的推理方法。本节再次讨论其中的两种方法：等值推理和自然演绎。

我们已经见到过许多谓词逻辑的规则。例如，规则 7.2 和规则 7.3 指出全称量词传递合取，存在量词传递析取。另外，规则 7.5 描述了量化表达式上的否定的效果。还有，我们还看到了形如 $\forall x: X \mid p \cdot t$ 和 $\exists x: X \mid p \cdot t$ 的表达式是如何分别与形如 $\forall x: X \cdot p \Rightarrow t$ 和 $\exists x: X \cdot p \wedge t$ 的表达式等值的。此外，下面给出若干直截了当的规则。

首先，如果一个谓词对一个集合的所有元素都成立，那么，它对集合中的某个元素也成立。

规则 7.8

$\forall x: X \cdot p \Rightarrow \exists x: X \cdot p$ □

其次，如果一个谓词对一个集合中的恰好一个元素成立，那么它对这个集合中的至少一个元素成立。

规则 7.9

$\exists_1 x: X \cdot p \Rightarrow \exists x: X \cdot p$ □

如果 p 对类型 X 的所有元素都成立，而且 t 是类型 X 的项，那么 p 对 t 成立。

规则 7.10

$(\forall x: X \cdot p(x) \wedge t \in X) \Rightarrow p(t)$ □

例 7.16 下面的谓词对所有自然数都成立。

$prime(x) \wedge x > 2 \Rightarrow x$ 是奇数

项 $3+4$ 的类型是 $\mathbb{N}$，因此，有

$prime(3+4) \wedge 3+4 > 2 \Rightarrow 3+4$ 是奇数

成立。 □

下面的规则说明：如果 t 是类型 X 的项且 p 对于 t 成立，那么，可以说 p 对 X 的某些元素成立。

规则 7.11

$(t \in X \wedge p(t)) \Rightarrow (\exists x: X \cdot p(x))$ □

例 7.17 命题 $7 \in \mathbb{N}$ 为真，另外，7 是素数，因此，下面的表达式成立。

$\exists n: \mathbb{N} \cdot n$ 是素数 □

对于总是真或总是假的命题，任何变元的量化都没有任何效果。下面的规则刻画了这一事实[1]。

[1] 再次注意，在本书中类型 X 不能是空集。——译者注

规则 7.12

$(\forall x: X \cdot \text{true}) \Leftrightarrow \text{true}$ □

规则 7.13

$(\forall x: X \cdot \text{false}) \Leftrightarrow \text{false}$ □

规则 7.14

$(\exists x: X \cdot \text{true}) \Leftrightarrow \text{true}$ □

规则 7.15

$(\exists x: X \cdot \text{false}) \Leftrightarrow \text{false}$ □

这些规则连同第3章的规则，使我们可以把等值推理运用于谓词逻辑陈述。与第3章的情况一样，上述的等值关系使我们能够确定特定表达式的真假性，或者验证给定逻辑陈述是否是定理。

例 7.18 可以证明下面的表达式是谓词逻辑的定理。

$(\neg \forall x: X \cdot p \Rightarrow q) \Leftrightarrow (\exists x: X \cdot p \wedge \neg q)$

证明如下。

$\neg \forall x: X \cdot p \Rightarrow q$

$\Leftrightarrow \exists x: X \cdot \neg(p \Rightarrow q)$ [规则 7.5]

$\Leftrightarrow \exists x: X \cdot \neg(\neg p \vee q)$ [规则 3.18]

$\Leftrightarrow \exists x: X \cdot p \wedge \neg q$ [规则 3.9]

□

例 7.19 可以如下计算 $\neg(\forall x: X \mid \text{false} \cdot p)$ 的真值。

$\neg(\forall x: X \mid \text{false} \cdot p)$

$\Leftrightarrow \neg(\forall x: X \cdot \text{false} \Rightarrow p)$ [规则 7.6]

$\Leftrightarrow \exists x: X \cdot \neg(\text{false} \Rightarrow p)$ [规则 7.5]

$\Leftrightarrow \exists x: X \cdot \neg(\text{true} \vee p)$ [规则 3.18]

$\Leftrightarrow \exists x: X \cdot \text{false} \wedge \neg p$ [规则 3.9]

$\Leftrightarrow \exists x: X \cdot \text{false}$ [规则 3.5]

$\Leftrightarrow \text{false}$ [规则 7.15]

□

练习 7.36 运用等值推理，证明下列表达式是谓词逻辑的定理。

1. $((\exists x: X \cdot p) \Rightarrow (\exists x: X \cdot q)) \Rightarrow (\exists x: X \cdot p \Rightarrow q)$

2. $(\exists x: X \cdot p \Rightarrow q) \Leftrightarrow ((\forall x: X \cdot p) \Rightarrow (\exists x: X \cdot q))$ □

谓词 $\exists x: X \cdot p(x) \wedge y = z$ 涉及3个变元 x、y 和 z。其中，x 是约束的，y 和 z 是自由的。因为 x 不在谓词 $y = z$ 中出现，可以自由地假设 x 的存在量化对这一谓词的真假性没有影响。因此，可以得出下面的等值成立的结论。

$\exists x: X \cdot (p(x) \wedge y = z) \Leftrightarrow$

$(\exists x: X \cdot p(x)) \wedge y = z$

这一原理构成下列规则的基础。

规则 7.16 设 x 在 q 中没有自由出现，那么下面的等值成立。

$(\forall x: X \cdot p \wedge q) \Leftrightarrow ((\forall x: X \cdot p) \wedge q)$

□

例 7.20 下面的等值成立。

$(\forall n: \mathbb{N} \cdot n > 3 \wedge l = 10) \Leftrightarrow ((\forall n: \mathbb{N} \cdot n > 3) \wedge l = 10)$ □

规则 7.17 设 x 在 q 中没有自由出现，那么下面的等值成立。

$(\forall x: X \cdot p \vee q) \Leftrightarrow ((\forall x: X \cdot p) \vee q)$

□

例 7.21 下面的等值成立。

$(\forall n: \mathbb{N} \cdot n > 3 \vee l = 10) \Leftrightarrow ((\forall n: \mathbb{N} \cdot n > 3) \vee l = 10)$ □

规则 7.18 设 x 在 q 中没有自由出现，那么下面的等值成立。

$(\exists x: X \cdot p \wedge q) \Leftrightarrow ((\exists x: X \cdot p) \wedge q)$

□

例 7.22 下面的等值成立。

$(\exists p: Person \cdot p \text{ 喜欢猪肉罐头} \wedge l = 10)$

$\Leftrightarrow ((\exists p: Person \cdot p \text{ 喜欢猪肉罐头}) \wedge l = 10)$

□

规则 7.19 设 x 在 q 中没有自由出现，那么下面的等值成立。

$(\exists x: X \cdot p \vee q) \Leftrightarrow ((\exists x: X \cdot p) \vee q)$

□

例 7.23 下面的等值成立。

$(\exists p: Person \cdot p \text{ 喜欢猪肉罐头} \vee l = 10)$

$\Leftrightarrow ((\exists p: Person \cdot p \text{ 喜欢猪肉罐头}) \vee l = 10)$

□

练习 7.37 通过等值推理，证明下列值是重言式。可以假设 x 不在 q 中自由出现。

1. $(\forall x: X \cdot p \Rightarrow q) \Leftrightarrow ((\exists x: X \cdot p) \Rightarrow q)$
2. $(\exists x: X \cdot p \Rightarrow q) \Leftrightarrow ((\forall x: X \cdot p) \Rightarrow q)$ □

7.11 自然演绎

就像等值推理一样，也可以把自然演绎拓广到谓词逻辑中。另外，正如每个命题逻辑运算符都有导出和消去规则一样，谓词逻辑运算符也有导出和消去规则。首先，我们考虑有关全称量词的自然演绎规则。

$$\frac{\forall x: X \cdot p \quad t \in X}{p[t/x]}$$ [全称量词消去规则]

这一规则与合取消去规则类似。本质上，这一规则说明，如果 p 对 X 中的所有值都成立，那么，对于类型 X 的项 t，p 也成立。

例 7.24 所有的自然数都大于或等于 0。6+3 是自然数，所以可以得出 6+3 大于 0 的结论。

$$\frac{\forall n: \mathbb{N} \cdot n \geqslant 0 \quad 6+3 \in \mathbb{N}}{6+3 \geqslant 0}$$ [全称量词消去规则] □

对于全称量词表达式的更一般的形式，需要确保项满足谓词。因此，这一证明规则的另一形式如下。

$$\frac{\forall x: X \mid p \cdot q \quad p[t/x] \quad t \in X}{q[t/x]}$$ [全称量词消去规则]

例 7.25 所有大于 2 的素数都是奇数。7 是素数且大于 2，所以可以得出 7 是奇数的结论。

$$\frac{\forall n: \mathbb{N} \mid prime(n) \wedge n > 2 \cdot odd(n) \quad prime(7) \wedge 7 > 2 \quad 7 \in \mathbb{N}}{odd(7)}$$ [全称量词消去规则] □

现在，我们来考虑全称量词引入规则。

如果我们能够证明谓词 p 对 x 的任意值都成立，这里的 x 的任意值指的是在 x 的取值上没有特殊的假设或条件，那么就可以得出谓词 p 对相应类型的每一个值都成立的结论。

$$\begin{array}{c} [x \in X]_1 \\ \vdots \\ \dfrac{p}{\forall x: X \cdot p} \end{array}$$ [全称量词引入规则$_1$（假定 x 在 p 的假设中没有自由出现）]

例 7.26

$$\frac{\dfrac{[n \in \mathbb{N}]_1}{n \geqslant 0}\text{[数学定理]}}{\forall n: \mathbb{N} \cdot n \geqslant 0}$$ [全称量词引入规则$_1$] □

对于形如 $\forall x: X \mid p \cdot q$ 这样的受限谓词，变元 x 也必须满足限制 p。

$$\frac{\begin{array}{c}\lceil x\in X\rceil_1 \quad \lceil p\rceil_1\\ \vdots \\ q\end{array}}{\forall x:\ X \mid p\cdot q}\text{[全称量词引入规则}_1\text{(假定 } x \text{ 在 } q \text{ 的假设中没有自由出现)]}$$

例 7.27

$$\frac{\dfrac{\lceil n\in\mathbb{N}\rceil_1 \quad \lceil prime(n)\wedge n>2\rceil_1}{odd(n)}\text{[数学定理]}}{\forall n:\ \mathbb{N} \mid prime(n)\wedge n>2\cdot odd(n)}\text{[全称量词引入规则}_1\text{]}$$ □

练习 7.38 使用自然演绎规则证明下式。

$(\forall x:\ X\cdot p\wedge \forall x:\ X\cdot q)\Rightarrow(\forall x:\ X\cdot p\wedge q)$ □

练习 7.39 使用自然演绎规则证明下式。

$(\forall x:\ X\cdot p\wedge q)\Rightarrow(\forall x:\ X\cdot p\wedge \forall x:\ X\cdot q)$ □

在考虑了全称量词的引入和消去规则之后，下面考虑存在量词的相应规则。

存在量词消去规则与析取消去规则相似。如果已经假设存在使 p 成立的某个 $x\in X$，那么，如果我们能够得到一个新的谓词 q，而且 x 在 q 中没有自由出现，那么就可以得出结论 q。

$$\frac{\exists x:\ X\cdot p \quad \begin{array}{c}\lceil x\in X\wedge p\rceil_1\\ \vdots \\ q\end{array}}{q}\left[\text{存在量词消去规则}_1\left(\begin{array}{l}x \text{ 在假设中没有自由出现，}\\ \text{且 } x \text{ 在 } q \text{ 中没有自由出现}\end{array}\right)\right]$$

在带有限制的谓词中，限制包含于假设之中；这一规则如下所示。

$$\frac{\exists x:\ X \mid p\cdot q \quad \begin{array}{c}\lceil x\in X\wedge p\wedge q\rceil_1\\ \vdots \\ r\end{array}}{r}\left[\text{存在量词消去规则}_1\left(\begin{array}{l}x \text{ 在假设中没有自由出现，}\\ \text{且 } x \text{ 在 } r \text{ 中没有自由出现}\end{array}\right)\right]$$

下面讨论存在量词引入规则。这一规则的基础是：如果对集合 X 的任意项 t，p 为真，那么就可以得出 $\exists x:\ X\cdot p$ 成立的结论。

$$\frac{t\in X \quad p[t/x]}{\exists x:\ X\cdot p}\text{[存在量词引入规则]}$$

例 7.28 7 是自然数且是素数。因此，$\exists n:\ \mathbb{N}\cdot prime(n)$成立。

$$\frac{7\in\mathbb{N} \quad prime(7)}{\exists n:\ \mathbb{N}\cdot prime(n)}\text{[存在量词引入规则]}$$ □

当然，如果我们要处理存在量词表达式的限制形式，那么 t 必须满足限制。

$$\frac{t\in X \quad p[t/x] \quad q[t/x]}{\exists x:\ X \mid p\cdot q}\text{[存在量词引入规则]}$$

例 7.29 7 是自然数且是素数，7 大于 2，而且 7 是奇数。因此，下面的谓词成立。

$\exists n: \mathbb{N} \mid prime(n) \wedge n>2 \cdot odd(n)$

其证明如下所示。

$$\frac{7\in\mathbb{N} \quad prime(7)\wedge 7>2 \quad odd(7)}{\exists n: \mathbb{N} \mid prime(n)\wedge n>2\cdot odd(n)}[\text{存在量词引入规则}]$$

□

例 7.30 可以证明下式是谓词逻辑的定理。

$\exists x: X \cdot p \wedge q \Rightarrow (\exists x: X \cdot p \wedge \exists x: X \cdot q)$

其证明如下所示。

$$\dfrac{\dfrac{\dfrac{[\exists x: X\cdot p\wedge q]_1 \quad \dfrac{\dfrac{\dfrac{[x\in X\wedge p\wedge q]_2}{p\wedge q}[\wedge\text{elim2}]^{①}}{p}[\wedge\text{elim1}]^{②}}{\exists x: X\cdot p}[\exists\text{intro}]^{③}}{\exists x: X\cdot p}[\exists\text{elim}_2]^{④} \quad \dfrac{\dfrac{\dfrac{[x\in X\wedge p\wedge q]_3}{p\wedge q}[\wedge\text{elim2}]^{①}}{q}[\wedge\text{elim2}]^{①}}{\exists x: X\cdot q}[\exists\text{intro}]^{③} \quad [\exists x: X\cdot p\wedge q]_1}{\exists x: X\cdot q}[\exists\text{elim}_3]^{⑤}}{(\exists x: X\cdot p)\wedge(\exists x: X\cdot q)}[\wedge\text{intro}]^{⑥}}{\exists x: X\cdot p\wedge q\Rightarrow((\exists x: X\cdot p)\wedge(\exists x: X\cdot q))}[\Rightarrow\text{elim}_1]^{⑦}$$

注：① 合取消去规则 2；② 合取消去规则 1；③ 存在量词引入规则；④ 存在量词消去规则$_2$；⑤ 存在量词消去规则$_3$；⑥ 合取引入规则；⑦ 蕴含消去规则$_1$。

□

练习 7.40 使用自然演绎规则证明下式。

$((\exists x: X \cdot p) \vee (\exists x: X \cdot q)) \Rightarrow (\exists x: X \cdot p \vee q)$ □

练习 7.41 使用自然演绎规则证明下式。

$(\exists x: X \cdot \exists y: Y \cdot p) \Rightarrow (\exists y: Y \cdot \exists x: X \cdot p)$ □

7.12 one-point 规则

考虑下面的谓词。

$\exists p$: Person $\cdot$ p 喜欢爱米丽 $\wedge$ p=里克

这一谓词不仅表明有一个喜欢爱米丽的人 p，而且还给出了那个人的名字：里克。因此，可以得出里克 $\in$ *Person* 且里克喜欢爱米丽同时为真的结论。

这就是 one-point 规则[○]。如果我们已经证明 $\exists x: X \cdot p$ 成立，而且在 p 中，x 的值被确定为等于某个项 t，那么就可以得出 $p[t/x]$ 和 $t\in X$ 同时成立的结论。当然，这一规则具有同时消去存在量词和约束变元的作用。

规则 7.20 假设 x 在 t 中没有自由出现，那么下式成立。

$(\exists x: X \cdot p \wedge x=t) \Leftrightarrow (t\in X \wedge p[t/x])$

□

例 7.31

$\exists n: \mathbb{N} \cdot n=n^2 \wedge n=0 \Leftrightarrow 0^2=0 \wedge 0\in\mathbb{N}$

□

在上面的例子中，确定 n 的值的等式是 $n=0$，而不是 $n=n^2$。如果没有约束变元在被替换的项中没有自由出现这一要求的话，就会导致下面的结果。

$\exists n: \mathbb{N} \cdot n=n^2 \wedge n=0 \Leftrightarrow n^2=0 \wedge n^2\in\mathbb{N}$

当然，这一结果没能从新的表达式中移

○ 这一规则没有有助于理解的恰当中文译名。one-point 带有一次完成两项工作的含义。另外，规则的说明与规则本身存在差异。——译者注

去原来的约束变元名。因此，没有达到运用 one-point 规则的目的。

练习 7.42 下列哪些陈述可以运用 one-point 规则？

1. $\exists n: \mathbb{N} \cdot n \geqslant 3 \wedge n = 5$
2. $\exists n: \mathbb{N} \cdot n \geqslant 3 \wedge n = m$
3. $\exists n: \mathbb{N} \cdot n \geqslant 3 \wedge n = m + n$
4. $\exists n: \mathbb{N} \cdot n \geqslant 3 \vee n = 5$ □

练习 7.43 对下列陈述运用 one-point 规则。

1. $\exists p: Person \cdot p$ 拥有红色汽车 $\wedge\ p =$ 贝基
2. $\exists p: Person \cdot p$ 拥有红色汽车 $\wedge\ p$ 喜欢里克 $\wedge\ p =$ 贝基
3. $\exists p: Person \cdot p$ 拥有红色汽车 $\wedge\ p =$ 贝基 $\wedge\ \exists q: Person \cdot \neg q$ 拥有红色汽车 $\wedge\ p$ 喜欢 q
4. $\exists p: Person \cdot p$ 拥有红色汽车 $\wedge\ p =$ 贝基 $\wedge\ \exists q: Person \cdot \neg q$ 拥有红色汽车 $\wedge\ p$ 喜欢 $q \wedge q =$ 里克 □

one-point 规则的自然演绎形式如下所示。

$$\frac{\exists x: X \cdot p \wedge x = t}{p[t/x] \wedge t \in X}\text{[one-point 规则]}$$

例 7.32

$$\frac{\exists p: Person \cdot p \text{ 拥有红色汽车} \wedge p = \text{贝基}}{\text{贝基拥有红色汽车} \wedge \text{贝基} \in Person}\text{[one-point 规则]}$$ □

注意，由于 one-point 规则是一个等值式，所以下面也是有效的推理规则。

$$\frac{p[t/x] \wedge t \in X}{\exists x: X \cdot p \wedge x = t}\text{[one-point 规则]}$$

例 7.33

$$\frac{\text{贝基拥有红色汽车} \wedge \text{贝基} \in Person}{\exists p: Person \cdot p \text{ 拥有红色汽车} \wedge p = \text{贝基}}\text{[one-point 规则]}$$ □

7.13 附加练习

练习 7.44 考虑集合 *People*，满足 {琼，史蒂夫，阿里} $\subseteq$ *People*，以及谓词 *happy*、*in_pain* 和 *angry*，所有这些谓词都与集合 *People* 相关。

使用谓词逻辑表示下列描述。

1. “阿里高兴。”
2. “琼高兴当且仅当阿里生气且史蒂夫痛苦。”
3. “如果阿里高兴那么琼就不高兴。”
4. “如果有人不高兴或有人痛苦，那么琼高兴。”
5. “如果没有人痛苦，那么琼不高兴。”
6. “如果每个人都高兴，那么琼就痛苦。”

练习 7.45 符号化下列关于狗的谓词。

1. “某些狗曾受过良好的训练。”
2. “对于任意一条狗，如果它整洁且曾受过良好训练，那么它有吸引力。”
3. “某些狗温顺且曾受过良好的训练。”
4. “所有温顺的狗都曾受过良好的训练。”
5. “某些狗温顺只有当它曾接受每一个训狗员的训练。”

练习 7.46 假设有谓词 *plane*、*train*、*boat*，使得 *plane*(c, d)表示可以乘飞机从国家 c 前往国家 d，*train*(c, d)表示可以坐火车从国家 c 前往国家 d，*boat*(c, d)表示可以乘船从国家 c 前往国家 d。

使用谓词逻辑表示下列描述。

1. 可以乘飞机从法国飞往新加坡。
2. 可以乘火车和乘飞机从法国前往德国。
3. 可以乘飞机、乘火车及乘船从法国前往英国。
4. 至少有一个国家可以从英国出发乘

火车到达。

5. 任何从法国出发乘飞机可以到达的国家都可以从英国出发乘飞机到达。

练习 7.47 假设有练习 7.46 中的谓词，给出下列谓词逻辑陈述的等价自然语言描述。

1. $\exists c$: $Country \cdot train$(德国, c) $\wedge \neg$ $train$(爱尔兰, c)

2. $\neg \exists c$: $Country \cdot boat$(瑞士, c)

3. $\forall c, d$: $Country \cdot plane(c, d) \Rightarrow plane(d, c)$

4. $\forall c, d$: $Country \cdot plane(c, d) \Rightarrow boat(c, d)$

5. $\forall c$: $Country \cdot \neg plane(c, c)$

练习 7.48

1. 给出一个谓词 p，x 在 p 中出现，使得 $\forall x$: $\mathbb{N} \cdot p$ 及 $\exists x$: $\mathbb{N} \cdot p$ 同时等值于 false。

2. 给出一个谓词 p，x 在 p 中出现，使得 $\forall x$: $\mathbb{N} \cdot p$ 及 $\exists x$: $\mathbb{N} \cdot p$ 同时等值于 true。

3. 给出一个谓词 p，x 在 p 中出现，使得 $\forall x$: $\mathbb{N} \cdot p$ 等值于 false，且 $\exists x$: $\mathbb{N} \cdot p$ 等值于 true。

练习 7.49 假设 $p = x - y > y$，给出下列陈述的真假性。

1. $\forall x$: $\mathbb{N} \cdot \exists y$: $\mathbb{N} \cdot p$

2. $\exists x$: $\mathbb{N} \cdot \forall y$: $\mathbb{N} \cdot p$

3. $\forall y$: $\mathbb{N} \cdot \exists x$: $\mathbb{N} \cdot p$

4. $\exists y$: $\mathbb{N} \cdot \forall x$: $\mathbb{N} \cdot p$

练习 7.50 假设有如下集合，使得 $Capital = Country \times City$。

$Capital = \{$(法国, 巴黎), (USA, 华盛顿)$\}$

假设 $c \in Country$ 及 $d \in City$，下列谓词哪些是有效的，哪些是可满足的，哪些是不可满足的？

1. $(c, d) \in Capital$

2. (USA, d) $\in Capital$

3. (USA, d) $\in Capital \wedge d =$ 巴黎

4. (USA, d) $\in Capital \wedge d =$ 华盛顿

5. (USA, d) $\in Capital \vee$ ($france$, d) $\in Capital$

练习 7.51 对于下列表达式，说明哪些变元是自由的。

1. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$

2. $(\forall x$: $X \cdot x = y) \wedge (\exists y$: $X \cdot y = z)$

3. $\forall x$: $X \cdot \exists y$: $X \cdot (x = y \wedge y = z)$

练习 7.52 给出下列替换的结果。

1. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[w/x]$

2. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[w/y]$

3. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[w/z]$

4. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[x/z]$

5. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[y/z][w/y]$

6. $\forall x$: $X \cdot (x = y \wedge \exists y$: $X \cdot y = z)$ $[y/z, w/y]$

练习 7.53 确定下列命题哪些为真，哪些为假，哪些未定义？

1. $(\mu n$: $\mathbb{N} \mid n - 1 < n) \geqslant 4$

2. $(\mu n$: $\mathbb{N} \mid n - 1 = 3) > 4$

3. $(\mu n$: $\mathbb{N} \mid n - 1 = 3 \cdot n^2) > 4$

7.14 练习解答

7.1

1. 所有的狗都嗅觉好。

2. 所有的狗都或者受过良好的训练或者是小狗。

3. 或者所有的狗都受过良好的训练或者所有的狗都是小狗。

4. 对于任意的狗，如果它是小狗，那么它一定受过良好的训练。

5. 如果所有的狗都是小狗，那么所有的狗都受过良好的训练。

7.2 不是。假设 *Dog* 只由两个元素 fido 和 lassie 组成，而且 fido 不是小狗并且受过良好的训练，而 lassie 是小狗而且没有受过良好的训练。在这种情况下，陈述 4 等值于下面的表达式。

(fido 是小狗⇒fido 受过良好的训练) ∧
(lassie 是小狗⇒lassie 受过良好的训练)

这一表达式等值于 false。另一方面，第二个陈述等值于下面的表达式。

(fido 是小狗 ∧ lassie 是小狗)
⇒(fido 受过良好的训练
∧ lassie 受过良好的训练)

这一表达式等值于 true。

7.3

1. $\forall x$: *People* • x 喜欢佳发蛋糕
2. $\forall x$: *People* • (x 是素食者⇒¬(x 喜欢佳发蛋糕))
3. $\forall x$: *People* • (x 喜欢佳发蛋糕 ∨ x 是素食者)
4. ($\forall x$: *People* • x 喜欢佳发蛋糕)
   ∨
   ($\forall x$: *People* • x 是素食者)
5. $\forall x$: *People* • (x 喜欢佳发蛋糕 ∧ x 是素食者)
6. ($\forall x$: *People* • x 喜欢佳发蛋糕)
   ∧
   ($\forall x$: *People* • x 是素食者)

7.4 不是。考虑只由两个元素吉姆和琼组成的集合 *People*。假设吉姆是素食者且不喜欢佳发蛋糕，而琼不是素食者且喜欢佳发蛋糕。在这里，陈述 3 等值于 *true*，而陈述 4 等值于 *false*。

7.5 是的。∀ 是合取的广义形式，因此，前者对于后者满足分配律。

7.6 1，2 是假的，3，4 是真的。

7.7

1. "某些狗嗅觉好。"
2. "某些狗或者嗅觉好，或者是小狗。"
3. "对于某些狗，如果它们是小狗，那么它们受过良好的训练。"
4. "如果某些狗是小狗，那么某些狗受过良好的训练。"

7.8

1. $\exists x$: *People* • x 喜欢佳发蛋糕
2. $\exists x$: *People* • (x 是素食者⇒¬(x 喜欢佳发蛋糕))
3. $\exists x$: *People* • (x 喜欢佳发蛋糕 ∨ x 是素食者)
4. ($\exists x$: *People* • x 喜欢佳发蛋糕)
   ∨
   ($\exists x$: *People* • x 是素食者)
5. $\exists x$: *People* • (x 喜欢佳发蛋糕 ∧ x 是素食者)
6. ($\exists x$: *People* • x 喜欢佳发蛋糕)
   ∧
   ($\exists x$: *People* • x 是素食者)

7.9 是的。∃ 是析取的广义形式，因此，前者对于后者满足分配律。

7.10 不是。考虑只由两个元素吉姆和琼组成的集合 *People*。假设吉姆是素食者，但是他不喜欢佳发蛋糕，而琼不是素食者，但是喜欢佳发蛋糕。在这里，陈述 5 等值于 false，而陈述 6 等值于 true。

7.11 它们都为真。

7.12 是的○。如果 p 对集合 X 中的所有元素都成立，那么可以得出它对集合 X 中的某些元素成立的结论。

7.13 不是。我们不能因为 p 对集合 X 中的某些元素成立，就说它对集合 X 中的所有元素都成立。

7.14

1. $p \Leftrightarrow (n=n)$

○ 这里还是要注意，因为 X 是类型，所以 $X \neq \varnothing$(在本书中)。——译者注

2. $p \Leftrightarrow (n>3)$

7.15 4 是有效的，1 和 3 是可满足的，2 和 5 是不可满足的。

7.16

1. “某人不喜欢某人。”
2. “某人不喜欢每个人。”
3. “每个人不喜欢某人。”
4. “每个人不喜欢每个人。”

7.17

1. $\exists p, q: Person \cdot \neg\, p$ 喜欢 q
2. $\exists p: Person \cdot \forall q: Person \cdot \neg\, p$ 喜欢 q
3. $\forall p: Person \cdot \exists q: Person \cdot \neg\, p$ 喜欢 q
4. $\forall p, q: Person \cdot \neg\, p$ 喜欢 q

7.18

1. $\exists x: \mathbb{N} \cdot x \neq x$
2. $\exists x: \mathbb{N} \cdot \forall y: \mathbb{N} \cdot x<y$
3. $\exists x: \mathbb{N} \cdot \exists y: \mathbb{N} \cdot x \leqslant y \wedge x \neq y \wedge x \geqslant y$

7.19

$$(\forall x: X \cdot \overbrace{p \wedge (\exists y: Y \cdot \underbrace{q}_{y\text{的辖域}})}^{x\text{的辖域}} \wedge r)$$

其中，使用约束来区分不同的辖域。

7.20

1. x 的出现是自由的。
2. x 的出现是约束的。
3. x 的两个出现都是约束的。
4. x 的出现是约束的，y 的出现是自由的。
5. x 的出现是约束的，y 的两个出现是自由的。
6. x 的第一个出现和 y 的唯一出现都是自由的，而 x 的第二个出现是约束的。
7. x 和 y 的出现都是约束的。
8. y 的第一个出现是自由的，y 的第二个出现和 x 的唯一出现都是约束的。

7.21 可以对所有不含自由变元的陈述赋真值，即 2、3 和 7。

7.22 1 和 3 是命题，2 是谓词。

7.23

1. 对于这一表达式，x 可以替换。
2. 对于这一表达式，没有可以替换的变元。
3. 对于这一表达式，y 可以替换。
4. 对于这一表达式，y 可以替换，x 不可替换。
5. 对于这一表达式，y 及 x 的第二个出现可以替换，但 x 的第一个出现不可替换。

7.24

1. 奈杰尔高兴
2. 奈杰尔高兴 $\wedge$ y 悲伤
3. 奈杰尔高兴 $\vee$ 奈杰尔悲伤
4. x 高兴 $\vee$ 奈杰尔悲伤
5. $\forall x: People \cdot x$ 高兴
6. $\forall x: People \cdot x$ 高兴 $\vee$ x 悲伤
7. $(\forall x: People \cdot x$ 高兴$)$ $\vee$ 奈杰尔悲伤

7.25

1. $\forall x: \mathbb{N} \cdot x \geqslant 0$
2. $\forall x: \mathbb{N} \cdot x \geqslant 3$
3. $3>0 \wedge \forall x: \mathbb{N} \cdot x \geqslant 3$
4. $3>0 \wedge \forall x: \mathbb{N} \cdot x \geqslant y$

7.26

1. $x^2 = 49[y+3/x][4/y] \Leftrightarrow (y+3)^2$
   $= 49[4/y] \Leftrightarrow (4+3)^2$
   $= 49 \Leftrightarrow \text{true}$
2. $x^2 = 49[y+3/x, 4/y] \Leftrightarrow (y+3)^2$
   $= 49$
3. $x^2 > y[y+3/x][4/y] \Leftrightarrow (y+3)^2$
   $> y[4/y] \Leftrightarrow (4+3)^2$
   $> 4 \Leftrightarrow \text{true}$
4. $x^2 > y[y+3/x, 4/y] \Leftrightarrow (y+3)^2$
   > 4

7.27 2、3、5 和 7 可能引发变元俘获。

7.28

1. $\forall x: \mathbb{N} \cdot x \geqslant z$

2. $\forall z: \mathbb{N} \cdot z \geqslant x$

3. $\forall z: \mathbb{N} \cdot z \geqslant x+y$

4. $\forall x, y: \mathbb{N} \cdot x \geqslant y$

5. $\forall w: \mathbb{N} \cdot w \geqslant x$

6. $\forall x: \mathbb{N} \cdot x \geqslant y$

7. $x+3>0 \wedge \forall z: \mathbb{N} \cdot z \geqslant x+3$

7.29

1. $\forall p: People \mid p$ 喜欢吉士 $\cdot p$ 喜欢猪肉罐头

2. $\exists p: People \mid p$ 喜欢吉士 $\cdot p$ 喜欢猪肉罐头

3. $\forall p: People \mid p$ 喜欢吉士 $\cdot p$ 喜欢猪肉罐头 $\wedge p$ 喜欢香蕉

4. $\forall p: People \mid p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头 $\cdot p$ 喜欢香蕉

5. $\exists p: People \mid p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头 $\cdot p$ 喜欢香蕉

6. $\exists p: People \mid p$ 喜欢吉士 $\cdot p$ 喜欢猪肉罐头 $\wedge p$ 喜欢香蕉

7.30

1. $\forall p: People \cdot p$ 喜欢吉士 $\Rightarrow p$ 喜欢猪肉罐头

2. $\exists p: People \cdot p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头

3. $\forall p: People \cdot p$ 喜欢吉士 $\Rightarrow (p$ 喜欢猪肉罐头 $\wedge p$ 喜欢香蕉$)$

4. $\forall p: People \cdot (p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头$) \Rightarrow p$ 喜欢香蕉

5. $\exists p: People \cdot p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头 $\wedge p$ 喜欢香蕉

6. $\exists p: People \cdot p$ 喜欢吉士 $\wedge p$ 喜欢猪肉罐头 $\wedge p$ 喜欢香蕉

7.31

1. $\exists n: \mathbb{N} \mid p(n) \cdot \text{true} \Rightarrow \neg p(n)$

$\Leftrightarrow \exists n: \mathbb{N} \cdot p(n) \wedge (\text{true} \Rightarrow \neg p(n))$

如果 $p(n)$ 等值于 true，那么第二个合取为 false；如果 $p(n)$ 等值于 false，那么第一个合取为 false。因此，这一陈述等值于 false。

2. $\forall n: \mathbb{N} \mid p(n) \cdot \text{false} \Rightarrow \neg p(n)$

$\Leftrightarrow \forall n: \mathbb{N} \cdot p(n) \Rightarrow (\text{false} \Rightarrow \neg p(n))$

结论总是等值于 true。因此，这一陈述等值于 true。

3. $\exists n: \mathbb{N} \mid n \in \varnothing \cdot n$ 是素数

$\Leftrightarrow \exists n: \mathbb{N} \cdot n \in \varnothing \wedge n$ 是素数

$\Leftrightarrow \exists n: \mathbb{N} \cdot \text{false} \wedge n$ 是素数

$\Leftrightarrow \text{false}$

4. $\forall n: \mathbb{N} \mid n \in \varnothing \cdot n$ 是素数

$\Leftrightarrow \forall n: \mathbb{N} \cdot n \in \varnothing \Rightarrow n$ 是素数

$\Leftrightarrow \forall n: \mathbb{N} \cdot \text{false} \Rightarrow n$ 是素数

$\Leftrightarrow \text{true}$

7.32 1 和 2 为真，3 和 4 为假○。

7.33

1. $\exists n: \mathbb{N} \cdot (n$ 是 7 的因子 $\wedge n \neq 1 \wedge \forall m: \mathbb{N} \mid m \neq n \cdot \neg m$ 是 7 的因子 $\vee m=1)$

2. $\exists n: \mathbb{N} \cdot (\forall m: \mathbb{N} \cdot n \leqslant m \wedge \forall l: \mathbb{N} \mid l \neq n \cdot \exists m: \mathbb{N} \cdot l > m)$

3. $\exists n: \mathbb{N} \cdot (\forall m: \mathbb{N} \cdot n \geqslant m \wedge \forall l: \mathbb{N} \mid l \neq n \cdot \exists m: \mathbb{N} \cdot l < m)$

4. $\forall n: \mathbb{N} \cdot (\exists m: \mathbb{N} \cdot (n \geqslant m \wedge \forall l: \mathbb{N} \mid l \neq m \cdot n < l))$

7.34

1. $(\mu m: Mountain \mid m$ 是世界上最高的$)$

2. $(\mu m: Mountain \mid m$ 是世界上最高的 $\cdot height(m))$

3. $(\mu p: People \mid p$ 是世界上最老的$)$

4. $(\mu p: People \mid p$ 是世界上最老的 $\cdot nationality(p))$

5. $(\mu n: \mathbb{N} \mid (\forall m: \mathbb{N} \cdot n \leqslant m))$

○ 原书的解答"4 为真"是错误的。因为对于 $n=1$，有两个 $m(m=1$ 和 $m=0)$ 使得 $n \geqslant m$。参照练习 7.33 的解答。——译者注

7.35 1、3 和 5 为真，2 和 4 未定义。

7.36

1. $(\exists x: X \cdot p) \Rightarrow (\exists x: X \cdot q)$

$\Leftrightarrow \neg(\exists x: X \cdot p) \vee (\exists x: X \cdot q)$ [规则 3.18]

$\Leftrightarrow (\forall x: X \cdot \neg p) \vee (\exists x: X \cdot q)$ [规则 7.5]

$\Rightarrow (\exists x: X \cdot \neg p) \vee (\exists x: X \cdot q)$ [规则 7.8]

$\Leftrightarrow \exists x: X \cdot \neg p \vee q$ [规则 7.3]

$\Leftrightarrow \exists x: X \cdot p \Rightarrow q$ [规则 3.18]

2. $\exists x: X \cdot p \Rightarrow q$

$\Leftrightarrow \exists x: X \cdot \neg p \vee q$ [规则 3.18]

$\Leftrightarrow (\exists x: X \cdot \neg p) \vee (\exists x: X \cdot q)$ [规则 7.3]

$\Leftrightarrow \neg(\forall x: X \cdot p) \vee (\exists x: X \cdot q)$ [规则 7.5]

$\Leftrightarrow (\forall x: X \cdot p) \Rightarrow (\exists x: X \cdot q)$ [规则 3.18]

7.37

1. $\forall x: X \cdot p \Rightarrow q$

$\Leftrightarrow \forall x: X \cdot \neg p \vee q$ [规则 3.18]

$\Leftrightarrow (\forall x: X \cdot \neg p) \vee q$ [规则 7.17]

$\Leftrightarrow (\neg \exists x: X \cdot p) \vee q$ [规则 7.5]

$\Leftrightarrow (\exists x: X \cdot p) \Rightarrow q$ [规则 3.18]

2. $\exists x: X \cdot p \Rightarrow q$

$\Leftrightarrow \exists x: X \cdot \neg p \vee q$ [规则 3.18]

$\Leftrightarrow (\exists x: X \cdot \neg p) \vee q$ [规则 7.19]

$\Leftrightarrow (\neg \forall x: X \cdot p) \vee q$ [规则 7.5]

$\Leftrightarrow (\forall x: X \cdot p) \Rightarrow q$ [规则 3.18]

7.38

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lceil \forall x: X \cdot p \wedge \forall x: X \cdot q \rceil_1}{\forall x: X \cdot p}[\text{合取消去规则 1}]}{p}[\text{全称量词消去规则}] \quad \dfrac{\dfrac{\lceil \forall x: X \cdot p \wedge \forall x: X \cdot q \rceil_1}{\forall x: X \cdot q}[\text{合取消去规则 2}]}{q}[\text{全称量词消去规则}]}{p \wedge q}[\text{合取引入规则 1}] \quad \lceil x \in X \rceil_2}{\forall x: X \cdot p \wedge q}[\text{全称量词引入规则}_2]}{(\forall x: X \cdot p \wedge \forall x: X \cdot q) \Rightarrow \forall x: X \cdot p \wedge q}[\text{蕴含引入规则}_1]$$

7.39

$$\dfrac{\dfrac{\dfrac{\dfrac{\lceil x \in X \rceil_2 \quad \lceil \forall x: X \cdot p \wedge q \rceil_1}{\dfrac{p \wedge q}{p}[\text{合取消去规则 1}]}[\text{全称量词消去规则}]}{\forall x: X \cdot p}[\text{全称量词引入规则}_2] \quad \dfrac{\dfrac{\dfrac{\lceil x \in X \rceil_3 \quad \lceil \forall x: X \cdot p \wedge q \rceil_1}{p \wedge q}[\text{全称量词消去规则}]}{q}[\text{合取消去规则}_2]}{\forall x: X \cdot q}[\text{全称量词引入规则}_3]}{(\forall x: X \cdot p) \wedge (\forall x: X \cdot q)}[\text{合取引入规则}]}{(\forall x: X \cdot p \wedge q) \Rightarrow (\forall x: X \cdot p \wedge \forall x: X \cdot q)}[\text{蕴含引入规则}_1]$$

7.40

$$\dfrac{\dfrac{\lceil \exists x: X \cdot p \vee \exists x: X \cdot q \rceil_1 \quad \dfrac{\dfrac{\dfrac{\lceil \exists x: X \cdot p \rceil_2}{p}[\text{存在量词消去规则}]}{p \vee q}[\text{析取引入规则 1}]}{\exists x: X \cdot p \vee q}[\text{存在量词引入规则}] \quad \dfrac{\dfrac{\dfrac{\lceil \exists x: X \cdot p \rceil_2}{q}[\text{存在量词消去规则}]}{p \vee q}[\text{析取引入规则 2}]}{\exists x: X \cdot p \vee q}[\text{存在量词引入规则}]}{\exists x: X \cdot p \vee q}[\text{析取消去规则}_2]}{((\exists x: X \cdot p) \vee (\exists x: X \cdot q)) \Rightarrow (\exists x: X \cdot p \vee q)}[\text{蕴含引入规则}_1]$$

7.41 我们以要证明的陈述为树根开始：

$\exists x: X \cdot \exists y: Y \cdot p \Rightarrow \exists y: Y \cdot \exists x: X \cdot p$

在此阶段运用蕴含引入规则，给出下面的结果。

$$\frac{\exists y: Y \cdot \exists x: X \cdot p}{\exists x: X \cdot \exists y: Y \cdot p \Rightarrow \exists y: Y \cdot \exists x: X \cdot p}[\text{蕴含引入规则}_1]$$

因此，只需证明从 $\exists x: X \cdot \exists y: Y \cdot p$ 可得到 $\exists y: Y \cdot \exists x: X \cdot p$ 即可。可以通过存在量词消去规则完成这一证明，因此，下一步证明如下所示。

$$\frac{[\exists x: X \cdot \exists y: Y \cdot p]_1 \quad \exists y: Y \cdot \exists x: X \cdot p}{\exists y: Y \cdot \exists x: X \cdot p}[\text{存在量词消去规则}_2]$$

可以按下面的形式完成证明树。

$$\frac{[\exists y: Y \cdot p]_2 \quad \dfrac{\dfrac{[y \in Y \wedge p]_3}{y \in Y}[\text{合取消去规则 1}] \quad \dfrac{\dfrac{[x \in X \wedge p]_2}{x \in X}[\text{合取消去规则 1}] \quad \dfrac{[x \in X \wedge p]_3}{p}[\text{合取消去规则 2}]}{\exists x: X \cdot p}[\text{存在量词引入规则}]}{\exists y: Y \cdot \exists x: X \cdot p}[\text{存在量词引入规则}]}{\exists y: Y \cdot \exists x: X \cdot p}[\text{存在量词消去规则}_3]$$

因为不存在没有释放的假设，所以可以得出下式是谓词逻辑定理的结论。

$\exists x: X \cdot \exists y: Y \cdot p \Rightarrow \exists y: Y \cdot \exists x: X \cdot p$

7.42 只能将 one-point 规则运用于陈述 1 和 2。在陈述 3 中，项 $m+n$ 中自由出现的 n 原本是约束变元。在陈述 4 中，无法确切地确定 n 的值(不符合规则的格式)。

7.43

1. 贝基 $\in$ *Person* $\wedge$ 贝基拥有红色汽车

2. 贝基 $\in$ *Person* $\wedge$ 贝基拥有红色汽车 $\wedge$ 贝基喜欢里克

3. 贝基 $\in$ *Person* $\wedge$ 贝基拥有红色汽车 $\wedge \exists q$: *Person* $\cdot \neg q$ 拥有红色汽车 $\wedge$ 贝基喜欢 q

4. 贝基 $\in$ *Person* $\wedge$ 贝基拥有红色汽车 $\wedge$ 里克 $\in$ *Person* $\wedge \neg$ 里克拥有红色汽车 $\wedge$ 贝基喜欢里克

7.44

1. *happy*(阿里)

2. *happy*(琼) $\Leftrightarrow$ *angry*(阿里) $\wedge$ *in_pain*(史蒂夫)

3. *happy*(阿里) $\Rightarrow \neg$ *happy*(琼)

4. $((\exists p$: *People* $\cdot \neg$ *happy*$(p)) \vee (\exists q$: *People* $\cdot$ *in_pain*$(q))) \Rightarrow$ *happy*(琼)

5. $(\neg \exists p$: *People* $\cdot$ *in_pain*$(p)) \Rightarrow \neg$ *happy*(琼)

6. $(\forall p$: *People* $\cdot$ *happy*$(p)) \Rightarrow$ *in_pain*(琼)

7.45

1. $\exists d$: *Dog* $\cdot$ *well_trained*(d)

2. $\forall d$: *Dog* $\cdot$ *neat*$(d) \wedge$ *well_trained*$(d) \Rightarrow$ *attractive*(d)

3. $\exists d$: *Dog* $\cdot$ *gentle*$(d) \wedge$ *well_trained*(d)

4. $\forall d$: *Dog* $\cdot$ *gentle*$(d) \Rightarrow$ *well_trained*(d)

5. $\exists d$: *Dog* $\cdot$ *gentle*$(d) \Rightarrow \forall t$: *Trainer* $\cdot$ *groomed*(d, t)

7.46

1. *plane*(法国，新加坡)

2. *train*(法国，德国) $\wedge$ *plane*(法国，德国)

3. *train*(法国，英国) $\wedge$ *plane*(法国，英国) $\wedge$ *boat*(法国，英国)

4. $\exists c$: *Country* $\cdot$ *train*(英国，c)

5. $\forall c$: *Country* $\cdot$ *plane*(英国，c) $\Rightarrow$ *plane*(英国，c)

7.47

1. “至少有一个乘火车能够从德国到达且不能乘火车从爱尔兰到达的国家。”

2. “不存在乘船能从瑞士到达的国家。”

3. “如果乘飞机能从一个国家到达另一个国家，那么也可以乘飞机按相反方向旅行。”

4. “如果乘飞机能从一个国家到达另

一个国家，那么也可以乘船做相同的旅行。”

5. “不能乘飞机从一个国家到达它自己。”

7.48

1. $x \neq x$

2. $x = x$

3. $x = 0$

7.49 1、2 和 4 为假；3 为真。

7.50 1、2 和 4 是可满足的；3 是不可满足的；5 是有效的。

7.51

1. y 的第一个出现是自由的，z 的出现也是自由的。

2. y 的第一个出现是自由的，z 的出现也是自由的。

3. z 的出现是自由的。

7.52

1. $\forall x: X \cdot (x = y \wedge \exists y: X \cdot y = z)$

2. $\forall x: X \cdot (x = w \wedge \exists y: X \cdot y = z)$

3. $\forall x: X \cdot (x = y \wedge \exists y: X \cdot y = w)$

4. $\forall v: X \cdot (v = y \wedge \exists y: X \cdot y = x)$

5. $\forall x: X \cdot (x = w \wedge \exists v: X \cdot v = w)$

6. $\forall x: X \cdot (x = w \wedge \exists v: X \cdot v = y)$

7.53 1 未定义，2 为假，3 为真。

第8章 关 系

在第6章，我们看到了笛卡儿积。给定两个集合 X 和 Y，笛卡儿积 $X\times Y$ 定义为形为 (x, y) 的所有序偶的集合，其中，$x\in X$ 且 $y\in Y$。然后，在第7章，我们学习了谓词逻辑。可以对具有某些值 u 和 v 的谓词 p 进行推理。例如，$p(u, v)$ 表示谓词 p 对序偶 (u, v) 成立。进而，如果 u 是 X 的元素，v 是 Y 的元素，那么 $(u, v)\in X\times Y$，而且所有满足 p 的序偶构成 $X\times Y$ 的子集。本章讨论这样的笛卡儿积的子集，称为关系。对此论题需要进一步了解的读者可以参考[DG00]，这本教材集中讲解集合论、关系(本章内容)和函数(下一章所学内容)。

8.1 二元关系

在第6章，我们学习了如何利用笛卡儿积定义序偶的集合，笛卡儿积 $A\times B$ 是满足 $a\in A$ 且 $b\in B$ 的所有形为 (a, b) 的序偶的集合。我们还看到了，所有标准的集合论运算符均能运用于笛卡儿积。因此，例如有

{(杰克，65)，(吉尔，60)}⊂{杰克，吉尔}×{60，65}

这里，序偶的集合{(杰克，65)，(吉尔，60)}将杰克和吉尔的年龄与他们的名字联系在一起，而笛卡儿积则单纯地生成适当类型的所有可能序偶。集合{(杰克，65)，(吉尔，60)}称为关系(relation)：在杰克与65以及吉尔与60之间存在某种关系。更一般地，笛卡儿积 $A\times B$ 的任意子集 R 是一个关系。

例8.1 考虑集合 $A=\{a, b, c\}$ 和 $B=\{0, 1\}$。笛卡儿积 $A\times B$ 为

$A\times B=\{(a, 0), (a, 1), (b, 0), (b, 1), (c, 0), (c, 1)\}$

这一集合的所有子集(包括空集和该集合本身)都是这一类型的关系。 □

如果序偶 (a, b) 在关系 R 中出现，那么，可以说"关系 R 在 a 和 b 间成立"。这可以用多种方式表示。例如，可以写成 $(a, b)\in R$、$R(a, b)$、$a\mapsto b\in R$，或者，在特定环境下，写成 $a\ R\ b$。当书写成 $a\mapsto b$ 时，将序偶 (a, b) 看成关系中的一个映射(maplet)。

例8.2 可以如下定义自然数集合上的 *less_than* 关系。

$less\_than=\{m, n:\mathbb{N} \mid m<n\cdot(m, n)\}$

因此，可以如此书写：$3\mapsto 5\in less\_than$、$less\_than(3, 5)$ 或者 $(3, 5)\in less\_than$。也可以写成 $3\ less\_than\ 5$。 □

用 $A\leftrightarrow B$ 表示集合 A 和 B 间的所有关系的集合。这里，

$A\leftrightarrow B=\mathbb{P}(A\times B)$

因此有 $less\_than\in\mathbb{N}\leftrightarrow\mathbb{N}$。

例8.3 关系的集合{杰克，吉尔}↔{60，65}如下所示。

{∅，{杰克↦60}，{杰克↦65}，{吉尔↦60}，{吉尔↦65}，

{杰克↦60，杰克↦65}，{杰克↦60，吉尔↦60}，{杰克↦60，吉尔↦65}，

{吉尔↦60，杰克↦65}，{吉尔↦60，吉尔↦65}，{杰克↦65，吉尔↦65}，

{杰克↦60，杰克↦65，吉尔↦60}，{杰克↦60，杰克↦65，吉尔↦65}，

{杰克↦60，吉尔↦60，吉尔↦65}，{杰克↦65，吉尔↦60，吉尔↦65}，

{杰克↦60，杰克↦65，吉尔↦60，吉尔↦65}} □

练习8.1 对于 $A=\{0, 1\}$ 和 $B=\{x, y\}$，给出关系的集合 $A\leftrightarrow B$。 □

练习8.2 回想一下第6章的公理定义

的概念。在那里，如下定义满足 $n<100$ 的自然数 n。

$$\begin{array}{|l} n:\mathbb{N} \\ \hline n<100 \end{array}$$

使用公理定义来定义下面的关系。

1. 自然数上的 *equals* 关系，其中 $(m, n)\in equals$ 当且仅当 $m=n$。

2. 集合 *People* 上的 *looks_like* 关系。这里，$(p, q)\in looks\_like$ 当且仅当人物 p 与人物 q 相像。

3. 集合 *People* 与 *Food* 间的 *likes* 关系。这里，当且仅当人物 p 喜欢食物 f 时，$(p, f)\in likes$ 成立。

4. 关系 *double_of_prime*，其中 $(m, n)\in double\_of\_prime$ 当且仅当 n 为素数且 m 是 n 的两倍。 □

8.2 关系的推理

本质上，关系只是序偶的集合。因此，判定两个集合是否相等的技术也可以用于判定两个关系是否相等。

当两个关系 R 和 S 刚好包含相同元素时称 R 与 S 相等；R 与 S 相等当且仅当对于适当类型的任意序偶 (x, y)，$(x, y)\in R\Leftrightarrow(x, y)\in S$ 成立。本章所给的等值式及前面章节所给的规则都可以用于证明关系相等。

练习 8.3 证明

$$\{m:\mathbb{N}\cdot(m, m)\}=\{m, n:\mathbb{N}\mid m=n\}$$

□

8.3 定义域和值域

给定关系 $R\in A\leftrightarrow B$，集合 A 称为 R 的源头（source），集合 B 称为 R 的目标（target）。因此，$R\in\{杰克, 吉尔\}\leftrightarrow\{60, 65\}$ 的源头和目标分别是 $\{杰克, 吉尔\}$ 和 $\{60, 65\}$，而 $S\in\mathbb{N}\leftrightarrow\mathbb{N}$ 的源头和目标均为 $\mathbb{N}$。这两个术语是由映射的相关概念而来的。例如，若 R 定义为

$$R=\{杰克\mapsto 65, 吉尔\mapsto 60\}$$

那么，源头中的元素出现于映射的起点一侧（pointing from），而目标中的元素出现于映射的终点一侧（pointing to）。

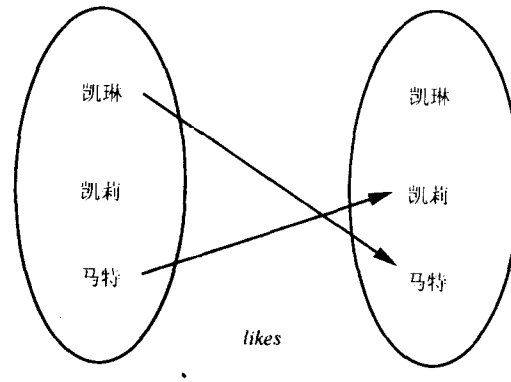
用图形表示关系的一种方法是使用“土豆图”（potato diagram）[○]；这样的关系图特别强调源头与目标的不同。这样的图中包含源头和目标的所有元素，以及标识源头的哪些元素与目标的哪些元素有关联。

例 8.4 设有如下集合和关系：

$House\_mates=\{凯琳, 凯莉, 马特\}$

$likes=\{凯琳\mapsto 马特, 马特\mapsto 凯莉\}$

likes 的土豆图如下所示。



在这里，图的左边是 *likes* 的源头，右边是它的目标。另外，箭头指出关系所包含的映射：凯琳喜欢马特及马特喜欢凯莉。特别要注意的是，两边所列的名字刻画了关系的源头和目标；两个集合的所有元素都必须在图中出现，而不仅是那些映射到的和被映射的元素。 □

关系 R 的定义域（domain）由出现于该关系的源头的元素组成，即由被映射的元素组成。若 $R\in A\leftrightarrow B$，则 R 的定义域 dom R

○ 中文参考书一般称作关系图。——译者注

定义为

$\mathrm{dom}\ R=\{a:A;\ b:B \mid a\mapsto b\in R\cdot a\}$

圆点符号(成分筛选运算符，下同)提供了 dom 的另一种定义，如下所示。

$\mathrm{dom}\ R=\{r:R\cdot r.1\}$

例 8.5　对于关系

$likes=\{$凯琳$\mapsto$马特，马特$\mapsto$凯莉$\}$

有

$\mathrm{dom}\ likes=\{$凯琳，马特$\}$　□

关系 R 的值域(range)由出现于该关系的目标中的元素组成，即由那些映射到的元素组成。若 $R\in A\leftrightarrow B$，则 R 的值域 ran R 定义为

$\mathrm{ran}\ R=\{a:A;\ b:B \mid a\mapsto b\in R\cdot b\}$

同样，圆点符号提供了 ran 的另一种定义，如下所示。

$\mathrm{ran}\ R=\{r:R\cdot r.2\}$

例 8.6　对于关系

$likes=\{$凯琳$\mapsto$马特，马特$\mapsto$凯莉$\}$

有

$\mathrm{ran}\ likes=\{$凯莉，马特$\}$　□

练习 8.4　给出下列关系的定义域和值域。

1. $\{(1,2),(2,3),(3,4)\}$
2. $\{n:\mathbb{N} \mid n\leq 10\cdot(n,2n)\}$
3. $\{n:\mathbb{N}\cdot(n,n)\}$
4. $\{n:\mathbb{N}\cdot(n,n+1)\}$
5. $\{n:\mathbb{N} \mid n\neq n\cdot(n,n)\}$　□

练习 8.5　设 R 和 S 都是 $X\leftrightarrow Y$ 的元素，证明

$\mathrm{dom}(R\cup S)=(\mathrm{dom}\ R)\cup(\mathrm{dom}\ S)$

成立。　□

练习 8.6　设 R 和 S 都是 $X\leftrightarrow Y$ 的元素，证明

$\mathrm{ran}(R\cup S)=(\mathrm{ran}\ R)\cup(\mathrm{ran}\ S)$

成立。　□

练习 8.7　设 R 和 S 都是 $X\leftrightarrow Y$ 的元素，证明

$\mathrm{ran}(R\cap\varnothing)=\mathrm{ran}(S\cap\varnothing)$

成立。　□

8.4　关系的逆

关系 $R\in A\leftrightarrow B$ 的逆关系是 R 的镜像。例如，若 R 是如下关系

$\{(a1,b1),(a2,b2)\}$

则它的逆关系(记作 $R^{\sim}$ 或 R^{-1})是

$\{(b1,a1),(b2,a2)\}$

R 的定义域是它的逆关系的值域，而 R 的值域是它的逆关系的定义域。另外，若 $R\in A\leftrightarrow B$，则 $R^{\sim}\in B\leftrightarrow A$。

可以如下给出关系 R 的逆关系的形式定义。

$R^{\sim}=\{a:A;\ b:B \mid a\mapsto b\in R\cdot b\mapsto a\}$

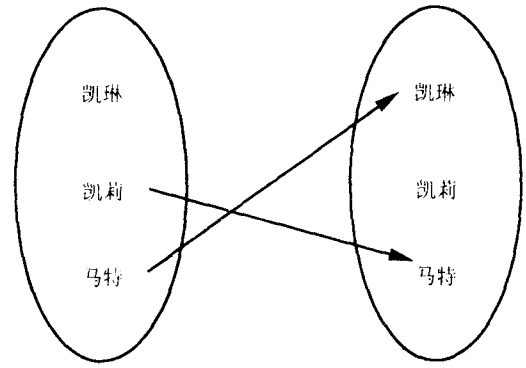
例 8.7　再次考虑如下给出的关系 $likes$。

$likes=\{$凯琳$\mapsto$马特，马特$\mapsto$凯莉$\}$

这一关系的逆关系 $likes^{\sim}$ 如下所示。

$likes^{\sim}=\{$凯莉$\mapsto$马特，马特$\mapsto$凯莉$\}$　□

用土豆图表示，上例的关系如下图所示。



规则 8.1　对于任意的关系 R，有

$\mathrm{dom}\ R^{\sim}=\mathrm{ran}\ R$

$\mathrm{ran}\ R^{\sim}=\mathrm{dom}\ R$　□

例 8.8　考虑以下集合 $People$ 和 $Channels$，以及关系 $tv\in People\leftrightarrow Channels$。

$People=\{$贝基，爱米丽，约翰，迈克尔$\}$

$Channels=\{bbc1, bbc2, itv, c4, c5\}$

$tv=\{$贝基$\mapsto bbc1$，爱米丽$\mapsto itv$，贝基$\mapsto itv$，约翰$\mapsto c5\}$

这时，$tv^{\sim} \in Channels \leftrightarrow People$ 如下所示。

$tv^{\sim}=\{bbc1 \mapsto$贝基，$itv \mapsto$爱米丽，$itv \mapsto$贝基，$c5 \mapsto$约翰$\}$ □

练习 8.8 给出以下各关系的逆关系。

1. $\varnothing$
2. $\{(1, 1), (2, 2)\}$
3. $\{(1, 2), (2, 3), (3, 4)\}$ □

练习 8.9 考虑练习 8.8 中的关系。设其中每个关系都具有类型$\{1, 2, 3, 4\} \times \{1, 2, 3, 4\}$，使用土豆图表示每个关系的逆关系。 □

练习 8.10 设 *Person*、*Mode* 和 *Animal* 为如下所示的集合。

$Person=\{$安迪，戴夫，吉姆，琼，瑞克$\}$

$Mode=\{$骑车，开车，飞行，步行$\}$

$Animal=\{$猫，狗，沙鼠，金鱼，鹦鹉$\}$

考虑如下关系：

$owns=\{$(安迪，狗)，(瑞克，鹦鹉)，(吉姆，金鱼)，(琼，猫)$\}$

$travels=\{$(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)$\}$

计算下列各题。

1. $owns^{\sim}$
2. $travels^{\sim}$
3. $\mathrm{dom}(owns^{\sim})$
4. $\mathrm{ran}(owns^{\sim})$
5. $\mathrm{dom}(travels^{\sim})$
6. $\mathrm{ran}(travels^{\sim})$
7. $\mathrm{ran}(travels^{\sim}) \cap \mathrm{ran}(owns^{\sim})$
8. $\mathrm{ran}(travels^{\sim}) \cup \mathrm{ran}(owns^{\sim})$
9. $\mathrm{ran}(travels^{\sim}) \setminus \mathrm{ran}(owns^{\sim})$ □

练习 8.11 设 R 是 $X \leftrightarrow Y$ 的元素，证明

$$\mathrm{dom}\, R^{\sim}=\mathrm{ran}\, R$$

成立。 □

练习 8.12 设 R 是 $X \leftrightarrow Y$ 的元素，证明

$$R^{\sim\sim}=R$$

成立。 □

8.5 关系上的运算

我们已经看到，给定关系 $R \in X \leftrightarrow Y$，$R$ 的定义域是其源头的子集(这里，源头是集合 X)，R 的值域是其目标的子集(这里，目标是集合 Y)。例如，练习 8.10 中的关系 *travels* 是类型为 $Person \leftrightarrow Mode$ 的关系，其定义如下。

$travels=\{$(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)$\}$

在这里，*travels* 的目标由集合$\{$骑车，开车，飞行，步行$\}$给出，而它的值域则由$\{$骑车，开车，步行$\}$给出。对于此例，值域是目标的真子集。而且，*travels* 的定义域(集合$\{$安迪，戴夫，吉姆，琼$\}$)是它的源头(集合$\{$安迪，戴夫，吉姆，琼，瑞克$\}$)的真子集。

也许(不论是何原因)我们希望只考虑琼和吉姆所用的交通工具，利用定义域限制(domain restriction)运算符可以做到这一点。给定关系 R 及其源头的某个子集 $A \subseteq X$，定义域限制 $A \lhd R$ 表示将 R 定义域限制到 A 上：只考虑由 A 的元素发出的映射，而扔掉所有其他映射。因此，有

$\{$吉姆，琼$\} \lhd travels=\{$(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)$\}$

注意，定义域限制符号指向左边，表示我们限制的是定义域，而不是值域。还要注意，写 $A \lhd R$ 时，A 必须是集合，而且 R 必须是关系。

下面是定义域限制的形式定义(假设 R 是类型为 $X \leftrightarrow Y$ 的关系)。

$A \lhd R = \{x : X;\ y : Y \mid x \in A \wedge x \mapsto y \in R \bullet (x, y)\}$

利用圆点符号的另一种定义如下所示。

$A \lhd R = \{z : R \mid z.1 \in A\}$

练习 8.13　设 *travels* 为如下关系。

travels = {(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

计算下列各题。

1. $\varnothing \lhd travels$
2. $Person \lhd travels$
3. {安迪} $\lhd travels$
4. {安迪，戴夫} $\lhd travels$
5. {安迪，戴夫，瑞克} $\lhd travels$　□

练习 8.14　设 R 是 $X \leftrightarrow Y$ 的元素且 $A \subseteq X$，证明

$\mathrm{dom}(A \lhd R) = A \cap \mathrm{dom}\ R$

成立。　□

定义域伴随限制(domain co-restriction)运算符是定义域限制运算符的补。给定关系 R 及源头的子集 $A \subseteq X$，定义域伴随限制 $A \ntriangleleft R$ 表示 R 的定义域伴随限制到 A：扔掉所有由 A 的元素发出的映射，保留所有其余映射。因此，有

{吉姆，琼} $\ntriangleleft travels$ = {(安迪，步行)，(戴夫，骑车)}

定义域伴随限制符号同样也指向左边，表示我们限制的是定义域。还有，这一符号中有一个减号，表示它是"否定的"伴随限制运算符，而不是"肯定的"限制运算符。

定义域伴随限制的形式定义如下所示(再次假定 $R \in X \leftrightarrow Y$)。

$A \ntriangleleft R = \{x : X;\ y : Y \mid x \notin A \wedge x \mapsto y \in R \bullet (x, y)\}$

使用圆点符号的另一种定义如下所示。

$A \ntriangleleft R = \{z : R \mid z.1 \notin A\}$

练习 8.15　再次设有关系 *travels*，计算下列各题。

1. $\varnothing \ntriangleleft travels$
2. $Person \ntriangleleft travels$
3. {安迪} $\ntriangleleft travels$
4. {安迪，戴夫} $\ntriangleleft travels$
5. {安迪，戴夫，瑞克} $\ntriangleleft travels$　□

练习 8.16　设 R 是 $X \leftrightarrow Y$ 的元素且 $A \subseteq X$，证明

$\mathrm{dom}(A \ntriangleleft R) = \mathrm{dom}\ R \setminus A$

成立。　□

继续考虑关系 *travels*，我们也许只想考虑 *travels* 中步行的人，利用值域限制(range restriction)运算符可以做到这一点。给定关系 R 及目标的子集 $B \subseteq Y$，$R \rhd B$ 表示将 R 的值域限制到 B 上：只考虑映射到 B 的元素的那些映射，扔掉所有其余映射。因此，有

travels $\rhd$ {骑车} = {(吉姆，骑车)，(琼，骑车)，(戴夫，骑车)}

注意，值域限制符号指向右边，表示我们限制的是值域。还要注意的是，写成 $R \rhd B$ 时，R 必须是关系，B 必须是集合。

值域限制的形式定义如下所示(再次假定 R 是类型为 $X \leftrightarrow Y$ 的关系)。

$R \rhd B = \{x : X;\ y : Y \mid y \in B \wedge x \mapsto y \in R \bullet (x, y)\}$

使用圆点符号的另一种定义如下所示。

$R \rhd B = \{z : R \mid z.2 \in B\}$

练习 8.17　计算下列各题。

1. $travels \rhd \varnothing$
2. $travels \rhd Mode$
3. $travels \rhd$ {步行}
4. $travels \rhd$ {步行，骑车}　□

练习 8.18　设 R 是 $X \leftrightarrow Y$ 的元素且 $A \subseteq X$，证明

$(A \lhd R)^{\sim} = (R^{\sim} \rhd A)$

成立。　□

最后，值域伴随限制(range co-restriction)运算符是值域限制运算符的补。给定关系 R 及目标的子集 $B \subseteq Y$，$R \ntriangleright B$ 表示 R 的值域

伴随限制到 B：扔掉那些指向 B 的映射，保留所有其余映射。因此，有

$travels \rhd\!\!- \{骑车\} = \{(安迪，步行)，(吉姆，开车)，(琼，开车)\}$

同样，值域伴随限制运算符指向右边，表示我们限制的是值域。另外，与定义域伴随限制相同，值域伴随限制运算符中也有一个减号，表示它是"否定的"伴随限制，而非"肯定的"限制。

值域伴随限制的形式定义如下所示(再次假定 R 是类型为 $X \leftrightarrow Y$ 的关系)。

$R \rhd\!\!- B = \{x : X;\ y : Y \mid y \notin B \wedge x \mapsto y \in R \bullet (x, y)\}$

使用圆点符号的另一种定义如下所示。

$R \rhd\!\!- B = \{z : R \mid z.2 \notin B\}$

练习 8.19 计算下列各题。

1. $travels \rhd\!\!- \varnothing$
2. $travels \rhd\!\!- Mode$
3. $travels \rhd\!\!- \{步行\}$
4. $travels \rhd\!\!- \{步行，骑车\}$ □

练习 8.20 设 R 是 $X \leftrightarrow Y$ 的元素且 $A \subseteq Y$，证明

$(R \rhd\!\!- A)^{\sim} = (A \lhd\!\!- R^{\sim})$

成立。 □

讨论完如何将关系限制在定义域或值域的某个子集后，自然而然我们会希望考虑特殊关系对定义域的子集的效应(effect)。关系的像(relational image)运算符就是为此目的而定义的。给定关系 R 及定义域的某个子集 A，关系的像 $R(\mid A \mid)$ 返回 ran R 的一个子集，该子集由 A 的元素在 R 下映射到的元素构成。

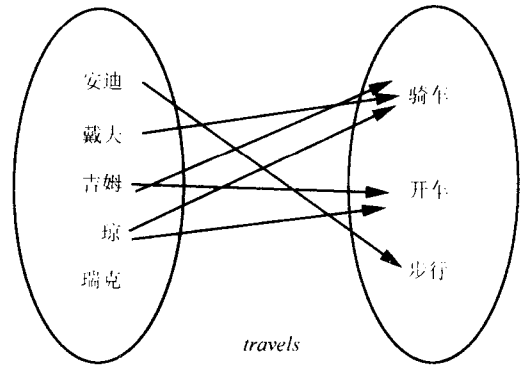
例 8.9 定义域限制 $\{安迪，戴夫\} \lhd$ travels 给出如下结果。

$\{(安迪，步行)，(戴夫，骑车)\}$

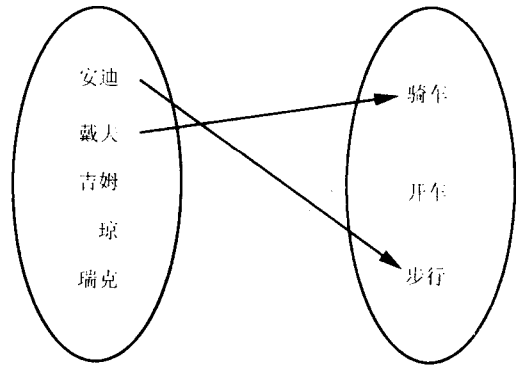
相应地，我们有

$travels(\mid \{安迪，戴夫\} \mid) = \{步行，骑车\}$ □

如果考虑如下 *travels* 的土豆图，关系的像运算符的效果就一目了然了。



如果只考虑从安迪或戴夫出来的箭头，而扔掉所有其余的箭头，则可以得到下面的示意图。



因此，定义域限制返回关系的一个子集(它是一个序偶的集合，其本身也是关系)，而关系的像则返回值域的一个子集(元素的集合)。

例 8.10 定义关系 *showing* 如下。

$showing = \{bbc1 \mapsto 新闻，bbc2 \mapsto 文献片，itv \mapsto 电视知识竞赛，c4 \mapsto 美国喜剧，c5 \mapsto 惊险片\}$

这里，有

$showing(\mid \{bbc1，bbc2\} \mid) = \{新闻，文献片\}$ □

设 $R \in X \leftrightarrow Y$，关系的像的形式定义如下所示。

$R(\mid A \mid) = \{x : X;\ y : Y \mid x \mapsto y \in R \wedge x \in A \bullet y\}$

使用圆点符号可以给出关系的像的另一

种定义，如下所示。

$R(|A|)=\{x:R \mid x.1\in A \cdot x.2\}$

练习 8.21　回忆关系 *travels*，其定义如下所示。

travels＝{（安迪，步行），（吉姆，骑车），（吉姆，开车），（琼，骑车），（琼，开车），（戴夫，骑车）}

计算下列各题。

1. $travels(|\varnothing|)$
2. $travels(|\{瑞克\}|)$
3. $travels(|\{琼\}|)$
4. $travels(|\{琼，吉姆\}|)$
5. $travels(|\{Person\}|)$ □

练习 8.22　设 R 是 $X\leftrightarrow Y$ 的元素且 $A\subseteq X$，证明

$R(|A|)=\mathrm{ran}(A\lhd R)$

成立。 □

8.6　关系的合成

在 8.4 节，我们考虑了逆关系运算符。给定关系 R，R 的逆关系 $R^{\sim}$ 实际上是它的镜像。然后，在 8.5 节，我们考虑了关系上的几个运算：定义域和值域限制及伴随限制，以及关系的像。所有这些运算都与一个关系相关；下面我们考虑的运算则允许把两个关系结合在一起，构成与它们相关的第 3 个关系。

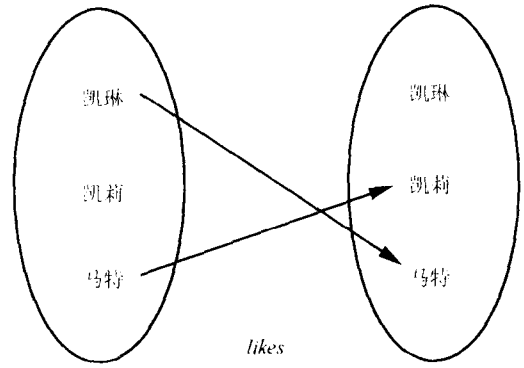
给定两个关系 R 和 S，我们用 $R\fatsemi S$[1] 表示 R 与 S 的合成关系(relational composition)。本质上，这一运算符可以把两个关系"粘合"起来构成一个新关系，新关系刻画了在运用 R 后再运用 S 所带来的效果。

例 8.11　考虑下面的关系。

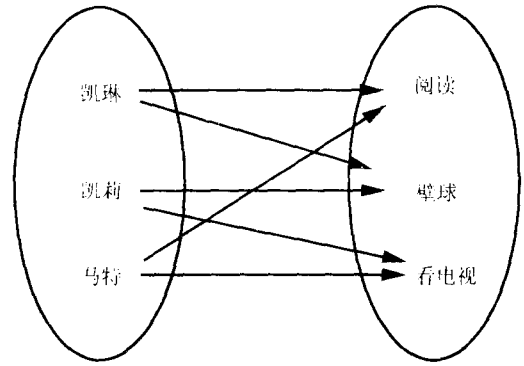
likes＝{凯琳↦马特，马特↦凯莉}

hobbies＝{凯琳↦阅读，凯琳↦壁球，凯莉↦壁球，凯莉↦看电视，马特↦阅读，马特↦看电视}

回想一下，*likes* 的土豆图如下所示。



另外，*hobbies* 的土豆图如下所示。

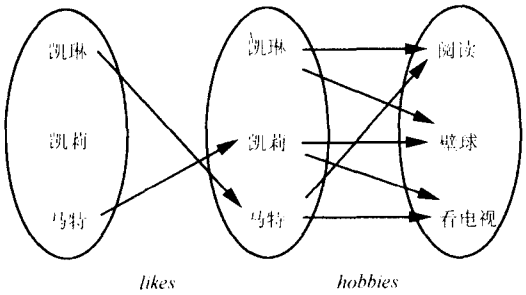


这两个关系的合成关系将这两个图有机地结合在一起：第一个关系的目标与第二个关系的源头结合在一起；当然，为使这样的合成有意义，第一个关系的目标与第二个关系的源头必须是同一类型的。这就如同我们并行绘制两个土豆图，然后把它们拉到一块，使第一个土豆图的目标和第二个土豆图的源头合二为一。

按这种方式，关系 *likes* 和 *hobbies* 的结合如下图所示。

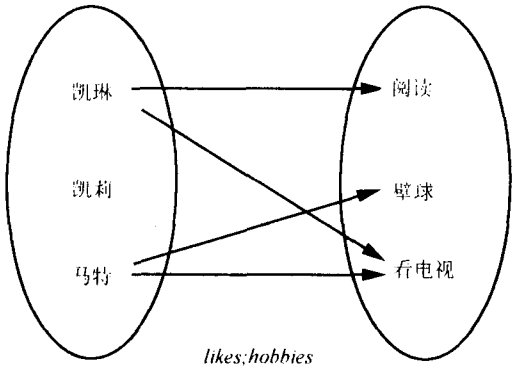
---

[1] 许多离散数学的教科书都将关系的合成 $R\fatsemi S$ 记为 $S\circ R$。这里之所以选择标记 $R\fatsemi S$，是因为它更能反映运用这一运算将两个关系结合在一起的结果：合成的效果等价于作用 R 后再作用关系 S。



其中，可以看出马特喜欢(*likes*)的人的喜好(*hobbies*)包括壁球和看电视：这正是我们期望这样的合成关系提供的信息。另外，凯琳喜欢的人的喜好包括阅读和看电视。

最后，由于每个关系有唯一的源头和唯一的目标，我们要求关系的合成的结果(其本身也是关系)符合这些规则。这样，“中间的土豆”被隐藏了起来。



在这里，因为 $likes \in People \leftrightarrow People$，$hobbies \in People \leftrightarrow Hobby$，可得到合成关系 $likes \fatsemi hobbies$ 的类型为 $People \leftrightarrow Hobby$。这一关系如下所示。

$likes \fatsemi hobbies = \{$凯琳$\mapsto$阅读，凯琳$\mapsto$看电视，马特$\mapsto$壁球，马特$\mapsto$看电视$\}$ □

注意，在上例中，无法定义合成关系 $hobbies \fatsemi likes$，因为 *hobbies* 的目标和 *likes* 的源头不属于同一类型。

形式上，两个关系 $R \in X \leftrightarrow Y$ 和 $S \in Y \leftrightarrow Z$ 的合成关系定义如下。

$R \fatsemi S = \{x : X;\ y : Y;\ z : Z \mid x \mapsto y \in R \wedge y \mapsto z \in S \bullet (x, z)\}$

使用圆点符号的另一种定义如下所示。

$R \fatsemi S = \{x : R;\ y : S \mid x.2 = y.1 \bullet (x.1, y.2)\}$

例 8.12 回忆一下，关系 *tv* 和 *showing* 是如下定义的。

$tv = \{$贝基$\mapsto bbc1$，爱米丽$\mapsto itv$，贝基$\mapsto itv$，约翰$\mapsto c5\}$

$showing = \{bbc1 \mapsto$新闻，$bbc2 \mapsto$文献片，$itv \mapsto$电视知识竞赛，$c4 \mapsto$美国喜剧，$c5 \mapsto$惊险片$\}$

因此，有

$tv \fatsemi showing = \{$贝基$\mapsto$新闻，爱米丽$\mapsto$电视知识竞赛，贝基$\mapsto$电视知识竞赛，约翰$\mapsto$惊险片$\}$ □

练习 8.23 假设有下面的关系，其中 $X \neq Y$。

$A \in X \leftrightarrow X$

$B \in X \leftrightarrow Y$

$C \in Y \leftrightarrow X$

$D \in Y \leftrightarrow Y$

下面的合成关系中哪些是合法的？

1. $A \fatsemi A$
2. $B \fatsemi B$
3. $C \fatsemi D$
4. $D \fatsemi C$
5. $D \fatsemi D$ □

练习 8.24 假设有下列关系。

$R = \{1 \mapsto 1,\ 2 \mapsto 2,\ 3 \mapsto 3\}$

$S = \{1 \mapsto 2,\ 2 \mapsto 3,\ 3 \mapsto 4\}$

$T = \{1 \mapsto 0,\ 2 \mapsto 0,\ 3 \mapsto 0\}$

计算下列各题。

1. $R \fatsemi R$
2. $R \fatsemi S$
3. $R \fatsemi T$
4. $S \fatsemi R$
5. $S \fatsemi S$
6. $S \fatsemi T$ □

练习 8.25　假设有下列关系。

$pets=\{$狗$\mapsto$费杜，狗$\mapsto$赖西，猫$\mapsto$布朗，牛$\mapsto$戴西，金鱼$\mapsto$鱼$\}$

$noises=\{$费杜$\mapsto$咆哮，费杜$\mapsto$怒吼，赖西$\mapsto$咆哮，布朗$\mapsto$喵喵，戴西$\mapsto$哞哞$\}$

计算下列各题。

1. $pets \fatsemi noises$
2. $pets \fatsemi noises(|\{狗\}|)$
3. $\{狗，猫\} \lhd (pets \fatsemi noises)$
4. $(\{狗，猫\} \lhd pets) \fatsemi noises$

练习 8.26　假设 R 是 $X \leftrightarrow Y$ 的元素，S 是 $Y \leftrightarrow Z$ 的元素，证明

$$\mathrm{dom}\ R \fatsemi S \subseteq \mathrm{dom}\ R$$

成立。 □

练习 8.27　假设 R 是 $X \leftrightarrow Y$ 的元素，S 是 $Y \leftrightarrow Z$ 的元素，证明

$$X \lhd (R \fatsemi S) = (X \lhd R) \fatsemi S$$

成立。 □

8.7　同类关系和异类关系

我们已经看到，关系的合成可以把两个关系组合起来定义一个新关系。在某些情况下，我们可能希望反复运用同一个关系。例如，在练习 8.23 中考虑了关系的合成 $A \fatsemi A$ 和 $B \fatsemi B$。如果 $A \fatsemi A$ 本身也是关系，那么当然也可以考虑合成 $A \fatsemi A \fatsemi A$ 及 $A \fatsemi A \fatsemi A \fatsemi A$。

再举一个例子。考虑如下定义的关系 *square*。

$$square=\{0 \mapsto 0,\ 1 \mapsto 1,\ 2 \mapsto 4,\ 3 \mapsto 9,\ \cdots\}$$

可以使用 *square* 定义将自然数与其四次方[①]相关联的关系 *fourth* 如下。

$$fourth = square \fatsemi square$$
$$=\{0 \mapsto 0,\ 1 \mapsto 1,\ 2 \mapsto 16,\ 3 \mapsto 81,\ \cdots\}$$

当然，之所以能够这样合成 *square* 是因为它的源头和目标是同一类型的：这样的关系称为同类关系(homogeneous relation)。

另一方面，下面的关系 *position* 不是同类的：它的源头和目标不是同一类型的。

$$position=\{1 \mapsto a,\ 2 \mapsto b,\ 3 \mapsto c,\ \cdots\}$$

因此，合成 $position \fatsemi position$ 的结果是未定义的。称这些非同类关系为异类关系(heterogeneous relation)。因此，同类关系是可以与其自身合成的关系，而异类关系是不能与其自身合成的关系。

练习 8.28　下面关系中哪些是同类关系，哪些是异类关系？

$A=\{1 \mapsto 2,\ 2 \mapsto 3\}$

$B=\{\{1\} \mapsto 2,\ \{2\} \mapsto 3\}$

$C=\{1 \mapsto \{2\},\ 2 \mapsto \{3\}\}$

$D=\{\{1\} \mapsto \{2\},\ \{2\} \mapsto \{3\}\}$ □

8.8　关系的性质

同类关系中有一些有趣的性质。本节考虑其中的一些性质，即自反性、传递性、对称性、非对称性、反对称性以及完全性。

8.8.1　自反性

类型为 $X \leftrightarrow X$ 的同类关系 R 是自反的(reflexive)，当且仅当在 R 中，X 的每个元素都与其自身有关系。即 R 是自反的，当且仅当

$$\forall x : X \bullet (x,\ x) \in R$$

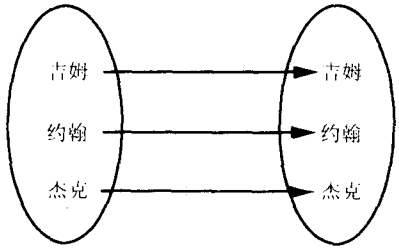
例 8.13　考虑关系 $looks\_like \in People \leftrightarrow People$，其中，

$People=\{$吉姆，约翰，杰克$\}$

关系 *looks_like* 如下图所示。

---

① 原书为 fourth root(四次方根)，与书中的关系定义相违。——译者注



looks_like

其中，源头中的每个元素都与其自身相关联，因此，这一关系是自反的。 □

例 8.14 自然数上的等于关系“=”是自反的。给定任意的自然数 $n \in \mathbb{N}$，总有 $n=n$。 □

例 8.15 自然数上的小于关系“<”不是自反的。给定任意的自然数 $n \in \mathbb{N}$，绝对不会有 $n<n$。 □

例 8.16 关系 *loves* 不是自反的：并非每个人都爱他自己。 □

练习 8.29 下列类型为 $\mathbb{N} \leftrightarrow \mathbb{N}$ 的关系中哪些是自反的？

1. $\varnothing$
2. $\geqslant$
3. $\{1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3\}$
4. $\mathbb{N} \times \mathbb{N}$ □

练习 8.30 下列类型为 $\{a, b, c\} \leftrightarrow \{a, b, c\}$ 的关系中哪些是自反的？

1. $\{a \mapsto a, b \mapsto b\}$
2. $\{a \mapsto a, b \mapsto a, b \mapsto b, c \mapsto c\}$ □

练习 8.31 证明：对任意的自反关系 $R \in X \leftrightarrow X$，有

$$\{x : X \bullet (x, x)\} \subseteq R$$

成立。 □

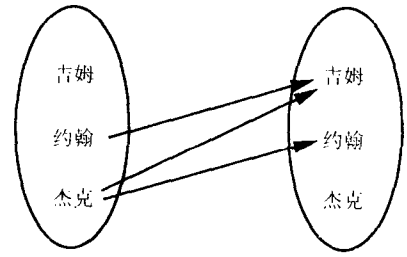
8.8.2 传递性

类型为 $X \leftrightarrow X$ 的同类关系 R 是传递的(transitive)，当且仅当，给定序偶 (x, y) 和 (y, z)，若 (x, y) 和 (y, z) 都是 R 的元素，(x, z) 也必须是 R 的元素。也就是说，R 是传递的，当且仅当，

$$\forall x, y, z : X \bullet (x, y) \in R \wedge (y, z) \in R \Rightarrow (x, z) \in R$$

非形式地说，我们可以认为，当关系 R 满足若 (x, z) 是 $R \fatsemi R$ 的元素则一定是 R 的元素时，它是传递的。

例 8.17 考虑定义在{吉姆，约翰，杰克}上的关系 *likes*。



likes

这里，杰克↦约翰，约翰↦吉姆，杰克↦吉姆是这一关系的元素。由于在 $likes \fatsemi likes$ 中，可以(通过约翰)将杰克映射到吉姆，所以需要杰克↦吉姆是 *likes* 的元素。事实正是如此，而且不存在其他间接的映射，所以这一关系是传递的。 □

例 8.18 考虑下面的关系 *RRN*(reduced rail network 的缩写)。

{伯明翰↦伦敦，伦敦↦伯明翰，伯明翰↦曼彻斯特，曼彻斯特↦伯明翰，伦敦↦曼彻斯特，曼彻斯特↦伦敦，伦敦↦伊普斯威奇，伊普斯威奇↦伦敦}

这里，可以通过伦敦从伯明翰到伊普斯威奇(即伯明翰↦伊普斯威奇是 $RRN \fatsemi RRN$ 的元素)，但无法从伯明翰直接到达伊普斯威奇(即伯明翰↦伊普斯威奇不是 *RRN* 的元素)。因此，这一关系不是传递的。 □

例 8.19 自然数上的等于关系“=”是传递的。给定任意的自然数 $l, m, n \in \mathbb{N}$，若 $l=m$ 且 $m=n$ 则总有 $l=n$。 □

例 8.20 自然数上的小于关系“<”是传递的。给定任意的自然数 $l, m, n \in \mathbb{N}$，若 $l<m$ 且 $m<n$ 则总有 $l<n$。 □

例 8.21 关系 *loves* 不是传递的。例

如，乔爱玛丽而且玛丽爱鲍勃未必能导出乔爱鲍勃。 □

练习 8.32　下列类型为$\mathbb{N}\leftrightarrow\mathbb{N}$的关系中哪些是传递的？

1. $\varnothing$
2. $\geqslant$
3. $\{1\mapsto 1,\ 2\mapsto 2,\ 3\mapsto 3\}$
4. $\mathbb{N}\times\mathbb{N}$ □

练习 8.33　下列类型为$\{a,\ b,\ c\}\leftrightarrow\{a,\ b,\ c\}$的关系中哪些是传递的？

1. $\{a\mapsto a,\ a\mapsto b,\ a\mapsto c,\ b\mapsto c\}$
2. $\{a\mapsto b,\ b\mapsto c\}$ □

练习 8.34　证明：对于任意的传递关系 $R\in X\leftrightarrow X$，$R\fatsemi R\subseteq R$ 成立。 □

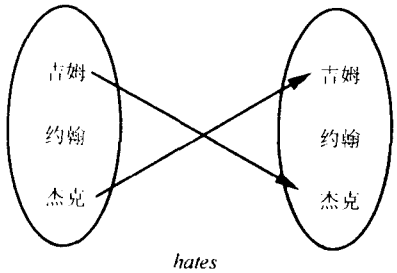
8.8.3　对称性

类型为 $X\leftrightarrow X$ 的同类关系 R 是对称的(symmetric)，当且仅当，对于 R 的每个元素$(x,\ y)$，$(y,\ x)$也是 R 的元素。也就是说，R 是对称的，当且仅当，

$$\forall x,\ y:X\bullet(x,\ y)\in R\Rightarrow(y,\ x)\in R$$

非形式地说，可以认为对称关系是包含自己的"镜像"的关系，即在 R 中，若 x 映射到 y，则 y 也必须映射到 x。

例 8.22　考虑如下定义的关系 *hates*。



hates

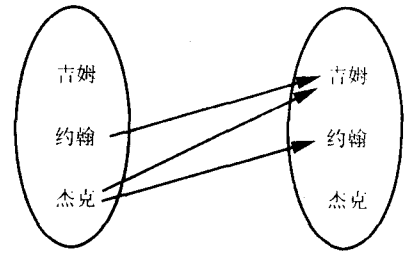
这里，杰克憎恨(*hates*)吉姆而且吉姆憎恨杰克。这一事实，以及 *hates* 不含其他序偶的事实表明这一关系是对称的。 □

例 8.23　再次考虑如下定义的关系 *RRN*。

RRN＝{伯明翰↦伦敦，伦敦↦伯明翰，
伯明翰↦曼彻斯特，曼彻斯特↦
伯明翰，伦敦↦曼彻斯特，
曼彻斯特↦伦敦，伦敦↦
伊普斯威奇，伊普斯威奇↦伦敦}

这一关系是对称的，这是因为，对于任意给定的元素 a 和 b，若在 *RRN* 中从 a 可以到达 b，那么同样在 *RRN* 中从 b 可以到达 a。 □

例 8.24　回顾如下定义的关系 *likes*。



likes

这一关系不是对称的：虽然约翰喜欢吉姆，但吉姆并不喜欢约翰。因此，这不是一个对称关系。 □

例 8.25　自然数上的等于关系"＝"是对称的。给定任意的 $m,\ n\in\mathbb{N}$，总有若 $m=n$ 则 $n=m$。 □

例 8.26　自然数上的小于关系"＜"不是对称的。给定任意的 $m,\ n\in\mathbb{N}$，若 $m<n$ 则绝不会有 $n<m$。 □

例 8.27　关系 *loves* 不是对称的：若乔爱玛丽则玛丽爱乔未必成立。 □

练习 8.35　下列类型为$\mathbb{N}\leftrightarrow\mathbb{N}$的关系中哪些是对称的？

1. $\varnothing$
2. $\geqslant$
3. $\{1\mapsto 1,\ 2\mapsto 2,\ 3\mapsto 3\}$
4. $\mathbb{N}\times\mathbb{N}$ □

练习 8.36　下列类型为$\{a,\ b,\ c\}\leftrightarrow\{a,\ b,\ c\}$的关系中哪些是对称的？

1. $\{a\mapsto a,\ a\mapsto b,\ b\mapsto a\}$
2. $\{a\mapsto a,\ b\mapsto b,\ c\mapsto b\}$ □

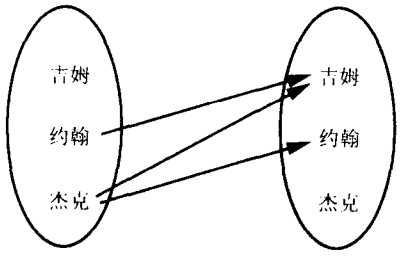
练习 8.37　证明：对于任意的对称关系 $R\in X\leftrightarrow X$，有 $R^{\sim}=R$。 □

8.8.4 非对称性

类型为 $X \leftrightarrow X$ 的同类关系 R 是非对称的(asymmetric)，当且仅当，对于 R 中出现的任意序偶(x, y)，(y, x)不出现于 R 中。也就是说，R 是非对称的，当且仅当，

$$\forall x, y : X \bullet (x, y) \in R \Rightarrow (y, x) \notin R$$

例 8.28 考虑如下关系 *older_than*。



older_than

这里，杰克比约翰和吉姆都大，而约翰比吉姆大。这一关系显然是非对称的。 □

例 8.29 自然数上的等于关系“$=$”不是非对称的。给定任意的自然数 m，$n \in \mathbb{N}$，若 $m=n$ 则 $n \neq m$ 绝不成立。 □

例 8.30 自然数上的小于关系“$<$”是非对称的。给定任意的自然数 m，$n \in \mathbb{N}$，若 $m<n$ 则 $n<m$ 总不成立。 □

例 8.31 关系 *loves* 不是非对称的：若乔爱玛丽则玛丽不爱乔未必成立。 □

练习 8.38 下列类型为$\mathbb{N} \leftrightarrow \mathbb{N}$ 的关系中哪些是非对称的？

1. $\varnothing$
2. $\geqslant$
3. $\{1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3\}$
4. $\mathbb{N} \times \mathbb{N}$ □

练习 8.39 下列类型为$\{a, b, c\} \leftrightarrow \{a, b, c\}$的关系中哪些是非对称的？

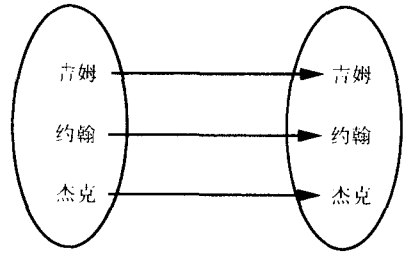
1. $\{a \mapsto a, a \mapsto b, b \mapsto a\}$
2. $\{a \mapsto b, b \mapsto c\}$ □

8.8.5 反对称性

类型为 $X \leftrightarrow X$ 的同类关系 R 是反对称的(antisymmetric)，当且仅当，对于 R 中出现的任意序偶(x, y)和(y, x)，有 $x=y$。也就是说，R 是反对称的，当且仅当，

$$\forall x, y : X \bullet (x, y) \in R \wedge (y, x) \in R \Rightarrow x=y$$

例 8.32 再次考虑如下定义的关系 *looks_like*。



looks_like

这一关系是反对称的，因为源头中的每个元素只映射到它自身。 □

例 8.33 再次考虑如下定义的关系 *RRN*。

RRN={伯明翰↦伦敦，伦敦↦伯明翰，
伯明翰↦曼彻斯特，曼彻斯特↦
伯明翰，伦敦↦曼彻斯特，
曼彻斯特↦伦敦，伦敦↦
伊普斯威奇，伊普斯威奇↦伦敦}

这一关系不是反对称的。例如，既可以从伊普斯威奇到达伦敦，又可以从伦敦到达伊普斯威奇，但显然伦敦和伊普斯威奇是不同的元素。 □

例 8.34 自然数上的等于关系“$=$”是反对称的。给定任意的自然数 m，$n \in \mathbb{N}$，若 $m=n$ 且 $n=m$ 则总有 $m=n$。 □

例 8.35 自然数上的小于关系“$<$”是反对称的。不存在自然数 m，$n \in \mathbb{N}$，使得 $m<n$ 且 $n<m$。因此，蕴含式的前提等值于 false。所以整个蕴含式等值于 true。 □

例 8.36 关系 *loves* 不是反对称的：乔爱玛丽而且玛丽爱乔未必能推出乔和玛丽是同一个人。 □

练习 8.40 下列类型为$\mathbb{N} \leftrightarrow \mathbb{N}$ 的关系

中哪些是反对称的？

1. $\varnothing$

2. $\geqslant$

3. $\{1\mapsto 1,\ 2\mapsto 2,\ 3\mapsto 3\}$

4. $\mathbb{N}\times\mathbb{N}$ □

练习 8.41　下列类型为$\{a,\ b,\ c\}\leftrightarrow\{a,\ b,\ c\}$的关系中哪些是反对称的？

1. $\{a\mapsto a,\ a\mapsto b,\ b\mapsto a\}$

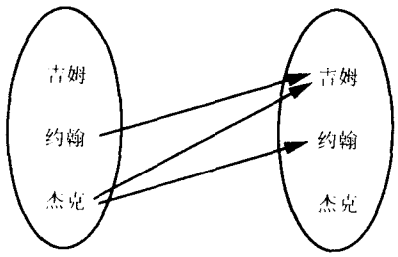
2. $\{a\mapsto b,\ b\mapsto c\}$ □

8.8.6　完全性

类型为 $X\leftrightarrow X$ 的同类关系 R 是完全的(total)，当且仅当，对于任意不同的 x，$y\in X$，有$(x,\ y)\in R$ 或者$(y,\ x)\in R$。也就是说，R 是完全的，当且仅当，

$$\forall x,\ y:X\bullet(x,\ y)\in R\vee(y,\ x)\in R\vee x=y$$

例 8.37　再次考虑如下定义的关系 *older_than*。



older_than

这一关系是完全的，因为给定任意的元素 x 和 y，或者 $x=y$，或者$(x,\ y)\in older_than$，或者$(y,\ x)\in older_than$ 成立。 □

例 8.38　再次考虑如下定义的关系 RRN。

$RRN=\{$伯明翰$\mapsto$伦敦，伦敦$\mapsto$伯明翰，
伯明翰$\mapsto$曼彻斯特，曼彻斯特$\mapsto$
伯明翰，伦敦$\mapsto$曼彻斯特，
曼彻斯特$\mapsto$伦敦，伦敦$\mapsto$
伊普斯威奇，伊普斯威奇$\mapsto$伦敦$\}$

这一关系不是完全的。因为，对于不同的元素 m 和 n，$m\mapsto n\in RRN$ 或 $n\mapsto m\in RRN$ 不总成立。 □

例 8.39　自然数上的等于关系“$=$”不是完全的。给定任意的自然数 m，$n\in\mathbb{N}$，$m=n$ 不总成立。 □

例 8.40　自然数上的小于关系“$<$”是完全的。对于任意的自然数 m，$n\in\mathbb{N}$，$m<n$或 $n<m$ 或 $m=n$ 总成立。 □

例 8.41　关系 *loves* 不是完全的：给定两个不同的人 p 和 q，p 爱 q 或 q 爱 p 不总成立。 □

练习 8.42　下列类型为$\mathbb{N}\leftrightarrow\mathbb{N}$ 的关系中哪些是完全的？

1. $\varnothing$

2. $\geqslant$

3. $\{1\mapsto 1,\ 2\mapsto 2,\ 3\mapsto 3\}$

4. $\mathbb{N}\times\mathbb{N}$ □

练习 8.43　下列类型为$\{a,\ b,\ c\}\leftrightarrow\{a,\ b,\ c\}$的关系中哪些是完全的？

1. $\{a\mapsto b,\ a\mapsto c,\ b\mapsto a\}$

2. $\{a\mapsto b,\ b\mapsto c,\ a\mapsto c,\ b\mapsto b\}$ □

8.9　顺序与等价

性质的不同组合引发不同的关系类。本节讨论三种最常见的关系：偏序、全序和等价关系。

8.9.1　偏序

给定同类关系 R，若 R 是自反、传递且反对称的，则称 R 为*偏序*(partial ordering)。如果集合 X 有偏序，那么这一集合的元素形成某种“层次”(或顺序)：某些元素高于其他元素。这样的顺序之所以是部分的(partial)，是因为并非所有元素的序偶都需要以此关系相关联。

例 8.42　考虑自然数上的关系“$\leqslant$”。这里，“$\leqslant$”是自反的，因为对于每个 $n\in\mathbb{N}$，有 $n\leqslant n$。进而，$\leqslant$是传递的，因为对于任意的 l、m，$n\in\mathbb{N}$，若 $l\leqslant m$ 且 $m\leqslant n$ 则有 $l\leqslant n$。

最后，$\leqslant$是反对称的，因为对于任意的 m，$n \in \mathbb{N}$，若 $m \leqslant n$ 且 $n \leqslant m$ 则一定有 $m=n$。因此，可以得出$\leqslant$是偏序的结论。 □

练习 8.44 自然数上的关系"$<$"不是偏序。为什么不是？ □

练习 8.45 证明：给定集合 X，$\subseteq$在 X 的子集上构成偏序。 □

8.9.2 全序

给定同类关系 R，若 R 是反对称、传递且完全的，则称 R 为全序(total ordering)。之所以这样称呼，是因为所有元素的序偶都以此关系相关联：给定全序集合 X 的任意两个不同的元素 x 和 y，在这样的关系中，x 总是或者高于 y 或者低于 y。即全序集合中的任意两个不同的元素总可以比较，这在偏序中是不成立的。

例 8.43 考虑自然数上的关系"$<$"。这里，$<$是反对称的，因为对于任意的 m，$n \in \mathbb{N}$，若 $m<n$ 且 $n<m$ 则 $m=n$(前提从不成立，所以蕴含式显然是可满足的)。进而，$<$是传递的，因为对于任意的 l，m，$n \in \mathbb{N}$，若 $l<m$ 且 $m<n$ 则有 $l<n$。最后，$<$是完全的，因为对于任意的 m，$n \in \mathbb{N}$，或者 $m<n$ 或者 $n<m$ 或者 $m=n$ 成立。 □

练习 8.46 自然数上的关系"$=$"不是全序。为什么不是？ □

练习 8.47 证明：给定集合 X，$\subseteq$在 X 的子集上不构成全序[○]。 □

8.9.3 等价关系

给定同类关系 R，若 R 是对称、传递且自反的，则称 R 为等价关系(equivalence relation)。若 R 是等价关系，那么若$(x, y) \in R$ 则在某种意义下 x"等价于"y。

例 8.44 考虑自然数上的关系"$=$"。这里，$=$是对称的，因为对于任意的 m，$n \in \mathbb{N}$，若 $m=n$ 则 $n=m$。进而，$=$是传递的，因为对于任意的 l，m，$n \in \mathbb{N}$，若 $l=m$ 且 $m=n$ 则有 $l=n$。最后，$=$是自反的，因为对于任意的 $n \in \mathbb{N}$，$n=n$ 成立。 □

练习 8.48 自然数上的关系"$\leqslant$"不是等价关系。为什么不是？ □

练习 8.49 证明：给定集合 X，$\subseteq$在 X 的子集上不构成等价关系[○]。 □

8.10 闭包

给定同类关系 R，若 R 不是自反的，那么，如果愿意，总可以在 R 中加入一些映射(序偶)来扩展 R，使扩展了的关系成为自反关系。在这种情况下，包含 R 且自反的最小关系称为 R 的自反闭包(reflexive closure)。类似地，可以在 R 中添加映射，使得扩展后的关系成为对称关系。在这种情况下，包含 R 且对称的最小关系称为 R 的对称闭包(symmetric closure)。最后，可以对传递性做同样的操作：包含 R 且传递的最小关系称为 R 的传递闭包(transitive closure)。本节学习自反、对称和传递闭包。

8.10.1 自反闭包

为了构造关系 S 的自反闭包，记作 S^R，我们需要寻找包含 S 且自反的最小关系。

给定某个类型 X，考虑恒等关系 Id，Id 是由使得 $x \in X$ 的所有形式为(x, x)的映射组成的集合。

$$Id[X]=\{x : X \cdot (x, x)\}$$

例如，

$$Id[\{a, b, c\}]=\{(a, a), (b, b), (c, c)\}$$

假设 S 是类型为 $Y \leftrightarrow Y$ 的关系，构造 S 的自反闭包的最简单的方法是通过并运算将

○ 这要求 X 既不是空集，也不是单集。——译者注

S 和 $Id[Y]$联合在一起。

$S^R = S \cup Id[Y]$

这显然是包含 S 的最小的自反关系。注意，如果 S 已经是自反的话，那么这一定义保证 $S^R = S$。

例 8.45　考虑集合 $People$={约翰，理查德，邓肯，卡尔斯顿}，以及如下定义的关系 $older\_than$。

$older\_than$={约翰↦理查德，约翰↦邓肯，理查德↦邓肯}

这一关系的自反闭包，即包含 $older\_than$ 的最小自反关系，可由 $older\_than$ 和 $Id[People]$ 的并给出。由定义，关系 $Id[People]$由下式给出。

$Id[People]$={约翰↦约翰，邓肯↦邓肯，理查德↦理查德，卡尔斯顿↦卡尔斯顿}

因此，$older\_than^R$ 由下式给出。

$older\_than^R$={约翰↦理查德，约翰↦邓肯，理查德↦邓肯，约翰↦约翰，理查德↦理查德，邓肯↦邓肯，卡尔斯顿↦卡尔斯顿}

□

练习 8.50　设集合 Pet={猫，狗，老鼠}。给出下列各类型为 $Pet \leftrightarrow Pet$ 的关系的自反闭包。

1. ∅
2. {猫↦狗，狗↦老鼠}
3. {猫↦狗，老鼠↦狗}
4. {猫↦猫，狗↦狗，老鼠↦老鼠}　□

8.10.2　传递闭包

为了构造关系 R 的传递闭包，记作 R^+，我们需要找到包含 R 的最小传递关系。

回顾关系的合成 $R \circ_9 R$。这一合成等效于两次运用 R：序偶$(x, z) \in R \circ_9 R$ 当且仅当存在某个 y，使得$(x, y) \in R$ 且$(y, z) \in R$。关系合成的另一种表示方式是 R^2，它表示关系 R 被运用了两次。做类似的扩展，关系 R^3 表示三次运用 R：序偶$(w, z) \in R \circ_9 R \circ_9 R$ 当且仅当存在元素 x 和 y，使得$(w, x) \in R$，$(x, y) \in R$ 且$(y, z) \in R$。更一般地，R^n 表示 n 次运用 R。这一表示如下所示。

$R^1 = R$

$R^2 = R \circ_9 R$

$R^3 = R \circ_9 R \circ_9 R$

$R^4 = R \circ_9 R \circ_9 R \circ_9 R$

给出上述的表示后，可以如下定义关系 R^+，即关系 R 的传递闭包。

$R^+ = \bigcup\{n : \mathbb{I} \setminus \{0\} \bullet R^n\}$

它表示所有出现于 R、出现于 $R \circ_9 R$、出现于 $R \circ_9 R \circ_9 R$ 等等的序偶所组成的集合。

例 8.46　考虑集合 $People$={约翰，理查德，邓肯，卡尔斯顿}，以及如下定义的关系 $likes$。

$likes$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿}

这里，$likes^2$ 由下式给出。

$likes^2$={约翰↦邓肯，理查德↦卡尔斯顿}

对 $likes^2$ 和 $likes$ 做关系的合成，可以得到 $likes^3$。

$likes^3$={约翰↦卡尔斯顿}

对此再次运用 $likes$ 可得

$likes^4 = \varnothing$

由上可知，对于任意的 $k > 3$，$likes^k$ 都等价于空集。因此，进一步运用 $likes$ 不会在 $likes^+$ 中添加任何新序偶。于是，有

$likes^+$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿，约翰↦邓肯，理查德↦卡尔斯顿，约翰↦卡尔斯顿}

□

练习 8.51　设集合 Pet={猫，狗，老

鼠}，给出下列类型为 $Pet \leftrightarrow Pet$ 的关系的传递闭包。

1. $\varnothing$

2. {猫↦狗，狗↦老鼠}

3. {猫↦狗，老鼠↦狗}

4. {猫↦猫，狗↦狗，老鼠↦老鼠} □

练习 8.52 给出下面关系的传递闭包。

$\{a \mapsto b, b \mapsto c, c \mapsto d, d \mapsto e\}$ □

8.10.3 自反传递闭包

为了构造关系 R 的自反传递闭包，记作 R^*，我们需要找到包含 R 的最小的自反且传递的关系。

设 R 是类型为 $X \leftrightarrow X$ 的关系，可以如下定义 R^*。

$R^* = \bigcup\{n : \mathbb{N} \bullet R^n\}$

回顾如下定义的集合 X 上的关系 $Id[X]$。

$Id[X] = \{X : X \bullet x \mapsto x\}$

由此，可以认为 $R^0 = Id[X]$。

例 8.47 考虑集合 $People$={约翰，理查德，邓肯，卡尔斯顿}，以及如下定义的关系 $likes$。

$likes$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿}

这里，$Id[People]$由下式给出。

$Id[People]$={约翰↦约翰，理查德↦理查德，邓肯↦邓肯，卡尔斯顿↦卡尔斯顿}

进而，$likes^+$ 由下式给出。

$likes^+$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿，约翰↦邓肯，理查德↦卡尔斯顿}

因此，$likes^*$ 由下式给出。

$likes^*$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿，约翰↦邓肯，理查德↦卡尔斯顿，约翰↦卡尔斯顿，约翰↦约翰，理查德↦理查德，邓肯↦邓肯，卡尔斯顿↦卡尔斯顿} □

练习 8.53 设集合 Pet={猫，狗，老鼠}，给出下列类型为 $Pet \leftrightarrow Pet$ 的关系的自反传递闭包。

1. $\varnothing$

2. {猫↦狗，狗↦老鼠}

3. {猫↦狗，老鼠↦狗}

4. {猫↦猫，狗↦狗，老鼠↦老鼠} □

8.10.4 对称闭包

为了构造关系 R 的对称闭包，记作 R^S，我们需要找到包含 R 的最小对称关系。

回忆一下，类型为 $X \leftrightarrow X$ 的关系 R 是对称的，当且仅当，对于任意的 $x, y \in X$，若 $(x, y) \in R$ 则 $(y, x) \in R$。

再回忆一下，给定类型为 $X \leftrightarrow X$ 的任意关系 R，可以如下定义 R 的逆关系 $R^\sim$。

$R^\sim = \{x, y : X \mid (x, y) \in R \bullet (y, x)\}$

因此，包含 R 的最小对称关系由下式给出。

$R^S = R \cup R^\sim$

例 8.48 再一次考虑集合 $People$={约翰，理查德，邓肯，卡尔斯顿}，以及如下定义的关系 $likes$。

$likes$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿}

$likes$ 的逆关系由下式给出。

$likes^\sim$={理查德↦约翰，邓肯↦理查德，卡尔斯顿↦邓肯}

因此，$likes^S$ 由下式给出。

$likes^S$={约翰↦理查德，理查德↦邓肯，邓肯↦卡尔斯顿，理查德↦约翰，邓肯↦理查德，卡尔斯顿↦邓肯} □

练习 8.54 设集合 Pet={猫，狗，老鼠}，给出下列类型为 $Pet \leftrightarrow Pet$ 的关系的对称闭包。

1. $\varnothing$

2. {猫↦狗，狗↦老鼠}

3. {猫↦狗，老鼠↦狗}

4. {猫↦猫，狗↦狗，老鼠↦老鼠}　□

8.11　*n* 元关系

回忆一下，第6章曾介绍过，有时序偶不能自然地表示我们要处理的数据；有时多元组，例如三元组或四元组更合适。例如，笛卡儿积 $Name \times Age \times Email$ 可以表示所有形式为(n, a, e)的多元组所组成的集合，其中，$n \in Name$，$a \in Age$，$e \in Email$。

其后，在本章的开始，我们指出，关系是笛卡儿积的子集。由于关系运算符依赖于二元关系，所以我们对关系的定义做了限制：关系是序偶的集合，而非任意尺度的多元组的集合。

回到名字、年龄和电子邮件地址的例子，可以发现定义类型为 $Name \leftrightarrow Age \leftrightarrow Email$ 的关系是不合法的。我们可以定义类型为 $Name \leftrightarrow (Age \leftrightarrow Email)$ 的关系，但是这一关系的典型元素将是

{(杰克，{(65，jack@the_hill.com)，(60，jill@the_hill.com)})}

这显然不是我们需要的。

另一方面，类型为 $Name \leftrightarrow (Age \times Email)$ 和 $(Name \times Age) \leftrightarrow Email$ 的关系完全是合法的；前者由形式为$(n, (a, e))$的元素组成，后者由形式为$((n, a), e)$的元素组成。两者都符合二元关系的要求，因此可以用它们来表示需要的信息。

练习 8.55　考虑下列集合。

Yes_No={yes，no}

$Name$={戴夫，吉姆，琼}

给出下列关系的类型。

1. {(戴夫，yes)，(吉姆，no)}

2. {((戴夫，yes)，yes)，((吉姆，no)，no)}

3. {((yes，yes)，(戴夫，yes))，((no，yes)，(戴夫，no))}

4. {(yes，(戴夫，吉姆))，(no，(戴夫，琼))}　□

练习 8.56　设关系 R 的定义如下所示。

$R = \{((1, 1), (0, 1)), ((1, 0), (0, 0)), ((0, 1), (1, 0)), ((0, 0), (1, 1))\}$

计算下列各题。

1. dom R

2. ran R

3. $\{(1, 1)\} \lhd R$

4. $R^{\sim}$

5. $R \fatsemi R$

6. $\{r : R \bullet (r.1.1, r.2)\}$　□

8.12　附加练习

练习 8.57　假设有下列关系。

$parent = \{p, q : Person \mid p$ 是 q 的父亲$\}$

$sibling = \{p, q : Person \mid p$ 是 q 的兄弟或姐妹$\}$

$married = \{p, q : Person \mid p$ 与 q 是夫妻$\}$

同时，假定有谓词 $male$ 和 $female$：当且仅当 p 为男性时，$male(p)$为 true，当且仅当 p 为女性时，$female(p)$为 true。

根据以上假设，定义下列关系。

1. $husband$：$(p, q) \in husband$ 表示 p 是 q 的丈夫。

2. $wife$：$(p, q) \in wife$ 表示 p 是 q 的妻子。

3. $mother$：$(p, q) \in mother$ 表示 p 是 q 的母亲。

4. $father$：$(p, q) \in father$ 表示 p 是 q 的父亲。

5. son：$(p, q) \in$ son 表示 p 是 q 的儿子。

6. $daughter$：$(p, q) \in daughter$ 表示 p 是 q 的女儿。

7. *brother*：$(p, q) \in brother$ 表示 p 是 q 的兄弟。

8. *sister*：$(p, q) \in sister$ 表示 p 是 q 的姐妹。

练习 8.58 假设有练习 8.57 的集合和谓词，给出下列关系的定义。

1. *aunt*：$(p, q) \in aunt$ 表示 p 是 q 的姑妈或姨妈。

2. *uncle*：$(p, q) \in uncle$ 表示 p 是 q 的伯父或舅父。

3. *nephew*：$(p, q) \in nephew$ 表示 p 是 q 的侄子或外甥。

4. *niece*：$(p, q) \in niece$ 表示 p 是 q 的侄女或外甥女。

5. *cousin*：$(p, q) \in cousin$ 表示 p 是 q 的堂兄弟、姐妹。

6. *father_in_law*：$(p, q) \in father\_in\_law$ 表示 p 是 q 的岳父或公公。

7. *mother_in_law*：$(p, q) \in mother\_in\_law$ 表示 p 是 q 的岳母或婆婆。

8. *son_in_law*：$(p, q) \in son\_in\_law$ 表示 p 是 q 的女婿。

9. *daughter_in_law*：$(p, q) \in daughter\_in\_law$ 表示 p 是 q 的儿媳妇。

练习 8.59 假设有练习 8.57 的集合和谓词，定义下列关系。

1. *stepfather*：$(p, q) \in stepfather$ 表示 p 是 q 的继父。

2. *stepmother*：$(p, q) \in stepmother$ 表示 p 是 q 的继母。

3. *stepdaughter*：$(p, q) \in stepdaughter$ 表示 p 是 q 的养女。

4. *stepson*：$(p, q) \in stepson$ 表示 p 是 q 的养子。

5. *stepbrother*：$(p, q) \in stepbrother$ 表示 p 是 q 的异父母兄弟。

6. *stepsister*：$(p, q) \in stepsister$ 表示 p 是 q 的异父母姐妹。

7. *grandmother*：$(p, q) \in grandmother$ 表示 p 是 q 的祖母或外祖母。

8. *grandfather*：$(p, q) \in grandfather$ 表示 p 是 q 的祖父或外祖父。

练习 8.60 假设已知前面几个练习的关系。利用上面的关系和关系运算符，定义以下关系或集合。

1. 关系 *child*：$(p, q) \in child$ 当且仅当 p 是 q 的孩子。

2. 给定人物 p 的孙子女的集合。

3. 给定人物 p 的父母的祖父母的集合。

4. 给定人物 p 的祖先的集合。

5. 所有母亲的集合。

6. 有女儿的所有父亲的集合。

7. 有兄弟的所有男性。

8. 孙子女都为孙女的祖母的集合。

9. 所有孩子均为女儿的祖父的集合。

练习 8.61 假设有如下定义的关系 R 和 S。

$R=\{(1, 2), (2, 4), (3, 6), (4, 8)\}$

$S=\{(1, 1), (2, 2), (3, 3), (4, 4)\}$

可以认为这两个关系都是类型为 $\mathbb{N} \leftrightarrow \mathbb{N}$ 的关系。

1. 使用集合描述定义 R 和 S。

2. dom R 是什么？

3. ran S 是什么？

4. $R^{\sim}$ 是什么？

5. $R \fatsemi S$ 是什么？

6. $S \fatsemi R$ 是什么？

练习 8.62 假设有下列关系。

$id=\{1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3, 4 \mapsto 4\}$

$plus\_one=\{1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4, 4 \mapsto 5\}$

$square=\{1 \mapsto 1, 2 \mapsto 4, 3 \mapsto 9, 4 \mapsto 16\}$

$double=\{1 \mapsto 2, 2 \mapsto 4, 3 \mapsto 6, 4 \mapsto 8\}$

假设上述关系都是类型为 $\mathbb{N} \leftrightarrow \mathbb{N}$ 的，计算以下各题。

1. dom *id*

2. ran *square*

3. $double^{\sim}$

4. $double \fatsemi square$

5. $square \fatsemi double$

6. $\{1, 2\} \lhd plus\_one$

7. $\{1, 2\} \lhd plus\_one$

8. $plus\_one \rhd \{1, 2\}$

9. $plus\_one \ntriangleright \{1, 2\}$

10. $square(|\{1, 2\}|)$

11. $double(|\{1, 2\}|)$

12. $double^{-}(|\{1, 2\}|)$

练习 8.63　假设有如下定义的关系 $loves \in Housemates \leftrightarrow Housemates$。

$loves = \{$杰姬$\mapsto$杰姬，布赖恩$\mapsto$杰姬$\}$

进一步假设 $Housemates = \{$布赖恩，杰姬$\}$。

1. $loves$ 是自反的吗？

2. $loves$ 是对称的吗？

3. $loves$ 是传递的吗？

练习 8.64　假设有练习 8.63 的关系 $loves$。

1. 给出 $loves$ 的自反闭包。

2. 给出 $loves$ 的对称闭包。

练习 8.65　考虑下面的关系。

$noises = \{$费杜$\mapsto$咆哮，费杜$\mapsto$怒吼，赖西$\mapsto$咆哮，布朗$\mapsto$喵喵，戴西$\mapsto$哞哞$\}$

给定上面 $noises$ 的定义，计算下列各题。

1. $\{$费杜$\} \lhd noises$

2. $\{$费杜，赖西$\} \lhd noises$

3. $\{$戴西$\} \ntriangleleft noises$

4. $\{$戴西，布朗$\} \ntriangleleft noises$

5. $noises \rhd \{$咆哮$\}$

6. $noises \rhd \{$怒吼，哞哞$\}$

7. $noises \ntriangleright \{$喵喵$\}$

8. $noises \ntriangleright \{$咆哮，喵喵$\}$

9. $noises(|\{$费杜$\}|)$

10. $noises(|\varnothing|)$

练习 8.66　下列关系中哪些是传递的，哪些是对称的？

1. $\varnothing$

2. $\{$(吉姆，乔)，(乔，比尔)，(吉姆，比尔)，(史蒂夫，瑞克)$\}$

3. $\{$(吉姆，乔)，(乔，比尔)，(吉姆，比尔)$\}$

4. $\{$(瑞克，史蒂夫)，(史蒂夫，瑞克)，(吉姆，比尔)，(比尔，吉姆)$\}$

练习 8.67　给出练习 8.66 中各关系的对称闭包。

练习 8.68　给出练习 8.66 中各关系的传递闭包。

练习 8.69　下列陈述中哪些为真？

1. 若 R 和 S 是自反的，则 $R \fatsemi S$ 也是自反的。

2. 若 R 和 S 是传递的，则 $R \fatsemi S$ 也是传递的。

3. 若 R 和 S 是对称的，则 $R \fatsemi S$ 也是对称的。

练习 8.70　解释为什么$\Leftrightarrow$是所有命题的集合上的等价关系。

练习 8.71　可以简化$(R^{+})^{+}$吗？

8.13　练习解答

8.1

$\{\varnothing, \{0 \mapsto x\}, \{0 \mapsto y\}, \{1 \mapsto x\}, \{1 \mapsto y\}, \{0 \mapsto x, 0 \mapsto y\},$

$\{0 \mapsto x, 1 \mapsto x\}, \{0 \mapsto x, 1 \mapsto y\}, \{1 \mapsto x, 0 \mapsto y\}, \{1 \mapsto x, 1 \mapsto y\},$

$\{0 \mapsto y, 1 \mapsto y\}, \{0 \mapsto x, 0 \mapsto y, 1 \mapsto x\}, \{0 \mapsto x, 0 \mapsto y, 1 \mapsto y\},$

$\{0 \mapsto x, 1 \mapsto x, 1 \mapsto y\}, \{0 \mapsto y, 1 \mapsto x, 1 \mapsto y\},$

$\{0 \mapsto x, 0 \mapsto y, 1 \mapsto x, 1 \mapsto y\}\}$

8.2

1.

$$\begin{array}{|l} equals : \mathbb{N} \leftrightarrow \mathbb{N} \\ \hline equals = \{m : \mathbb{N} \bullet (m, m)\} \end{array}$$

2.

$$\begin{array}{|l} looks\_like : People \leftrightarrow People \\ \hline looks\_like = \{p, q : People \mid p \\ \text{looks like } q \bullet (p, q)\} \end{array}$$

3.

$$\begin{array}{|l} likes : People \leftrightarrow Food \\ \hline likes = \{p : People;\ f : Food \mid p\ likes\ f \bullet (p,\ f)\} \end{array}$$

4.

$$\begin{array}{|l} double\_of\_prime : \mathbb{N} \leftrightarrow \mathbb{N} \\ \hline double\_of\_prime = \{n : \mathbb{N} \mid prime(n) \bullet (2 \times n,\ n)\} \end{array}$$

8.3

$(x,\ y) \in \{m : \mathbb{N} \bullet (m,\ m)\} \Leftrightarrow x = y$

$\Leftrightarrow (x,\ y) \in \{m,\ n : \mathbb{N} \mid m = n\}$

8.4

1. $domain = \{1,\ 2,\ 3\}$

   $range = \{2,\ 3,\ 4\}$

2. $domain = \{0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6,\ 7,\ 8,\ 9,\ 10\}$

   $range = \{0,\ 2,\ 4,\ 6,\ 8,\ 10,\ 12,\ 14,\ 16,\ 18,\ 20\}$

3. $domain = \mathbb{N}$

   $range = \mathbb{N}$

4. $domain = \mathbb{N}$

   $range = \mathbb{N} \setminus \{0\}$

5. $domain = \varnothing$

   $range = \varnothing$

8.5

$x \in \mathrm{dom}(R \cup S)$

$\Leftrightarrow \exists y : Y \bullet x \mapsto y \in R \cup S$

$\Leftrightarrow \exists y : Y \bullet x \mapsto y \in R \vee x \mapsto y \in S$

$\Leftrightarrow \exists y : Y \bullet x \mapsto y \in R \vee \exists y : Y \bullet x \mapsto y \in S$

$\Leftrightarrow x \in \mathrm{dom}\ R \vee x \in \mathrm{dom}\ S$

$\Leftrightarrow x \in (\mathrm{dom}\ R) \cup (\mathrm{dom}\ S)$

8.6

$y \in \mathrm{ran}(R \cup S)$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \cup S$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \vee x \mapsto y \in S$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \vee \exists x : X \bullet x \mapsto y \in S$

$\Leftrightarrow y \in \mathrm{ran}\ R \vee y \in \mathrm{ran}\ S$

$\Leftrightarrow y \in (\mathrm{ran}\ R) \cup (\mathrm{ran}\ S)$

8.7

$y \in \mathrm{ran}(R \cap \varnothing)$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \cap \varnothing$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \wedge x \mapsto y \in \varnothing$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in R \wedge false$

$\Leftrightarrow false$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in S \wedge false$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in S \wedge x \mapsto y \in \varnothing$

$\Leftrightarrow \exists x : X \bullet x \mapsto y \in S \cap \varnothing$

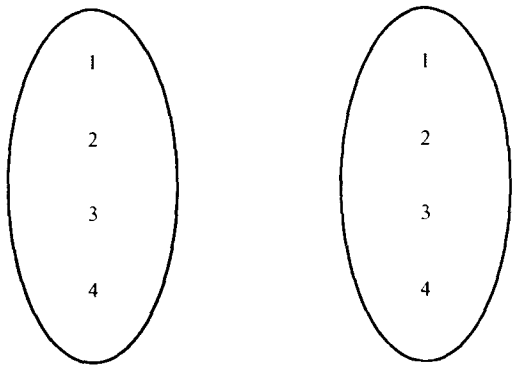
$\Leftrightarrow y \in \mathrm{ran}(S \cap \varnothing)$

8.8

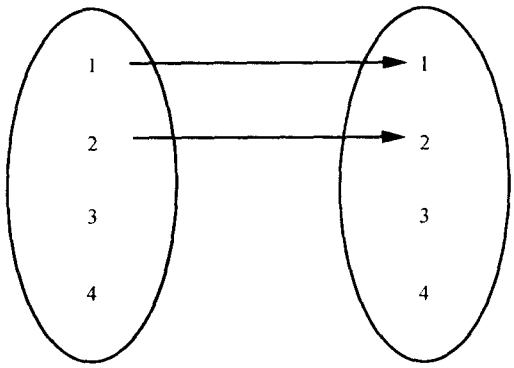
1. $\varnothing$
2. $\{(1,\ 1),\ (2,\ 2)\}$
3. $\{(2,\ 1),\ (3,\ 2),\ (4,\ 3)\}$

8.9

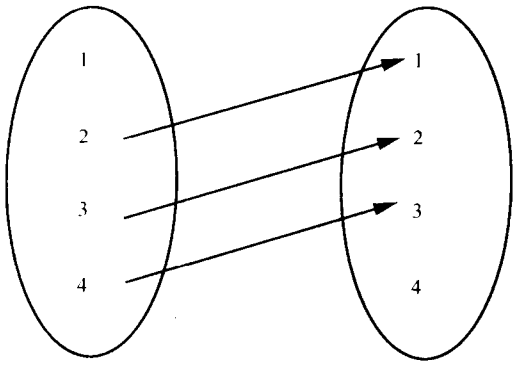
1.



2.



3.



8.10

1.{(狗，安迪)，(鹦鹉，瑞克)，(金鱼，吉姆)，(猫，琼)}

2.{(步行，安迪)，(骑车，吉姆)，(开车，吉姆)，(骑车，琼)，(开车，琼)，(骑车，戴夫)}

3.{狗，鹦鹉，金鱼，猫}

4.{安迪，瑞克，吉姆，琼}

5.{步行，骑车，开车}

6.{安迪，戴夫，吉姆，琼}

7.{安迪，吉姆，琼}

8.{安迪，戴夫，吉姆，琼，瑞克}

9.{戴夫}

8.11

$x \in \mathrm{dom} R^{\sim}$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in R^{\sim}$

$\Leftrightarrow \exists y : Y \cdot y \mapsto x \in R$

$\Leftrightarrow x \in \mathrm{ran}\ R$

8.12

$(x, y) \in R^{\sim\sim}$

$\Leftrightarrow (y, x) \in R^{\sim}$

$\Leftrightarrow (x, y) \in R$

8.13

1. $\varnothing$

2.{(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

3.{(安迪，步行)}

4.{(安迪，步行)，(戴夫，骑车)}

5.{(安迪，步行)，(戴夫，骑车)}

8.14

$x \in \mathrm{dom}(A \lhd R)$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in A \lhd R$

$\Leftrightarrow x \in A \wedge \exists y : Y \cdot x \mapsto y \in R$

$\Leftrightarrow x \in A \wedge x \in \mathrm{dom}\ R$

$\Leftrightarrow x \in A \cap \mathrm{dom}\ R$

8.15

1.{(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

2. $\varnothing$

3.{(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

4.{(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)}

5.{(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)}

8.16

$x \in \mathrm{dom}(A \unlhd R)$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in A \unlhd R$

$\Leftrightarrow \exists y : Y \cdot x \notin A \wedge x \mapsto y \in R$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in R \wedge x \notin A$

$\Leftrightarrow (\exists y : Y \cdot x \mapsto y \in R) \wedge x \notin A$

$\Leftrightarrow x \in \mathrm{dom}\ R \wedge x \notin A$

$\Leftrightarrow x \in \mathrm{dom}\ R \setminus A$

8.17

1. $\varnothing$

2.{(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

3.{(安迪，步行)}

4.{(安迪，步行)，(吉姆，骑车)，(琼，骑车)，(戴夫，骑车)}

8.18

$(x, y) \in (A \lhd R)^{\sim}$

$\Leftrightarrow (y, x) \in A \lhd R$

$\Leftrightarrow y \in A \wedge (y, x) \in R$

$\Leftrightarrow (y, x) \in R \wedge y \in A$

$\Leftrightarrow (x, y) \in R^{\sim} \wedge y \in A$

$\Leftrightarrow (x, y) \in (R^{\sim} \rhd A)$

8.19

1. {(安迪，步行)，(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

2. $\varnothing$

3. {(吉姆，骑车)，(吉姆，开车)，(琼，骑车)，(琼，开车)，(戴夫，骑车)}

4. {(吉姆，开车)，(琼，开车)}

8.20

$(x, y) \in (R \rhd A)^{\sim}$

$\Leftrightarrow (y, x) \in R \rhd A$

$\Leftrightarrow (y, x) \in R \wedge x \notin A$

$\Leftrightarrow (x, y) \in R^{\sim} \wedge x \notin A$

$\Leftrightarrow (x, y) \in (A \lhd R^{\sim})$

8.21

1. $\varnothing$

2. $\varnothing$

3. {开车，骑车}

4. {开车，骑车}

5. {开车，骑车，步行}

8.22

$y \in R(|A|)$

$\Leftrightarrow \exists x : X \cdot x \mapsto y \in R \wedge x \in A$

$\Leftrightarrow \exists x : X \cdot x \mapsto y \in (A \lhd R)$

$\Leftrightarrow y \in \mathrm{ran}(A \lhd R)$

8.23 1、4 及 5 是合法的，2 和 3 是不合法的。

8.24

1. $\{1 \mapsto 1, 2 \mapsto 2, 3 \mapsto 3\}$

2. $\{1 \mapsto 2, 2 \mapsto 3, 3 \mapsto 4\}$

3. $\{1 \mapsto 0, 2 \mapsto 0, 3 \mapsto 0\}$

4. $\{1 \mapsto 2, 2 \mapsto 3\}$

5. $\{1 \mapsto 3, 2 \mapsto 4\}$

6. $\{1 \mapsto 0, 2 \mapsto 0\}$

8.25

1. {狗↦咆哮，狗↦怒吼，猫↦喵喵，牛↦哞哞}

2. {咆哮，怒吼}

3. {狗↦咆哮，狗↦怒吼，猫↦喵喵}

4. {狗↦咆哮，狗↦怒吼，猫↦喵喵}

8.26

$x \in \mathrm{dom} R \fatsemi S$

$\Leftrightarrow \exists z : Z \cdot x \mapsto z \in R \fatsemi S$

$\Leftrightarrow \exists z : Z \cdot \exists y : Y \cdot x \mapsto y \in R \wedge y \mapsto z \in S$

$\Rightarrow \exists z : Z \cdot \exists y : Y \cdot x \mapsto y \in R$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in R$

$\Leftrightarrow x \in \mathrm{dom}\ R$

8.27

$x \mapsto z \in X \lhd (R \fatsemi S)$

$\Leftrightarrow x \in X \wedge x \mapsto z \in R \fatsemi S$

$\Leftrightarrow x \in X \wedge \exists y : Y \cdot x \mapsto y \in R \wedge y \mapsto z \in S$

$\Leftrightarrow \exists y : Y \cdot x \in X \wedge x \mapsto y \in R \wedge y \mapsto z \in S$

$\Leftrightarrow \exists y : Y \cdot x \mapsto y \in X \lhd R \wedge y \mapsto z \in S$

$\Leftrightarrow x \mapsto z \in (X \lhd R) \fatsemi S$

8.28 A 和 D 是同类的，B 和 C 是异类的。

8.29 2 和 4 是自反的；其余两个关系不是，因为它们并非对每个自然数 n 都包含 (n, n)。

8.30 2 是自反的，1 不是。

8.31 根据定义，若 R 是自反的，则 $\forall x : X \cdot (x, x) \in R$ 成立。因此，一定有

$\{x : X \cdot (x, x)\} \subseteq R$

成立。

8.32 所有这些关系都是传递的。

8.33 1 是传递的，而 2 不是。

8.34 集合 $R \fatsemi R$ 的定义如下。

$R \fatsemi R = \{x, y : X \mid \exists z : X \cdot x \mapsto z \in R \wedge z \mapsto y \in R\}$

若 R 是传递的，那么由传递性的定义可知 R 满足如下性质。

$\forall x, y, z : X \cdot x \mapsto z \in R \wedge z \mapsto y \in R \Rightarrow x \mapsto y \in R$

由于这正定义了 $R \fatsemi R$ 的性质，所以有 $R \fatsemi R \subseteq R$。

8.35　1、3 和 4 是对称的。

8.36　1 是对称的，而 2 不是。

8.37　集合 $R^{\sim}$ 的定义如下所示。

$$R^{\sim}=\{x,\ y:X \mid x\mapsto y\in R\cdot y\mapsto x\}$$

若 R 是对称的，那么由对称性的定义可知，R 满足以下性质。

$$\forall x,\ y:X\cdot x\mapsto y\in R\Rightarrow y\mapsto x\in R$$

因为这正定义了 $R^{\sim}$ 的性质，所以有 $R^{\sim}=R$。

8.38　1 是非对称的。

8.39　2 是非对称的，而 1 不是非对称的。

8.40　1、2 和 3 是反对称的。

8.41　2 是反对称的，而 1 不是。

8.42　2 和 4 是完全的。

8.43　2 是完全的，而 1 不是。

8.44　"<"不是偏序，因为它不是自反的：对于任意的自然数 $n\in\mathbb{N}$ 都有 $n<n$ 这一论断不成立。

8.45　X 的子集上的关系 $\subseteq$ 是自反的，因为对于所有的 $S\in\mathbb{P}X$，有 $S\subseteq S$。其次，X 的子集上的关系 $\subseteq$ 是传递的，因为若 $R\subseteq S$ 且 $S\subseteq T$ 那么有 $R\subseteq T$。最后，X 的子集上的关系 $\subseteq$ 是反对称的，因为若 $R\subseteq S$ 且 $S\subseteq R$ 那么有 $R=S$。因此，这一关系构成 X 的子集上的偏序。

8.46　由于"="不是完全的，所以它不是全序：对于每个自然数的序偶 m 和 n，$m=n$ 不总成立。

8.47　为使关系 $\subseteq$ 构成 X 的子集上的全序，需要这一关系是反对称、传递和完全的。虽然它是反对称和传递的，但它不是完全的。考虑 X 的子集 S 和 T，并非对所有这样的子集都有 $S\subseteq T$ 或 $T\subseteq S$ 或 $S=T$。

8.48　因为 $\leqslant$ 不是对称的，所以它不是等价关系：并非对每对自然数 m 和 n 都有若 $m\leqslant n$ 则 $n\leqslant m$。

8.49　为使关系 $\subseteq$ 构成 X 的子集上的等价关系，需要这一关系是自反、传递且对称的。虽然它是自反和传递的，但它不是对称的。考虑 X 的两个子集 S 和 T，并不总有若 $S\subseteq T$ 则 $T\subseteq S$。

8.50

1. {猫↦猫，狗↦狗，老鼠↦老鼠}
2. {猫↦狗，狗↦老鼠，猫↦猫，狗↦狗，老鼠↦老鼠}
3. {猫↦狗，老鼠↦狗，猫↦猫，狗↦狗，老鼠↦老鼠}
4. {猫↦猫，狗↦狗，老鼠↦老鼠}

8.51

1. $\varnothing$
2. {猫↦狗，狗↦老鼠，猫↦老鼠}
3. {猫↦狗，老鼠↦狗}
4. {猫↦猫，狗↦狗，老鼠↦老鼠}

8.52

$$\begin{aligned}&\{a\mapsto b,\ b\mapsto c,\ c\mapsto d,\ d\mapsto e\}\\&\cup\\&\{a\mapsto c,\ b\mapsto d,\ c\mapsto e\}\\&\cup\\&\{a\mapsto d,\ b\mapsto e\}\\&\cup\\&\{a\mapsto e\}\\=&\{a\mapsto b,\ b\mapsto c,\ c\mapsto d,\ d\mapsto e,\ a\mapsto c,\ b\mapsto d,\ c\mapsto e,\ a\mapsto d,\ b\mapsto e,\ a\mapsto e\}\end{aligned}$$

8.53

1. {猫↦猫，老鼠↦老鼠，狗↦狗}
2. {猫↦狗，狗↦老鼠，猫↦老鼠，猫↦猫，老鼠↦老鼠，狗↦狗}
3. {猫↦狗，老鼠↦狗，猫↦猫，老鼠↦老鼠，狗↦狗}
4. {猫↦猫，狗↦狗，老鼠↦老鼠}

8.54

1. $\varnothing$
2. {猫↦狗，狗↦老鼠，狗↦猫，老鼠↦狗}
3. {猫↦狗，老鼠↦狗，狗↦猫，狗↦老鼠}
4. {猫↦猫，狗↦狗，老鼠↦老鼠}

8.55

1. $Name \leftrightarrow Yes\_No$

2. $(Name \times Yes\_No) \leftrightarrow Yes\_No$

3. $(Yes\_no \times Yes\_No) \leftrightarrow (Name \times Yes\_No)$

4. $Yes\_No \leftrightarrow (Name \times Name)$

8.56

1. $\{(1, 1), (1, 0), (0, 1), (0, 0)\}$

2. $\{(1, 1), (1, 0), (0, 1), (0, 0)\}$

3. $\{((1, 1), (0, 1))\}$

4. $\{((0, 1), (1, 1)), ((0, 0), (1, 0)), ((1, 0), (0, 1)), ((1, 1), (0, 0))\}$

5. $\{((1, 1), (1, 0)), ((1, 0)(1, 1)), ((0, 1), (0, 0)), ((0, 0), (0, 1))\}$

6. $\{(1, (0, 1)), (1, (0, 0)), (0, (1, 0)), (0, (1, 1))\}$

8.57

1. $husband = \{p, q : Person \mid (p, q) \in married \wedge male(p)\}$

2. $wife = \{p, q : Person \mid (p, q) \in married \wedge female(p)\}$

3. $mother = \{p, q : Person \mid (p, q) \in parent \wedge female(p)\}$

4. $father = \{p, q : Person \mid (p, q) \in parent \wedge male(p)\}$

5. $son = \{p, q : Person \mid (q, p) \in parent \wedge male(p)\}$

6. $daughter = \{p, q : Person \mid (q, p) \in parent \wedge female(p)\}$

7. $brother = \{p, q : Person \mid (p, q) \in sibling \wedge male(p)\}$

8. $sister = \{p, q : Person \mid (p, q) \in sibling \wedge female(p)\}$

8.58

1. $aunt = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in sister \wedge (r, q) \in parent\}$

2. $uncle = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in brother \wedge (r, q) \in parent\}$

3. $nephew = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in son \wedge (r, q) \in sibling\}$

4. $niece = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in daughter \wedge (r, q) \in sibling\}$

5. $cousin = \{p, q : Person \mid \exists r, s : Person \cdot (r, p) \in parent \wedge (s, q) \in parent \wedge (r, s) \in sibling\}$

6. $father\_in\_law = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in father \wedge (r, q) \in married\}$

7. $mother\_in\_law = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in mother \wedge (r, q) \in married\}$

8. $son\_in\_law = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in husband \wedge (r, q) \in daughter\}$

9. $daughter\_in\_law = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in wife \wedge (r, q) \in son\}$

8.59

1. $stepfather = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in husband \wedge (r, q) \in mother \wedge (p, q) \notin father\}$

2. $stepmother = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in wife \wedge (r, q) \in father \wedge (p, q) \notin mother\}$

3. $stepdaughter = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in daughter \wedge (r, q) \in married \wedge (p, q) \notin daughter\}$

4. $stepson = \{p, q : Person \mid \exists r : Person \cdot (p, r) \in son \wedge (r, q) \in married \wedge (p, q) \notin son\}$

5. $stepbrother = \{p, q : Person \mid \exists r, s : Person \cdot (r, s) \in married \wedge (p, r) \in son \wedge (s, q) \in parent \wedge (p, q) \notin brother \wedge p \neq q\}$

6. $stepsister = \{p, q : Person \mid \exists r, s : Person \cdot (r, s) \in married \wedge (p, r) \in daughter \wedge (s, q) \in parent \wedge (p, q) \notin sister \wedge p \neq q\}$

7. $grandmother = \{p, q : Parent \mid \exists r : Person \cdot (p, r) \in mother \wedge (r, q) \in parent\}$

8. $grandfather = \{p, q : Parent \mid \exists r : Person \cdot (p, r) \in father \wedge (r, q) \in parent\}$

8.60

1. $child = parent^{\sim}$

2. $parent \underset{9}{\circ} parent(\mid \{p\} \mid)$

3. $child \underset{9}{\circ} child \underset{9}{\circ} child(\mid \{p\} \mid)$

4. $child^{+}(\mid \{p\} \mid)$

5. dom $mother$

6. dom $father \rhd \{p : People \mid female(p)\}$

7. dom $brother \cap$ ran $brother$

8. dom $grandmother \rhd \{p : People \mid male(p)\}$

9. dom $grandfather \cap$ (dom $father \rhd \{p : People \mid male(p)\}$)

8.61

1. $R = \{n : \mathbb{N} \mid n > 0 \wedge n < 5 \cdot (n, 2 \times n)\}$

   $S = \{n : \mathbb{N} \mid n > 0 \wedge n < 5 \cdot (n, n)\}$

2. dom $R = \{1, 2, 3, 4\}$

3. ran $S = \{1, 2, 3, 4\}$

4. $R^{\sim} = \{(2, 1), (4, 2), (6, 3), (8, 4)\}$

5. $R \underset{9}{\circ} S = \{(1, 2), (2, 4)\}$

6. $S \underset{9}{\circ} R = R$

8.62

1. $\{1, 2, 3, 4\}$

2. $\{1, 4, 9, 16\}$

3. $\{2 \mapsto 1, 4 \mapsto 2, 6 \mapsto 3, 8 \mapsto 4\}$

4. $\{1 \mapsto 4, 2 \mapsto 16\}$

5. $\{1 \mapsto 2, 2 \mapsto 8\}$

6. $\{1 \mapsto 2, 2 \mapsto 3\}$

7. $\{3 \mapsto 4, 4 \mapsto 5\}$

8. $\{1 \mapsto 2\}$

9. $\{2 \mapsto 3, 3 \mapsto 4, 4 \mapsto 5\}$

10. $\{1, 4\}$

11. $\{2, 4\}$

12. $\{1\}$

8.63

1. *loves* 不是自反的，因为布赖恩↦布赖恩∉*loves*。

2. *loves* 不是对称的，因为布赖恩↦杰姬∈*loves*，但杰姬↦布赖恩∉*loves*。

3. *loves* 是传递的，因为有 $R^{+} = R$。

8.64

1. {布赖恩↦布赖恩，布赖恩↦杰姬，杰姬↦杰姬}

2. {杰姬↦布赖恩，布赖恩↦杰姬，杰姬↦杰姬}

8.65

1. {费杜↦咆哮，费杜↦怒吼}

2. {费杜↦咆哮，费杜↦怒吼，赖西↦咆哮}

3. {费杜↦咆哮，费杜↦怒吼，赖西↦咆哮，布朗↦喵喵}

4. {费杜↦咆哮，费杜↦怒吼，赖西↦咆哮}

5. {费杜↦咆哮，赖西↦咆哮}

6. {费杜↦怒吼，戴西↦哞哞}

7. {费杜↦咆哮，费杜↦怒吼，赖西↦咆哮，戴西↦哞哞}

8. {费杜↦怒吼，戴西↦哞哞}

9. {咆哮，怒吼}

10. ∅

8.66 1 既是传递的，又是对称的。2 和 3 是传递的，但不是对称的。而 4 是对称的，但不是传递的。

8.67

1. ∅

2. {(吉姆，乔)，(乔，比尔)，(吉姆，比尔)，(史蒂夫，瑞克)，(乔，吉姆)，(比尔，乔)，(比尔，吉姆)，(瑞克，史蒂夫)}

3. {(吉姆，乔)，(乔，比尔)，(吉姆，比尔)，(乔，吉姆)，(比尔，乔)，(比尔，吉姆)}

4. {(瑞克，史蒂夫)，(史蒂夫，瑞克)，(吉姆，比尔)，(比尔，吉姆)}

8.68

1. ∅

2. {(吉姆，乔)，(乔，比尔)，(吉姆，

比尔)，(史蒂夫，瑞克)}

3. {(吉姆，乔)，(乔，比尔)，(吉姆，比尔)}

4. {(瑞克，史蒂夫)，(史蒂夫，瑞克)，(吉姆，比尔)，(比尔，吉姆)，(瑞克，瑞克)，(史蒂夫，史蒂夫)，(吉姆，吉姆)，(比尔，比尔)}

8.69

1. 是真的。

2. 不是真的；反例如下所示。

$\{a \mapsto b, c \mapsto e\} \underset{g}{\circ} \{b \mapsto c, e \mapsto d\} = \{a \mapsto c, c \mapsto d\}$

3. 不是真的；反例如下所示。

$\{a \mapsto b, b \mapsto a\} \underset{g}{\circ} \{b \mapsto c, c \mapsto b\} = \{a \mapsto c\}$

8.70 为使⇔构成所有命题的集合上的等价关系，需要⇔是自反、对称且传递的。给定任意的命题 p，$p \Leftrightarrow p$ 为真。因此，⇔是自反的。给定任意的命题 p 和 q，若 $p \Leftrightarrow q$ 为真，则 $q \Leftrightarrow p$ 为真。因此，⇔是对称的。给定任意的命题 p、q 和 r，若 $p \Leftrightarrow q$ 为真且 $q \Leftrightarrow r$ 为真，则 $p \Leftrightarrow r$ 为真。因此，⇔是传递的。由上可知，⇔是所有命题的集合上的等价关系。

8.71 是的。R^+ 是传递的，所以它的传递闭包与其相同。因此，$(R^+)^+$ 与 R^+ 相同。

第9章 函　　数

前一章讨论了关系的概念。在那里，我们看到，在一个关系中，源头中的任意元素都能映射到目标中任何数目的元素。本章考虑一种特殊的关系：函数。

9.1 一种特殊的关系

前面曾介绍过类型为$\mathbb{N} \leftrightarrow \mathbb{N}$的关系*less_than*，其定义如下。

$less\_than = \{m, n : \mathbb{N} \mid m < n\}$

回顾关系的像的记法，可以写成$2 \in less_than(\mid \{1\} \mid)$；等价地，也可以写成$(1, 2) \in less_than$或者$1 \mapsto 2 \in less_than$。

类型同样为$\mathbb{N} \leftrightarrow \mathbb{N}$的关系*one_less_than*可以定义如下。

$one\_less\_than = \{m, n : \mathbb{N} \mid m+1 = n\}$

另外，可以写成$2 \in one_less_than(\mid \{1\} \mid)$；在这种情况下，也可以书写如下。

$one\_less\_than(\mid \{1\} \mid) = \{2\}$

进而，很明显，有$one_less_than \subseteq less_than$。

在第二个关系中，源头中的每一个元素恰好映射到目标中的一个元素：0映射到1，1映射到2，2映射到3，等等。这样，把具有这种特殊性质的关系称作函数(function)：源头中的每一个元素都至多映射到目标的一个元素(或者等价地说，定义域中的每个元素都恰好映射到值域中的一个元素)。

例 9.1　考虑如下定义的关系*holidays*。

holidays={吉姆↦肯尼亚，琼↦埃及，杰夫↦西班牙}

可以对"杰夫是在哪里度假的?"这样的问题给出一个唯一的答案(如果答案存在的话)，从而使得这一关系成为函数。

现在我们考虑如下定义的关系*souvenir*。

souvenir={琼↦金字塔，杰夫↦驴，杰夫↦帽子}

这里，如果我们问"杰夫带回了什么纪念品?"，可以得到两个可能的答案：一头驴和一顶帽子，因此，这个关系不符合函数所要求的特殊性质，也就是说，不是定义域中的每个元素都映射到值域中的唯一一个元素。

□

例 9.2　如下定义的恒等关系$Id[X] : X \leftrightarrow X$是一个函数。

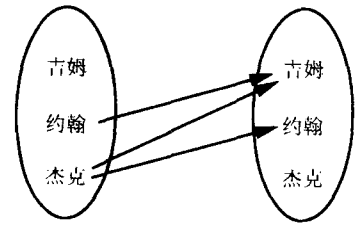
$Id[X] = \{x : X \bullet x \mapsto x\}$

这里，源头中的每一个元素都恰好映射到目标中的一个元素。　□

我们把从X到Y的所有函数组成的集合记作$X \rightarrow Y$。注意，$X \rightarrow Y$是$X \leftrightarrow Y$的子集。事实上，可以按照下面所示的方法，通过X到Y的所有关系的集合，来定义从X到Y的所有函数组成的集合。定义如下。

$X \rightarrow Y = \{r : X \leftrightarrow Y \mid \forall x : \mathrm{dom}\ r \bullet \exists_1 y : \mathrm{ran}\ r \bullet x \mapsto y \in r\}$

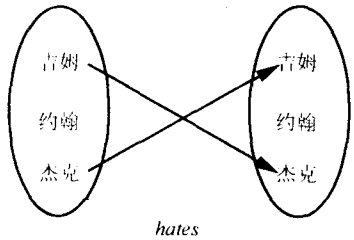
利用关系的可视化图形，确定一个关系是否是函数是很直接的。例如，考虑如下所示的关系*likes*。



likes

这个关系不是函数，因为杰克可以映射到吉姆和约翰。在图形上，它带有分岔的箭头。

另一方面，考虑如下所示的关系*hates*。这个关系是函数，因为定义域的每一个元素



hates

都恰好映射到值域中的一个元素，它没有分岔的箭头。

因此，对于给定的关系，可以通过检测其关系图中是否有分岔箭头来可视地决定这个关系是否是函数。

练习 9.1 下列哪些关系是函数？

1. $\varnothing$
2. $\{(1, 2), (2, 3), (3, 1)\}$
3. $\{(1, 2), (2, 1)\}$
4. $\{(1, 2), (1, 1)\}$ □

练习 9.2 列举出从$\{1, 2\}$到$\{a, b\}$的所有函数。 □

练习 9.3 函数的逆是否是函数？ □

练习 9.4 函数的子集是否是函数？ □

练习 9.5 给定类型相同的函数 f 和 g，下列关系中哪些是函数？

1：$f \cap g$

2：$f \cup g$

3：$f \setminus g$ □

9.2 全函数

再次考虑如下定义的函数 *holidays*。

holidays＝{吉姆↦肯尼亚，琼↦埃及，杰夫↦西班牙}

进一步假设这个函数的源头和目标分别为如下给出的集合 *People* 和 *Places*。

People＝{吉姆，琼，杰夫，杰克}

Places＝{肯尼亚，埃及，西班牙，希腊}

这个关系是函数的关键在于，如果被问道"p 在哪里度假？"的话（p 为吉姆、琼或杰夫中的一个），我们能够给出唯一的答案。然而，如果问题是"杰克在哪里度假？"，那么就不能给出正确的答案，因为虽然杰克出现在这一函数的源头，但没有出现在这一函数的定义域中。

在函数 f 中，如果源头的每一个元素都恰好映射到目标的一个元素，那么我们称 f 是全函数（total function）。如果 $f \in X \rightarrow Y$，f 是全函数当且仅当 $\mathrm{dom}\ f = X$。

我们用 $X \rightarrow Y$ 表示所有从 X 到 Y 的全函数组成的集合。注意，下式成立。

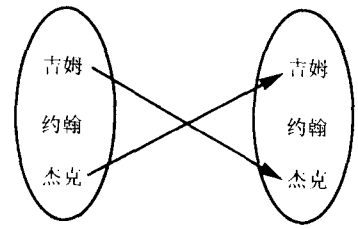
$X \rightarrow Y \subseteq X \rightarrow Y$

还可以如下定义从 X 到 Y 的全函数组成的集合。

$X \rightarrow Y = \{r : X \leftrightarrow Y \mid r \in X \rightarrow Y \wedge \mathrm{dom}\ r = X\}$

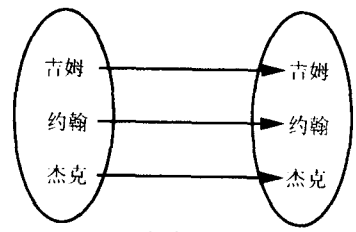
在关系的可视化表示中，可以很容易地确定一个函数是否是全函数：一个函数是全函数当且仅当对于它的源头的每一个元素，都有从这个元素出发的箭头。

例 9.3 考虑如下所示的关系 *hates*。



hates

这个关系不是全函数，因为约翰没有映射到目标中的任何元素。相反，如下所示的关系 *looks_like* 是全函数。



looks_like

这是因为，源头中的每一个元素都恰好映射到目标中的一个元素。 □

例 9.4　考虑集合{1，2，3}上的如下同类关系。

$\{1 \mapsto 1, 1 \mapsto 2, 2 \mapsto 3\}$

这个关系不是全函数，因为 3 没有映射到目标中的元素。而如下所示的{1，2，3}上的同类关系是全函数，因为源头中的每一个元素都恰好映射到目标中的一个元素。

$\{1 \mapsto 1, 3 \mapsto 2, 2 \mapsto 3\}$　□

练习 9.6　假设有如下定义的集合 *People* 和 *Mode*。

People={巴利，奈杰尔，大卫}

Mode={汽车，自行车，火车}

下列哪些类型为 *People*↔*Mode* 的关系是全函数？

1. ∅

2. {巴利↦汽车，奈杰尔↦汽车}

3. {巴利↦汽车，奈杰尔↦汽车，大卫↦火车}

4. {巴利↦汽车，巴利↦自行车，奈杰尔↦汽车，大卫↦火车}　□

练习 9.7　列举出从{1，2}到{a，b}的所有的全函数。　□

练习 9.8　全函数的逆(关系)是否一定是函数？　□

练习 9.9　给定类型相同的全函数 f 和 g，$f \cap g$ 是否一定是全函数？　□

9.3　函数的作用

就像我们已经看到的那样，函数具有把其源头的每一个元素映射到目标中的至多一个元素的效用。因此，对于函数 f 和元素 x，$f(|\{x\}|)$的值总是或者是空集或者是由 x 映射到的值组成的单集。正因为此，我们可以考虑对于特定元素的函数的作用(function application)；这仅仅是关系的像运算的特殊形式。

给定函数 $f \in X \rightarrow Y$ 和某个 $x \in X$，我们用 fx 表示 f 对 x 的作用。此时，$fx = y$ 当且仅当$(x, y) \in f$[①]。

例 9.5　考虑如下定义的函数 *pets*。

pets={爱米丽↦金鱼，理查德↦沙鼠，迈克尔↦猫}

这里，

pets 爱米丽＝金鱼

且

pets 理查德＝沙鼠　□

当然，仍可以把关系的像运算与函数相关联，但是，当我们处理函数时，考虑函数的作用更有意义。

也可以如下形式地定义函数的作用。给定函数 $f \in X \rightarrow Y$ 和元素 $x \in X$，f 对 x 的函数的作用如下所示。

$fx = (\mu y : Y \mid x \mapsto y \in f)$

当然，如果 x 不在 f 的定义域中出现，那么 fx 是未定义的。注意，只可以对函数实施函数的作用；关系上的函数作用是未定义的。

练习 9.10　考虑下列函数。

likes={肯↦邓肯，邓肯↦约翰，理查德↦贝基，贝基↦理查德}

dislikes={肯↦贝基，贝基↦肯，邓肯↦肯}

下列函数作用中哪些是有定义的？

1. *likes* 肯

2. (*likes*~)肯

3. *dislikes* 肯

4. (*dislikes*~)肯

5. (*likes* $\underset{9}{\circ}$ *likes*)肯

6. (*dislikes* $\underset{9}{\circ}$ *dislikes*)肯　□

练习 9.11　考虑如下所给的函数 *pets*。

pets={爱米丽↦金鱼，理查德↦沙鼠，迈克尔↦猫}

① 在很多教科书中，函数 f 对项 x 的作用写作 $f(x)$，而不是 fx。这两种形式都是合理的，但是本书采用后者。

1. *pets* 理查德的值是什么？

2. *pets* 迈克尔的值是什么？

3. *pets*⁻ 金鱼的值是什么？

4. ({爱米丽} ⩤ *pets*) 爱米丽的值是什么？ □

练习 9.12 假设有练习 9.11 的关系 *pets*，给出下列集合描述的结果。

1. $\{p : pets\}$

2. $\{p : pets \cdot p.1\}$

3. $\{p : pets \mid p.2 = 金鱼\}$

4. $\{p : pets \mid p.2 = 金鱼' \cdot p.1\}$

5. $\{p : pets \mid p.1 = 爱米丽 \cdot pets\ p.1\}$

6. $\{p : pets \mid p.1 = 爱米丽 \cdot p.2\}$ □

9.4 覆盖

如果要用函数来对真实世界中的对象建立模型，就需要有一种更新世界描述的手段。总之，尽管与某些函数相关的值可能总是不变的(例如，自然数的加法)，但是，我们也许期望与其他一些函数相关的值，特别是那些与真实世界系统的建模相关的函数的值，在必要时发生必要的变化。

例如，考虑如下定义的函数 *wage*。

$wage = \{托马斯 \mapsto 23\ 000, 埃文斯 \mapsto 18\ 750, 琼斯 \mapsto 18\ 750, 布朗 \mapsto 28\ 000\}$

如果埃文斯的薪水每年增加 50 英镑，那么我们希望 *wage* 的新描述将是如下所示的形式。

$new\_wage = \{托马斯 \mapsto 23\ 000, 埃文斯 \mapsto 18\ 800, 琼斯 \mapsto 18\ 750, 布朗 \mapsto 28\ 000\}$

与其从头定义一个新函数 *new_wage*，更可取的做法是，使用 *wage* 来定义 *new_wage*；利用函数覆盖运算符(function override operator)⊕可以做到这一点。

给定两个相同类型的函数 f 和 g。$f \oplus g$ 表示用 g 覆盖 f，我们称 f 被 g 覆盖了。这个运算符可以根据 g 的值有效地更新 f。因此，有

$new\_wage = wage \oplus \{埃文斯 \mapsto 18\ 800\}$

上面的表达式反映了我们所希望的变化。这里，*wage* 被函数 $\{埃文斯 \mapsto 18\ 800\}$ 所更新：我们希望与埃文斯相关联的薪水发生变化，而其余薪水保持不变。

可以形式地定义覆盖如下。

$f \oplus g = (\mathrm{dom}\ g \lhd f) \cup g$

如果考虑下面的事实，那么这个定义的含义就很清楚了。我们考虑的源头中存在三类元素：第一类是出现在 f 的定义域中但不出现在 g 的定义域中的元素；第二类是同时出现在 f 和 g 的定义域中的元素；第三类是不出现在 f 的定义域中但出现在 g 的定义域中的元素。我们依次考虑每种情况。

如果映射 $a_1 \mapsto b_1$ 出现在 f 中，而且 a_1 不出现在 g 的定义域中，那么这个从 a_1 到 b_1 的映射不受 g 对 f 的覆盖的影响。因此，$a_1 \mapsto b_1$ 出现在 $f \oplus g$ 中。

其次，如果映射 $a_2 \mapsto b_2$ 出现在 f 中，而且 a_2 出现在 g 的定义域中，那么从 a_2 到 b_2 的映射受到 g 对 f 的覆盖的影响。因此，$a_2 \mapsto ga_2$ 出现在 $f \oplus g$ 中。

最后，如果映射 $a_3 \mapsto b_3$ 出现在 g 中，而且 a_3 不出现在 f 的定义域中，那么映射 a_3 到 b_3 被加入到结果函数中。因此，$a_3 \mapsto b_3$ 出现在 $f \oplus g$ 中。

例 9.6 考虑如下给出的函数 f 和 g。

$f = \{0 \mapsto 0, 1 \mapsto 1\}$

$g = \{0 \mapsto 1, 2 \mapsto 3\}$

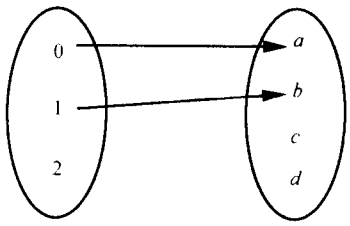
如果我们依次考虑源头的每一个元素，有如下结果。

- 在 f 中 0 映射到 0，在 g 中 0 映射到 1，所以 $(f \oplus g)0 = 1$；
- 在 f 中 1 映射到 1，1 不出现在 g 的定义域中，所以 $(f \oplus g)1 = 1$；
- 2 不出现在 f 的定义域中，且在 g 中 2 映射到 3，所以 $(f \oplus g)2 = 3$。

因此，g 对 f 的覆盖如下所示。

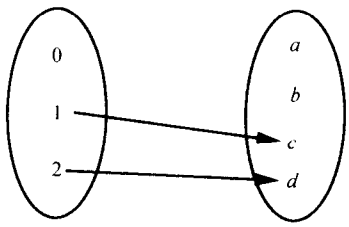
$f \oplus g = \{0 \mapsto 1,\ 1 \mapsto 1,\ 2 \mapsto 3\}$ □

例 9.7　考虑下图给出的函数 f。



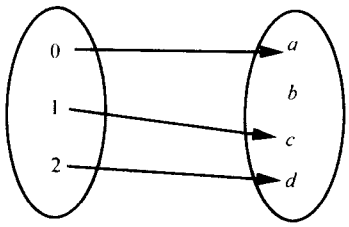
f

现在考虑下图给出的函数 g。



g

g 对 f 的覆盖 $f \oplus g$ 如下图所示。



□

练习 9.13　假设有如下定义的函数。

$A = \{$琼$\mapsto$咖啡，戴夫$\mapsto$橙汁$\}$

$B = \{$史蒂夫$\mapsto$茶，戴夫$\mapsto$咖啡$\}$

$C = \{$琼$\mapsto$橙汁，史蒂夫$\mapsto$咖啡$\}$

计算下列各题。

1. $A \oplus B$
2. $B \oplus A$
3. $B \oplus C$
4. $(A \oplus B) \oplus C$
5. $A \oplus (B \oplus C)$

□

练习 9.14　证明下列各式。

1. $f \oplus f = f$
2. $f \oplus \varnothing = f$
3. $\varnothing \oplus f = f$

□

9.5　函数的性质

正如函数是特殊类型的关系，我们也可以定义特殊类型的函数。在本节，我们介绍三种特殊类型的函数：单射(injection)、满射(surjection)和双射(bijection)。

9.5.1　单射

称函数 $f \in X \rightarrow Y$ 是单射函数或单射，当且仅当，对于 Y 中的每一个元素，X 中最多有一个元素映射到它。我们将类型为 $X \leftrightarrow Y$ 的所有单射函数组成的集合记作 $X \rightarrowtail Y$。显然，有

$$X \rightarrowtail Y \subseteq X \rightarrow Y$$

另外，我们把类型为 $X \leftrightarrow Y$ 的所有全单射(total injection)的函数组成的集合记作 $X \rightarrowtail\!\!\!\!\rightarrow Y$。形式地定义上述集合如下。

$$X \rightarrowtail\!\!\!\!\rightarrow Y$$
$$= \{r : X \leftrightarrow Y \mid r \in X \rightarrow Y \wedge \forall y : \operatorname{ran} r \bullet (\exists_1 x : \operatorname{dom} r \bullet (x, y) \in r)\}$$
$$X \rightarrowtail\!\!\!\!\rightarrow Y = X \rightarrowtail Y \cap X \rightarrow Y$$

例 9.8　假设有如下给出的集合 *Person* 和 *Colour*。

Person$=\{$罗宾，马特，辛巴$\}$

Colour$=\{$绿色，蓝色，黄色$\}$

假设有如下给出的函数 *favourite*。

favourite$=\{$罗宾$\mapsto$绿色，马特$\mapsto$蓝色，辛巴$\mapsto$黄色$\}$

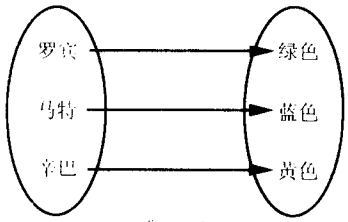
这里，*favourite* 显然是单射：因为对于目标中的每一个元素，源头中最多有一个元素映射到这一元素。而且，这一函数是全单射，因为 *dom favourite* $=$ *Person*。

而另一方面，下面的函数 *car_colour* $\in$ *Person* $\rightarrow$ *Colour* 不是单射，因为罗宾和马特都映射到蓝色。

car_colour$=\{$罗宾$\mapsto$蓝色，马特$\mapsto$蓝色$\}$

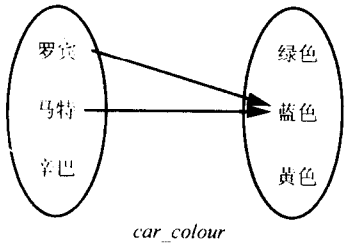
□

可以使用土豆图来判定给定关系是否是单射。例如，考虑下图所示的 $favourite \in Person \rightarrow Colour$ 的土豆图。



favourite

如上图所示，因为没有汇集到一起的箭头，所以可以得出这一函数是单射的结论。而在下图所示的 *car_colour* 的土豆图中有汇集到一起的箭头：有两个箭头指向蓝色，因此，这一函数不是单射。



car_colour

因此，是否存在汇集到一起的箭头可以判定给定函数是否是单射。

练习 9.15 下列哪些函数是单射？

1. $\varnothing$
2. $\{1 \mapsto a, 1 \mapsto b\}$
3. $\{1 \mapsto a, 2 \mapsto a\}$
4. $\{1 \mapsto a, 2 \mapsto b\}$ □

练习 9.16 给出从 $\{0, 1\}$ 到 $\{a, b\}$ 的所有单射的集合。 □

练习 9.17 给出从 $\{0, 1\}$ 到 $\{a, b\}$ 的所有全单射的集合。 □

练习 9.18 单射的逆一定是单射吗？ □

练习 9.19 全单射的逆一定是全单射吗？ □

9.5.2 满射

称函数 $f \in X \rightarrow Y$ 是满射函数或满射，当且仅当，对于 Y 中的每一个元素 y，都有 X 中的元素 x，x 映射到 y。我们将类型为 $X \leftrightarrow Y$ 的所有满射函数组成的集合记作 $X \twoheadrightarrow Y$。显然有

$$X \twoheadrightarrow Y \subseteq X \rightarrow Y$$

另外，我们将类型为 $X \leftrightarrow Y$ 的所有全满射组成的集合记作 $X \twoheadrightarrow Y$。形式地定义这些集合如下。

$$X \twoheadrightarrow Y = \{r : X \leftrightarrow Y \mid r \in X \rightarrow Y \wedge \operatorname{ran} r = Y\}$$

$$X \twoheadrightarrow Y = X \twoheadrightarrow Y \cap X \rightarrow Y$$

例 9.9 再一次考虑如下给出的集合 *Person* 和 *Colour*，以及函数 $favourite \in Person \rightarrow Colour$。

$Person = \{$罗宾，马特，辛巴$\}$

$Colour = \{$绿色，蓝色，黄色$\}$

$favourite = \{$罗宾$\mapsto$绿色，马特$\mapsto$蓝色，辛巴$\mapsto$黄色$\}$

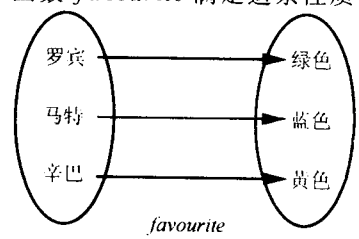
在这里，*favourite* 是满射，因为对于目标集合中的每一个元素，都存在源头的元素映射到它。而且，这一函数是全满射，因为 $\operatorname{dom} favourite = Person$。

我们再一次考虑如下给出的函数 $car\_colour \in Person \rightarrow Colour$。

$car_colour = \{$罗宾$\mapsto$蓝色，马特$\mapsto$蓝色$\}$

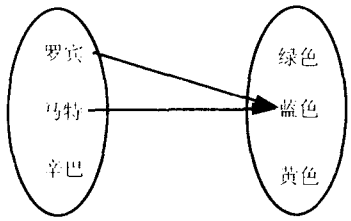
它不是满射，因为在定义域中没有映射到绿色和黄色的元素。 □

同样，可以用土豆图来判定一个关系是否是满射：一个关系具有满射性质，仅当它的目标中的每一个元素都被映射到。由下图可知，函数 *favourite* 满足这条性质。



favourite

然而，由下图可知，*car_colour* 不满足这条性质。



car_colour

因此，可以通过考察目标中的每一个元素是否都被映射到来判定给定函数是否是满射。

练习 9.20　下列类型为 $\{a, b, c\} \leftrightarrow \{0, 1\}$ 的关系中哪些是满射？

1. $\varnothing$
2. $\{a \mapsto 0, b \mapsto 1\}$
3. $\{a \mapsto 0, b \mapsto 1, c \mapsto 1\}$
4. $\{a \mapsto 0, b \mapsto 0, b \mapsto 1\}$ □

练习 9.21　给出从 $\{0, 1\}$ 到 $\{a, b\}$ 的所有满射的集合。 □

练习 9.22　给出从 $\{0, 1\}$ 到 $\{a, b\}$ 的所有全满射的集合。 □

练习 9.23　如果 X 有 m 个元素，且 Y 有 n 个元素，那么为使 $f \in X \rightarrow Y$ 成为满射 m 和 n 间必须有什么关系？ □

练习 9.24　假设 X 和 Y 都有 n 个元素，且 $f \in X \rightarrow Y$ 是全满射，那么 f 一定是单射吗？ □

练习 9.25　满射的逆一定是满射吗？ □

练习 9.26　全单射的逆一定是满射吗？ □

9.5.3　双射

称函数 $f \in X \rightarrow Y$ 是双射函数或双射，当且仅当 f 既是单射又是满射。我们将类型为 $X \leftrightarrow Y$ 的所有双射函数组成的集合记作 $X \rightarrowtail\!\!\!\!\twoheadrightarrow Y$。显然有

$$X \rightarrowtail\!\!\!\!\twoheadrightarrow Y \subseteq X \rightarrow Y$$

另外，我们将类型为 $X \leftrightarrow Y$ 的所有全双射组成的集合记作 $X \twoheadrightarrow Y$。形式地定义这些集合如下。

$$X \rightarrowtail\!\!\!\!\twoheadrightarrow Y = X \rightarrowtail Y \cap X \twoheadrightarrow Y$$

$$X \twoheadrightarrow Y = X \rightarrowtail\!\!\!\!\twoheadrightarrow Y \cap X \rightarrow Y$$

例 9.10　再次考虑如下给出的函数 $favourite \in Person \rightarrow Colour$。

$$favourite = \{罗宾 \mapsto 绿色, 马特 \mapsto 蓝色, 辛巴 \mapsto 黄色\}$$

它是双射，因为它既是单射又是满射。而且，它是全双射，因为 $\mathrm{dom}\ favourite = Person$。相反，如下定义的函数 $car\_colour \in Person \rightarrow Colour$ 不是双射，因为它既不是满射也不是单射。

$$car\_colour = \{罗宾 \mapsto 蓝色, 马特 \mapsto 蓝色\}$$ □

例 9.11　回忆我们在第 5 章对布尔代数的同构的讨论。对于同构的两个布尔代数，把其中一个布尔代数转换成另一个布尔代数的函数必须是全双射。 □

练习 9.27　下列类型为 $\{a, b, c, d\} \leftrightarrow \{0, 1, 2\}$ 的函数中哪些是双射？

1. $\varnothing$
2. $\{a \mapsto 0, b \mapsto 1\}$
3. $\{a \mapsto 0, b \mapsto 0, c \mapsto 1, d \mapsto 2\}$
4. $\{a \mapsto 0, b \mapsto 1, c \mapsto 2\}$ □

练习 9.28　给出从 $\{0, 1\}$ 到 $\{a, b\}$ 的所有双射的集合。 □

练习 9.29　假设 X 有 m 个元素，且 Y 有 n 个元素。如果 $f \in X \rightarrow Y$ 是全双射，那么 m 和 n 之间一定满足什么关系？ □

练习 9.30　双射的逆一定是双射吗？ □

练习 9.31　全双射的逆一定是全双射吗？ □

9.6　递归函数

递归函数这一本书较高级的论题最初是在第 2 章介绍的。在那里，我们对 Peano 算

术如下定义一个函数 *mult*。

$mult(0, n)=0$

$mult(0+1, n)=n$

$mult(m+1, n)=n+mult(m, n)$

这里，用0乘 n 的结果为0，用0+1乘 n 的结果为 n。另一方面，如果与 n 相乘的数大于0+1，那么，可以通过 *mult* 来定义 *mult* 的结果。

任意由其自身定义的函数(*mult* 就是一个典型的例子)称作*递归函数*(recursively defined function)。例如，计算包含在给定自然数中的数字的数目的函数 *digits* 的定义如下。

$digits(n)=1$，如果 $n<10$

$digits(n)=1+digits(n \text{ div } 10)$，其他

这里，对 *digits*(397)的计算如下。

$$\begin{aligned} digits(397) &= 1+digits(39) \\ &= 1+(1+digits(3)) \\ &= 1+(1+1) \\ &= 3 \end{aligned}$$

因此，通过 *digits*(39)定义了 *digits*(397)，同样，使用 *digits*(3)定义了 *digits*(39)，最后，使用1给出 *digits*(3)的结果。可以看到，每一次对 *digits* 的调用都对 *digits*(397)的最后结果产生了影响。

再举一个例子，可以通过函数 *add* 构造 *mult* 的新定义，而函数 *add* 本身也是递归定义的。

$$\begin{aligned} mult(0,n) &= 0 \\ mult(0+1,n) &= n \\ mult(m+1,n) &= add(n,mult(m,n)) \\ add(0,n) &= n \\ add((m+1),n) &= add(m,n)+1 \end{aligned}$$

因此，在Peano算术中，可以如下递归地计算0+1+1乘以0+1+1+1的结果。

$$\begin{aligned} & mult(0+1+1, 0+1+1+1) \\ =& add(0+1+1+1, mult(0+1, 0+1+1+1)) \\ =& add(0+1+1+1, 0+1+1+1) \\ =& add(0+1+1, 0+1+1+1)+1 \\ =& add(0+1, 0+1+1+1)+1+1 \\ =& add(0, 0+1+1+1)+1+1+1 \\ =& 0+1+1+1+1+1+1 \end{aligned}$$

递归函数的一般形式如下：

$f(n)=x$，若 $n=m$

$\quad =y+f(n')$，其他

一般情况下，在递归函数的定义中，存在一个或多个简单或"基底"情况，以及一个或多个通过函数自身定义的复杂情况。上面的表达式里，$n=m$ 是基底，当计算到达这一步时，递归终止。对于更复杂的情况，使用 $y+f(n')$ 来定义 $f(n)$。一般来说，n' 是"小于"n、某种意义上说是更接近 m 的值。

只有小心地选择基底情况，并确保递归定义的每一步都更靠近基底情况，才能保证递归过程不会无休止地进行下去。因此，称满足下列性质的递归定义 f 为*良定义的*(well-defined)。

1. 存在称为*基值*(base value)的某些值，对于这些值，这一函数不引用其自身。

2. 每当函数引用其自身时，函数的自变量都更靠近基值。

在函数 *digits* 的例子中，有

$digits(n)=1$，若 $n<10$

充当着基底的角色，而下式

$digits(n)=1+digits(n \text{ div } 10)$

使用 $digits(n \text{ div } 10)$ 定义 $digits(n)$，其中 $n \text{ div } 10$ 比 n 更靠近基底。

例 9.12 可以如下递归定义阶乘函数!。

$0!=1$

$1!=1$

$n!=n\times(n-1)!$，若 $n>1$

而且，这一递归函数是良定义的。 □

练习 9.32 为什么下面这个递归函数不是良定义的？

$f(0)=1$

$f(n)=f(n+1)+1$，若 $n>0$ □

练习 9.33　为什么下面这个递归函数不是良定义的？

$f(1)=1$

$f(n)=f(n-2)+1$，若 $n>1$　□

练习 9.34　定义计算给定自然数中 0 的数目的递归函数。　□

练习 9.35　定义对给定自然数中出现的所有数字求和的递归函数。例如，这一函数作用于 347 的结果为 $3+4+7=14$。　□

练习 9.36　定义对给定自然数中出现的所有数字求积的递归函数。例如，这一函数作用于 347 的结果为 $3\times4\times7=84$。　□

练习 9.37　定义返回自然数中最大数字的递归函数。例如，这一函数作用 347 的结果为 7。可以假设存在函数 *greater*，*greater*(m，n)返回 m 和 n 中较大的数。　□

9.7　附加练习

练习 9.38　一个学校的数据库中保存有学生所选的课程，以及每一个老师所负责的所有学生。给出能表示这些信息的两个函数。

练习 9.39　函数 f 对于每个英文短语计算组成这一短语的不同字母的数目。下列各函数值是什么？

1. f(simple)
2. f(slightly harder)
3. f(c'est impossible)

练习 9.40　下列定义中哪些是函数？

1. $\varnothing$
2. {佛瑞德↦猫，鲍勃↦狗，佛瑞德↦金鱼}
3. {佛瑞德↦猫，鲍勃↦狗，琼↦狗}

练习 9.41　部分函数和全函数之间有什么差异？请举例说明。

练习 9.42　假设有如下定义的函数。

holiday={吉姆↦冰岛，爱米丽↦坦纳利佛，戴夫↦新加坡，史蒂夫↦冰岛，阿里↦希腊，贝基↦希腊}

likes={吉姆↦{冷}，爱米丽↦$\varnothing$，戴夫↦{冷，热，潮湿}，史蒂夫↦{冷，热，潮湿}，阿里↦{热}，贝基↦{冷}}

climate={冰岛↦冷，坦纳利佛↦热，新加坡↦潮湿，希腊↦热}

计算下列各题。

1. *likes*(｜{吉姆，爱米丽}｜)
2. (*holiday* ⨾ *climate*)⊳{冷}
3. *holiday*⊕{爱米丽↦希腊}

练习 9.43　如何使用练习 9.42 中所定义的函数来描述到自己所喜欢的气候的国家度假的人的集合？

练习 9.44　假设有以下定义的集合。

$A=\{(1, a), (2, b), (3, c), (4, d)\}$

$B=\{(2, b), (4, b)\}$

$C=\{x : A \mid x.1=1 \vee x.1=3 \cdot (x.1, c)\}$

计算下列各题。

1. $A\oplus B$
2. $A\oplus C$
3. $A\oplus(C\oplus B)$
4. $(A\oplus C)\oplus B$

练习 9.45　假设有集合 $A=\{1, 2\}$，$B=\{x, y\}$。

1. 给出从 A 到 B 的所有全函数的集合。
2. 给出从 A 到 B 的所有单射的集合。
3. 给出从 A 到 B 的所有全单射的集合。
4. 给出从 A 到 B 的所有满射的集合。
5. 给出从 A 到 B 的所有双射的集合。

练习 9.46　考虑如下定义的关系 *parent*(见练习 8.57)。

parent={p，q : *Person* ｜ p 是 q 的父母}

使用 *parent* 定义下列函数。

1. *children*：*Person* → $\mathbb{P}$ *Person*，使得 *children*(p)表示 p 的孩子的集合。

2. *number_of_children*：*Person* → $\mathbb{N}$，使得 *number_of_children*(p)表示 p 的孩子

的数目。

练习 9.47 考虑如下定义的集合。

$People=\{$迈克尔，多米尼克，爱米丽$\}$

$Fruit=\{$苹果，香蕉$\}$

下列函数中哪些是单射，哪些是满射，哪些是双射？

1. $\varnothing$

2. $\{$迈克尔$\mapsto$苹果，多米尼克$\mapsto$苹果$\}$

3. $\{$爱米丽$\mapsto$苹果，多米尼克$\mapsto$香蕉，爱米丽$\mapsto$香蕉$\}$

4. $\{$爱米丽$\mapsto$苹果，多米尼克$\mapsto$香蕉$\}$

练习 9.48 斐波纳契序列(Fibonacci sequence)是如下所示的数列。

0，1，1，2，3，5，8，13，21，34，55，89，144，…

给出生成这一序列的递归函数。

练习 9.49 阿克曼函数(Ackermann function)是如下定义的函数。

$A(0, n)=n+1$

$A(m, 0)=A(m-1, 1)$，若 $m>0$

$A(m, n)=A(m-1, A(m, n-1))$，若 $m>0$ 且 $n>0$

计算下列各题。

1. $A(0, 2)$

2. $A(2, 0)$

9.8 练习解答

9.1 1、2 和 3 是函数，4 不是。

9.2 从$\{1, 2\}$到$\{a, b\}$的所有函数的集合如下所示。

$\varnothing$，$\{1\mapsto a\}$，$\{1\mapsto b\}$，$\{2\mapsto a\}$ $\{2\mapsto b\}$，$\{1\mapsto a, 2\mapsto a\}$，$\{1\mapsto a, 2\mapsto b\}$，$\{1\mapsto b, 2\mapsto a\}$，$\{1\mapsto b, 2\mapsto b\}$

9.3 不总是。考虑如下定义的函数 f。

$f=\{0\mapsto 1, 1\mapsto 1\}$

在此，如下所示的 $f^{\sim}$ 显然不是函数。

$f^{\sim}=\{1\mapsto 0, 1\mapsto 1\}$

9.4 是的。如果 f 是函数，那么源头中的每一个元素最多映射到目标中的一个元素。如果 g 是 f 的一个子集，那么$\subseteq$的性质确保 g 同样满足函数所需要的这一必要的性质。

9.5

1. 是的。f 和 g 都是函数。因为 $f\cap g$ 既是 f 的子集又是 g 的子集，而且函数的任意子集也一定是函数。因此，$f\cap g$ 一定是函数。

2. 不总是。考虑 $f=\{0\mapsto 0\}$ 和 $g=\{0\mapsto 1\}$。这里，f 和 g 都是函数，但是 $f\cup g$ 不是函数。

3. 是的。f 是一个函数。因为 $f\setminus g$ 是 f 的子集，而且函数的任意子集也是函数。因此，$f\setminus g$ 一定是函数。

9.6 3 是全函数，1 和 2 是部分函数，而 4 不是函数。

9.7 从$\{1, 2\}$到$\{a, b\}$的所有全函数如下所示。

$\{1\mapsto a, 2\mapsto a\}$，$\{1\mapsto a, 2\mapsto b\}$，$\{1\mapsto b, 2\mapsto a\}$，$\{1\mapsto b, 2\mapsto b\}$

9.8 不一定是。假设有全函数 $f: X\rightarrow X$ 使得 $X=\{0, 1\}$且 $f=\{0\mapsto 1, 1\mapsto 1\}$。那么，如下所示的 $f^{\sim}$ 显然不是函数。

$f^{\sim}=\{1\mapsto 0, 1\mapsto 1\}$

9.9 不一定是。考虑如下定义的全函数 f 和 g。

$f=\{0\mapsto 0, 1\mapsto 1\}$

$g=\{0\mapsto 1, 1\mapsto 0\}$

在此，$f\cap g=\varnothing$，显然不是全函数。

9.10 1、3、5 和 6 有定义。

9.11

1. 沙鼠

2. 猫

3. 爱米丽

4. 未定义

9.12

1. $\{$爱米丽$\mapsto$金鱼，理查德$\mapsto$沙鼠，迈克尔$\mapsto$猫$\}$

2. {爱米丽，理查德，迈克尔}

3. {爱米丽↦金鱼}

4. {爱米丽}

5. {金鱼}

6. {金鱼}

9.13

1. {琼↦咖啡，戴夫↦咖啡，史蒂夫↦茶}

2. {琼↦咖啡，戴夫↦橙汁，史蒂夫↦茶}

3. {琼↦橙汁，戴夫↦咖啡，史蒂夫↦咖啡}

4. {琼↦橙汁，戴夫↦咖啡，史蒂夫↦咖啡}

5. {琼↦橙汁，戴夫↦咖啡，史蒂夫↦咖啡}

9.14

1. $f \oplus f = (\mathrm{dom}\ f \lhd f) \cup f$
$= \varnothing \cup f$
$= f$

2. $f \oplus \varnothing = (\mathrm{dom}\ \varnothing \lhd f) \cup \varnothing$
$= (\varnothing \lhd f) \cup \varnothing$
$= f \cup \varnothing$
$= f$

3. $\varnothing \oplus f = (\mathrm{dom}\ f \lhd \varnothing) \cup f$
$= \varnothing \cup f$
$= f$

9.15 1 和 4 是单射函数，2 不是函数，3 是函数但不是单射函数。

9.16 从{0，1}到{a，b}的所有单射的集合如下所示。

$\{\varnothing, \{0 \mapsto a\}, \{0 \mapsto b\}, \{1 \mapsto a\}, \{1 \mapsto b\}, \{0 \mapsto a, 1 \mapsto b\}, \{0 \mapsto b, 1 \mapsto a\}\}$

9.17 从{0，1}到{a，b}的所有全单射的集合如下所示。

$\{\{0 \mapsto a, 1 \mapsto b\}, \{0 \mapsto b, 1 \mapsto a\}\}$

9.18 是的。这样的函数既没有汇聚的箭头，也没有分岔的箭头。因此，它的逆也没有这样的箭头。

9.19 不是。考虑函数 $f \in \{0, 1\} \rightarrowtail \{a, b, c\}$，定义如下。

$f = \{0 \mapsto a, 1 \mapsto b\}$

如下所示，f 的逆$\{a \mapsto 0, b \mapsto 1\}$是单射函数，但不是全单射函数。

9.20 只有 2 和 3 是满射函数，4 不是函数，1 是函数但不是满射函数。

9.21 从{0，1}到{a，b}的所有满射的集合如下所示。

$\{\{0 \mapsto a, 1 \mapsto b\}, \{0 \mapsto b, 1 \mapsto a\}\}$

9.22 从{0，1}到{a，b}的所有全满射的集合如下所示。

$\{\{0 \mapsto a, 1 \mapsto b\}, \{0 \mapsto b, 1 \mapsto a\}\}$

9.23 为了使 f 是满射，必须有 $m \geqslant n$。

9.24 是的。由于 X 和 Y 具有相同的势，X 的所有元素都被映射出去，而 Y 中的所有元素都被映射到。因此，X 中的每一个元素一定映射到 Y 中的唯一元素。

9.25 不一定是。当源头的势严格大于目标的势时，满射的逆不是满射。

9.26 是的。如果 f 是一个全单射，那么源头中的每一个元素都被映射出去，而且没有汇集和分岔的箭头。因此，$f^{\sim}$ 的目标中的每一个元素都被映射到，而且 $f^{\sim}$ 没有分岔的箭头。所以，$f^{\sim}$ 是满射。

9.27 1 和 2 是单射但不是满射，3 是满射但不是单射，4 是双射。

9.28 从{0，1}到{a，b}的所有双射的集合如下所示。

$\{\{0 \mapsto a, 1 \mapsto b\}, \{0 \mapsto b, 1 \mapsto a\}\}$

9.29 如果 f 是全双射，那么一定有 $m=n$。

9.30 不一定是。考虑如下定义的 $f \in \{a, b, c\} \rightarrow \{0, 1\}$。

$f = \{a \mapsto 0, b \mapsto 1\}$

这一函数既是单射又是满射，因此是双射。然而，$f^{\sim}$ 不是满射，进而不是双射。

9.31 是的。考虑全双射 f。根据定义，f 没有汇聚的箭头，也没有分岔的箭头，而且源头的所有元素都被映射出去，目标的所有

元素都被映射到。因此，f 也有这些性质。所以，$f^{\sim}$ 是全双射。

9.32 它不是良定义的递归函数。因为递归的情况是自变量离开基底，而不是靠近基底。考虑 $f(4)$ 的计算，有

$$\begin{aligned} f(4) &= f(5)+1 \\ &= f(6)+1+1 \\ &\vdots \end{aligned}$$

9.33 不是良定义的递归函数，因为函数的递归调用可能达不到它的基底。考虑 $f(4)$ 的计算，有

$$\begin{aligned} f(4) &= f(2)+1 \\ &= f(0)+1+1 \\ &= f(-2)+1+1+1 \\ &\vdots \end{aligned}$$

9.34

$zero_count(n)=1$，若 $n=0$

$zero_count(n)=0$，若 $n>0 \wedge n \leqslant 9$

$zero_count(n)=1+zero_count(n \text{ div } 10)$，若 $n \geqslant 10 \wedge n \bmod 10=0$

$zero_count(n)=zero_count(n \text{ div } 10)$，若 $n \geqslant 10 \wedge n \bmod 10 \neq 0$

9.35

$digit_add(n)=n$，若 $n<10$

$digit_add(n)=n \bmod 10+digit_add(n \text{ div } 10)$，其他

9.36

$digit_mult(n)=n$，若 $n<10$

$digit_mult(n)=n \bmod 10 \times digit_mult(n \text{ div } 10)$，其他

9.37

$largest(n)=n$，若 $n<10$

$largest(n)=greater(n \bmod 10, largest(n \text{ div } 10))$，其他

9.38

$subjects: Student \to \mathbb{F}\ Subject$

$tutees: Teacher \to \mathbb{F}\ Student$

9.39

1. 6

2. 11

3. 未定义。因为参数不是英文短语。

9.40 1 和 3 是函数；2 不是函数，因为它含有分岔的箭头。

9.41 全函数把源头中的每一个元素恰好映射到目标中的一个元素，而部分函数不一定满足这一性质。例如，定义在自然数上的平方函数是全函数，而定义在自然数上的除函数则是部分函数。

9.42

1. {{冷}，∅}

2. {爱米丽↦热，戴夫↦潮湿，阿里↦热，贝基↦热}

3. {吉姆↦冰岛，爱米丽↦希腊，戴夫↦新加坡，史蒂夫↦冰岛，阿里↦希腊，贝基↦希腊}

9.43

$\{n: \text{dom } holiday \mid holiday \mathbin{⨾} weather(\lvert\{n\}\rvert) \subseteq likes\ n\}$

9.44

1. $\{(1, a), (2, b), (3, c), (4, b)\}$
2. $\{(1, c), (2, b), (3, c), (4, d)\}$
3. $\{(1, c), (2, b), (3, c), (4, b)\}$
4. $\{(1, c), (2, b), (3, c), (4, b)\}$

9.45

1. $\{\{(1, x), (2, x)\}, \{(1, x), (2, y)\}, \{(1, y), (2, x)\}, \{(1, y), (2, y)\}\}$
2. $\{\varnothing, \{(1, x)\}, \{(1, y)\}, \{(2, x)\}, \{(2, y)\}, \{(1, x), (2, y)\}, \{(1, y), (2, x)\}\}$
3. $\{\{(1, x), (2, y)\}, \{(1, y), (2, x)\}\}$
4. $\{\{(1, x), (2, y)\}, \{(1, y), (2, x)\}\}$
5. $\{\{(1, x), (2, y)\}, \{(1, y), (2, x)\}\}$

9.46

1. $children=\{p: Person; q: \mathbb{F}\ Person$

$\mid (\forall x : q \cdot (p, x) \in parent) \cdot (p, q)\}$[○]

2. $number\_of\_children = \{p : Person \cdot (p, \# children\ p)\}$

9.47 1 是单射，4 是双射。3 不是函数，2 是函数但是既不是单射也不是满射。

9.48

$F(0) = 0$

$F(1) = 1$

$F(n) = F(n-1) + F(n-2)$，若 $n > 1$

9.49

1. $A(0, 2) = 2 + 1$
$= 3$

2. $A(2, 0) = A(1, 1)$
$= A(0, A(1, 0))$
$= A(1, 0) + 1$
$= A(0, 1) + 1$
$= 1 + 1 + 1$
$= 3$

○ 由于类型不能是空集，而 q 有可能是空集(当 p 没有孩子时)，因此这一函数的定义不严密。更严格的定义是：

$children = \{p : Person; q : \mathbb{P}\ Person \mid (\forall x : Person \cdot x \in q \Rightarrow (p, x)\ parent) \cdot (p, q)\}$

译者注

第10章 序　　列

在第4章，我们介绍了集合论。在那里，把集合定义为对象的无序汇集：在集合论的范畴，没有顺序和重复的概念。

本章介绍更具结构性的表示信息的手法。在这种表示信息的方法中，不仅可以表示多次出现的元素，而且元素出现的次序也很重要。因此，重复和顺序都有其用武之地。

在介绍称为*序列*(sequence)的结构之前，我们讨论一个不考虑顺序但却很重要的结构。

10.1 元包

集合是既不注重顺序又不注重重复的结构。例如，我们已经看到，下面的所有集合都彼此相等。

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

然而，有时，特别是当我们模拟现实实体时，需要考虑元素的重复度。

例如，考虑以下集合。

pets={狗，猫，仓鼠}

从这一集合的元素可以得知，问题所涉及的人有三个宠物：一条狗、一只猫和一只仓鼠。然而，假设这个人收养了第二只猫，我们可能按下面的方式更新宠物的集合。

pets_now={狗，猫，仓鼠}∪{猫}

可是，由于集合中没有重复的概念，所以有

pets_now={狗，猫，仓鼠}

如果使用集合来进行表述，就没有办法确定一个人拥有几只猫：只知道他至少拥有一只猫、一条狗和一只仓鼠。

如果元素出现的数目对我们很重要，那么就需要使用另一种结构来表示信息；在此，适宜的结构是*元包*(bag)。然而，如果重复度不重要，那么集合完全符合要求。

首先，考虑表示元包的标记方法。

把元包的元素置于方括号“⟦”和“⟧”之间。因此，可以使用元包表示宠物集合如下：

⟦狗，猫，仓鼠，猫⟧

就像使用运算符∪、∩和\对集合对进行操作那样，对元包也有类似的运算符。

设有元包 $B=⟦a, a, b, c⟧$和 $C=⟦b, b, c, d⟧$，下面介绍作用于 B 和 C 上的元包运算符。

与集合的并运算符一样，元包并运算符合并两个相关元包的元素。与集合的并不同的是，元包并保留重复的元素。因此，有

$B\cup C=⟦a, a, b, b, b, c, c, d⟧$

元包 B 和 C 的交给出下面的结果。

$B\cap C=⟦b, c⟧$

这里，b 的第一个出现同时出现于两个元包，而 b 的第二个出现只出现于 C 中。因此，b 的一个拷贝出现于交中。另外，c 在 B 和 C 中各出现一次。因此，c 在交中出现。

最后，B 和 C 的差以及 C 和 B 的差由下式给出。

$B\setminus C=⟦a, a⟧$

$C\setminus B=⟦b, d⟧$

这里，a 在 B 中出现两次，在 C 中不出现。d 在 C 中出现，在 B 中不出现。而 b 在 C 中出现两次，在 B 中只出现一次。

练习 10.1 下面元包中哪些是相同的？

1. $⟦a, a, b, b⟧$
2. $⟦a, b, a, b⟧$
3. $⟦a, b, b, a⟧$ □

练习 10.2 当用集合表示元包时，下面各元包所对应的集合是什么？

1. $[\![\ ]\!]$
2. $[\![1, 2]\!]$
3. $[\![1, 1, 2, 2]\!]$
4. $[\![\{1, 2\}, \{2\}]\!]$ □

练习 10.3 设有以下元包。

$B=[\![2, 3, 4]\!]$

$C=[\![2, 2]\!]$

$D=[\![3, 3, 4]\!]$

计算下列各题。

1. $C \cup D$
2. $B \cap D$
3. $C \setminus B$
4. $C \cap (B \cup D)$
5. $(C \cap B) \cup D$ □

10.2 顺序的需要

正如我们所看到的，有时会涉及到元素的重复；对于这种情况，元包可能满足建模的需要。然而，在有些情况下，既要考虑元素的重复，又要考虑元素的顺序。例如，需要跟踪需求资源的过程的实时调度程序。调度算法确定下一步执行哪个作业，当相关的资源可用时，将那些资源分配给该作业。用包含那些作业的集合或元包来建模这样的实体是有问题的，这是由于我们的表示没有顺序的概念，调度算法进行的确定作业运行顺序的艰难工作将会失败。因此，需要更复杂的表示这一信息的方法：需要*序列*(sequence)。

我们将序列的元素置于尖括号“〈”和“〉”之间。

例 10.1 序列$\langle a, b\rangle$有两个元素：a 和 b。由于序列的元素是有序的，所以可以进一步说 a 在序列的位置 1，b 在序列的位置 2。顺序很重要，这一序列与$\langle b, a\rangle$不同。而且，重复也很重要，序列$\langle a, b, b\rangle$与$\langle a, b\rangle$是完全不同的对象，当然，它与$\langle b, a\rangle$也不同。 □

练习 10.4 下列序列中哪些是相同的？

1. $\langle a, a, b, b\rangle$
2. $\langle a, b, a, b\rangle$
3. $\langle a, b, b, a\rangle$ □

练习 10.5 当用集合表示下列序列时，其结果如何？

1. $\langle\rangle$
2. $\langle 1, 2\rangle$
3. $\langle 1, 1, 2, 2\rangle$
4. $\langle\{1, 2\}, \{2\}\rangle$ □

我们用 seq X 表示类型为 X 的所有序列的集合。

例 10.2 自然数的所有序列的集合表示为 seq $\mathbb{N}$，$\langle 1, 2, 3\rangle$是 seq $\mathbb{N}$ 的一个元素。因此，可以记作$\langle 1, 2, 3\rangle \in \text{seq}\ \mathbb{N}$。

□

练习 10.6 下列序列属于什么集合？

1. $\langle 7, 11, 13, 17, 19, 23\rangle$
2. $\langle (7, 11), (13, 17), (19, 23)\rangle$
3. $\langle \{7, 11\}, \{13, 17\}, \{19, 23\}\rangle$
4. $\langle \{(7, 11), (13, 17)\}, \{(19, 23)\}\rangle$

□

10.3 建模序列

就像函数是特殊的关系，而关系是特殊的集合一样，序列也可以看成是特殊的函数。

考虑一级方程式赛车，对此我们希望跟踪前三名赛车手。如果舒马赫是第一名，那么我们希望将他与值 1 相关联；如果伊万是第二名，那么我们希望将他与值 2 相关联；如果哈肯宁是第三名，那么我们希望将他与值 3 相关联。对于序列的记法，给出如下表示。

〈舒马赫，伊万，哈肯宁〉

另一个解决方案是定义一个函数 *position*，它将(表示名次的)自然数映射到赛车手。这样，假定存在类型 *Driver*，可能有

$position: \mathbb{I} \rightarrow Driver$

对于上述的情景，有

$position = \{1 \mapsto$舒马赫，$2 \mapsto$伊万，$3 \mapsto$哈肯宁$\}$

这一表示提供的信息实际上与序列所提供的信息相同：对于这两种形式，第一名都与舒马赫相关联，第二名都与伊万相关联，第三名都与哈肯宁相关联。

因此，可以将类型为 X 的序列看成以自然数的开始于 1 的某个连续子序列为定义域的部分函数。也就是说，

$\mathrm{seq}\ X \subseteq \mathbb{I} \setminus \{0\} \rightarrow X$

当然，由于序列是函数，所以可以作用于函数的所有合法运算符同样可以作用于序列。

练习 10.7　使用集合描述给出 seq X 的形式定义。 □

练习 10.8　下面的函数中哪些也是序列？

1. $\varnothing$
2. $\{a \mapsto 1, b \mapsto 2, c \mapsto 3\}$
3. $\{0 \mapsto a, 1 \mapsto b, 2 \mapsto c\}$
4. $\{1 \mapsto a, 3 \mapsto b, 4 \mapsto c\}$
5. $\{1 \mapsto a, 2 \mapsto b, 3 \mapsto c\}$ □

练习 10.9　计算下列各题。

1. dom〈伦敦，爱丁堡，加的夫，贝尔法斯特〉
2. ran〈伦敦，爱丁堡，加的夫，贝尔法斯特〉
3. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉⊳{加的夫，贝尔法斯特}
4. {3，4}⊲〈伦敦，爱丁堡，加的夫，贝尔法斯特〉
5. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉⊕〈曼彻斯特，格拉斯哥〉
6. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉(⌈{1，3}⌉)
7. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉
8. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉3 □

10.4　空序列

正如空集是一个特殊集合一样，空序列也是一个特殊的序列；用〈〉表示空序列。这是不含元素的序列。

例 10.3　考虑下面的函数。

$id\_seq(n) = \{m : 1..n \bullet m \mapsto m\}$

若 $n=3$，那么

$id\_seq(n) = \langle 1, 2, 3 \rangle$

进一步，若 $n=0$，那么

$id\_seq(n) = \langle\rangle$ □

10.5　长度

我们使用集合定义了势运算符 #。给定集合 S，$\#S$ 表示出现在 S 中的元素的数目。因为序列是用函数来定义的，而函数本身又是序偶的集合，因此势运算符同样适用于序列。然而，对于序列，我们称 # 为长度(length)运算符。给定序列 s，$\#s$ 表示 s 的长度。

例 10.4　空序列的长度是 0。 □

规则 10.1

$\#\langle\rangle = 0$ □

例 10.5

$\#\langle a, b, c \rangle = 3$ □

练习 10.10　下列序列的长度是多少？

1. $\langle a, b, b, a \rangle$
2. $id\_seq(10)$
3. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉⊕〈曼彻斯特，格拉斯哥〉 □

练习 10.11　证明：对于任意的序列 s，有

$\#id\_seq(\#s) = \#s$ □

练习 10.12　下列各式中哪些总为真？

1. $\#s = \#(\mathrm{dom}\ s)$

2. $\#s = \#(\mathrm{ran}\ s)$ □

10.6 连接

到目前为止，我们讨论了如何表示序列以及序列的长度。本书介绍结合两个序列的方法。

利用连接(concatenation)运算符⌒可以将一个序列加到另一个序列的后面，产生一个按原有的顺序包含这两个序列的元素的序列。给定两个序列 s 和 t，s 和 t 的连接记作 $s \frown t$。

例 10.6

〈舒马赫，伊万，哈肯宁〉⌒〈巴顿，库尔特哈德〉=〈舒马赫，伊万，哈肯宁，巴顿，库尔特哈德〉 □

例 10.7

〈伦敦，爱丁堡〉⌒〈加的夫，贝尔法斯特〉=〈伦敦，爱丁堡，加的夫，贝尔法斯特〉 □

使用函数的概念，两个序列 s 和 t 的连接有如下形式定义。

$$s \frown t = s \cup \{x : (\#s+1)..(\#s+\#t) \bullet x \mapsto t(x-\#s)\}$$

例 10.8 若 $s=\langle a, b\rangle$，$t=\langle c, d\rangle$，那么，上面的定义给出

$$s \frown t = s \cup \{x : 3..4 \bullet x \mapsto \langle c, d\rangle(x-2)\}$$

当然，这等价于$\langle a, b, c, d\rangle$。 □

存在一些与连接相关的规则。首先，$s \frown t$ 的长度由 s 和 t 的长度决定。

规则 10.2 对于任意的序列 s 和 t，有

$$\#(s \frown t) = (\#s) + (\#t)$$ □

其次，空序列与任何序列 s 相连接，其结果都等于 s。

规则 10.3 对于任意的序列 s，有

$$\langle\rangle \frown s = s$$

及

$$s \frown \langle\rangle = s$$ □

最后，连接运算符满足结合律。

规则 10.4 对于任意的序列 s、t 和 u，有

$$s \frown (t \frown u) = (s \frown t) \frown u$$ □

练习 10.13 下列哪些是⌒的合法运用？

1. $\{1 \mapsto a, 2 \mapsto b\} \frown \{\}$
2. $\{1 \mapsto a, 2 \mapsto b\} \frown \langle\rangle$
3. $\{1 \mapsto a, 2 \mapsto b\} \frown \{1 \mapsto c\}$
4. $\{1 \mapsto a, 2 \mapsto b\} \frown \{1 \mapsto 1\}$
5. $\{1 \mapsto a, 2 \mapsto b\} \frown \{3 \mapsto c\}$ □

练习 10.14 以下陈述中哪些总为真？

1. $s \frown t = t \frown s$
2. $s \frown s = s$
3. $s \frown (t \frown u) = (s \frown t) \frown (s \frown u)$ □

练习 10.15 计算下列各题。

1. $\langle a, b\rangle \frown \langle c, d\rangle$
2. $\langle a, b\rangle \frown \langle\rangle$
3. $\langle a, b\rangle \frown (\langle c, d\rangle \frown \langle e, f\rangle)$
4. $\#(id\_seq(300) \frown id\_seq(200))$
5. $\#(\langle\rangle \frown id\_seq(200))$
6. $\#((id\_seq(300) \frown id\_seq(200)) \setminus id\_seq(100))$ □

练习 10.16 计算下列各题。

1. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉⌒〈曼彻斯特，格拉斯哥〉
2. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉⊕〈曼彻斯特，格拉斯哥〉
3. 〈伦敦，爱丁堡，加的夫，贝尔法斯特〉∪〈曼彻斯特，格拉斯哥〉 □

10.7 头和尾

当我们使用集合存储信息遇到麻烦而考虑利用序列来存储信息时，自然会尝试并利

用序列的信息有序这一事实。在许多利用序列存储信息的应用中，经常需要求序列的第一个元素。在实时调度程序的例子中，如果将作业按其要运行的次序来存储，那么序列头部的元素就是下一次要执行的作业；这显然是重要而且唯一的一个元素。我们称序列的第一个元素为该序列的头(head)。给定任意非空序列 s，s 的头记作 $head\ s$。

例 10.9　给定序列 $\langle a, b, c\rangle$，有

$head\langle a, b, c\rangle = a$　□

另外，还有一个头运算符的补充运算符：尾运算符(tail operator)。它表示序列中跟随在头后面的部分。给定任意非空序列 s，s 的尾记作 $tail\ s$。

例 10.10　给定序列 $\langle a, b, c\rangle$，有

$tail\langle a, b, c\rangle = \langle b, c\rangle$　□

作为进一步的例子，有

$tail\ tail\langle a, b, c\rangle = \langle c\rangle$

以及

$head\ tail\langle a, b, c\rangle = b$

注意，头运算符和尾运算符只对非空序列有定义。将两个运算符运用于 $\langle\rangle$ 的结果都是未定义的。

当然，考虑到序列是用函数来定义的，对于非空序列 s，可以如下定义 $head\ s$。

$head\ s = s\ 1$

同样，可以如下定义尾运算符。

$tail\ s = \{n : 1..\#(s-1) \bullet n \mapsto s(n+1)\}$

例 10.11

$tail\langle a, b, c\rangle = \{n : 1..2 \bullet n \mapsto \langle a, b, c\rangle(n+1)\}$　□

注意，对于类型为 X 的序列，头运算的类型是 $\mathrm{seq}\ X \rightarrow X$，尾运算的类型是 $\mathrm{seq}\ X \rightarrow \mathrm{seq}\ X$：前者返回一个元素，而后者返回一个序列。

关于头和尾的第一个规则表明：每个序列都等价于它的头和尾的连接[○]。

规则 10.5　对于任意的非空序列 s，有

$s = \langle head\ s\rangle \frown \langle tail\ s\rangle$　□

例 10.12

$\langle a, b, b, a\rangle = \langle head\langle a, b, b, a\rangle\rangle \frown (tail\langle a, b, b, a\rangle)$　□

第二个规则表明，每个非空序列的长度都比它的尾的长度大 1。

规则 10.6　对于任意的非空序列 s，有

$\#s = 1 + (\#\ tail\ s)$　□

例 10.13

$\#\langle a, b, b, a\rangle = 1 + (\#\ tail\langle a, b, b, a\rangle)$　□

练习 10.17　下列各式中哪些是有定义的？

1. $head\langle\rangle$
2. $tail\langle a\rangle$
3. $head\ tail\langle a\rangle$
4. $head\ id\_seq(0)$　□

练习 10.18　计算下列各题。

1. $head\ tail\ id\_seq(3)$
2. $tail\ tail\ tail\ id\_seq(3)$
3. $head\langle id\_seq(0)\rangle$
4. $head\langle id\_seq(1), id\_seq(2), id\_seq(3)\rangle$
5. $tail\langle head\ id\_seq(1), head\ id\_seq(2), head\ id\_seq(3)\rangle$
6. $\#\ tail\ id\_seq(100)$
7. $\#(tail\ id\_seq(100) \frown tail\ id\_seq(100))$　□

10.8　限制

给定序列 s，我们也许希望只考虑其元素的某个特定的子集，利用限制运算符(restriction operator)可以做到这一点。给

○ 严格地说，应该是"每个非空序列都等价于以它的头为元素的序列和它的尾的连接"。——译者注

定序列 s，以及元素的一个集合 X，$s \upharpoonright X$ 表示 s 到出现于 X 中的元素的限制。这一运算的结果是这样一个序列，只有同时出现于 s 和 X 的元素出现于该序列中，而且这些元素的顺序由它们在 s 中的顺序决定。

例 10.14 考虑如下给出的序列。

〈雷切尔，邓肯，约翰，理查德，爱米丽，安迪〉

这里，序列中有六个元素，其中雷切尔出现在序列的头。这一序列到集合{爱米丽，雷切尔}的限制如下所示。

〈雷切尔，邓肯，约翰，理查德，爱米丽，安迪〉↾{爱米丽，雷切尔}

这一运算给出〈雷切尔，爱米丽〉的结果。□

可以如下递归地定义这一运算符。

$$\langle\rangle \upharpoonright X = \langle\rangle$$

$$(\langle x\rangle \frown s) \upharpoonright X = \langle x\rangle \frown (s \upharpoonright X), 若\ x \in X$$

$$(\langle x\rangle \frown s) \upharpoonright X = s \upharpoonright X, 若\ x \notin X$$

这一定义对上面例子的应用如下所示。

〈雷切尔，邓肯，约翰，理查德，
爱米丽，安迪〉↾{爱米丽，雷切尔}
=〈雷切尔〉⌢(〈邓肯，约翰，理查德，
爱米丽，安迪〉↾{爱米丽，雷切尔})
=〈雷切尔〉⌢(〈约翰，理查德，
爱米丽，安迪〉↾{爱米丽，雷切尔})
=〈雷切尔〉⌢(〈理查德，爱米丽，
安迪〉↾{爱米丽，雷切尔})
=〈雷切尔〉⌢(〈爱米丽，安迪〉↾
{爱米丽，雷切尔})
=〈雷切尔〉⌢〈爱米丽〉⌢(〈安迪〉↾
{爱米丽，雷切尔})
=〈雷切尔〉⌢〈爱米丽〉⌢(〈〉↾
{爱米丽，雷切尔})
=〈雷切尔〉⌢〈爱米丽〉⌢〈〉
=〈雷切尔，爱米丽〉

与这一运算符相关的第一条规则表明，将空序列限制到任意集合的结果都是空序列。

规则 10.7 对于任意的集合 X，有

$\langle\rangle \upharpoonright X = \langle\rangle$ □

例 10.15

〈〉↾{爱米丽，雷切尔}=〈〉 □

下一条规则表明，任意序列限制到空集的结果都是空序列。

规则 10.8 对于任意的序列 s，有

$s \upharpoonright \varnothing = \langle\rangle$ □

例 10.16

〈雷切尔，邓肯，约翰，理查德，爱米丽，安迪〉↾∅=〈〉 □

最后，↾对⌢满足分配律。

规则 10.9 对于任意的序列 s，$t \in seq\ X$，以及任意的 $S \in \mathbb{F} X$，有

$(s \frown t) \upharpoonright S = (s \upharpoonright S) \frown (t \upharpoonright S)$ □

例 10.17 可以证明，下式成立。

$(\langle a, b, c\rangle \frown \langle b, c, d\rangle) \upharpoonright \{b\} = (\langle a, b, c\rangle \upharpoonright \{b\}) \frown (\langle b, c, d\rangle \upharpoonright \{b\})$

这里，可以对等式的左边做如下简化。

$(\langle a, b, c\rangle \frown \langle b, c, d\rangle) \upharpoonright \{b\}$
$= \langle a, b, c, b, c, d\rangle \upharpoonright \{b\}$
$= \langle b, b\rangle$

同样，可以对等式的右边做如下简化。

$(\langle a, b, c\rangle \upharpoonright \{b\}) \frown (\langle b, c, d\rangle \upharpoonright \{b\})$
$= \langle b\rangle \frown \langle b\rangle$
$= \langle b, b\rangle$

因此，两边相同。 □

练习 10.19 计算下列各题。

1. 〈茶，咖啡，茶，茶，咖啡〉↾{茶}
2. 〈茶，咖啡，茶，茶，咖啡〉↾{咖啡}
3. 〈茶，咖啡，茶，茶，咖啡〉↾{茶，咖啡}
4. 〈茶，咖啡，茶，茶，咖啡〉↾∅ □

练习 10.20 计算下列各题。

1. $head(\langle a, b, c\rangle \upharpoonright \{b, c\})$
2. $tail(\langle a, b, c\rangle \upharpoonright \{b, c\})$
3. $tail(tail(\langle a, b, c\rangle \upharpoonright \{b, c\}))$ □

练习 10.21　设 $A \in seq(People \times People)$ 为如下所示的序列。

$A = \langle$(吉姆，琼)，(戴夫，邓肯)，(爱米丽，伊丽莎白)$\rangle$

计算下列各题。

1. $A \upharpoonright \{p : A \mid p.1 = 吉姆\}$

2. $A \upharpoonright \{p : tail\ A \mid p.1 = 吉姆\}$

3. $A \upharpoonright \{p, q : Person \mid (p, q) \in A \wedge p = 爱米丽\}$

4. $A \upharpoonright \{p : \{吉姆, 戴夫\}; q : Person \mid (p, q) \in A\}$ □

练习 10.22　考虑练习 10.21 中的集合 A，叙述如何通过集合描述从 A 生成下列序列。

1. $\langle$吉姆，戴夫，爱米丽$\rangle$

2. $\langle$吉姆，邓肯$\rangle$

3. $\langle$戴夫，爱米丽$\rangle$ □

10.9　逆置

给定序列 s，s 的逆置(reverse)还是序列，但是，元素出现的顺序被颠倒了，也就是说，s 的第一个元素是 s 的逆置的最后元素，s 的第二个元素是 s 的逆置的倒数第二个元素，以此类推。给定序列 s，用 $reverse\ s$ 表示 s 的逆置。

例 10.18　序列 $\langle a, b, c\rangle$ 的逆置是 $\langle c, b, a\rangle$。 □

例 10.19

$reverse\langle 1, 2, 3, 4, 5\rangle = \langle 5, 4, 3, 2, 1\rangle$ □

使用递归定义函数，可以如下形式地定义 $reverse$。

$$reverse\langle\rangle = \langle\rangle$$

$$reverse\langle x\rangle = \langle x\rangle$$

$$reverse\langle x\rangle \frown s = (reverse\ s) \frown \langle x\rangle$$

例 10.20　可以如下计算 $reverse\langle a, b, c\rangle$。

$$\begin{aligned} reverse\langle a, b, c\rangle &= (reverse\langle b, c\rangle) \frown \langle a\rangle \\ &= ((reverse\langle c\rangle) \frown \langle b\rangle) \frown \langle a\rangle \\ &= ((\langle c\rangle) \frown \langle b\rangle) \frown \langle a\rangle \\ &= \langle c, b, a\rangle \end{aligned}$$ □

关于逆置的第一条规则表明，序列 s 的逆置的逆置是 s 本身。

规则 10.10　给定任意的序列 s，有

$reverse\ reverse\ s = s$ □

例 10.21

$reverse\ reverse\langle 1, 2, 3\rangle = \langle 1, 2, 3\rangle$ □

下一条规则表明，如果我们希望逆置序列并对序列进行限制，那么执行这些运算的顺序是无关紧要的。

规则 10.11　给定任意的序列 $s \in seq\ X$，以及任意的集合 $S \in \mathbb{F} X$，有

$(reverse\ s) \upharpoonright S = reverse(s \upharpoonright S)$ □

例 10.22　可以证明

$$\begin{aligned} &(reverse\langle a, b, c\rangle) \upharpoonright \{a, b\} \\ &= reverse(\langle a, b, c\rangle \upharpoonright \{a, b\}) \end{aligned}$$

成立。

等式的左边可以如下计算。

$$\begin{aligned} &(reverse\langle a, b, c\rangle) \upharpoonright \{a, b\} \\ &= \langle c, b, a\rangle \upharpoonright \{a, b\} \\ &= \langle b, a\rangle \end{aligned}$$

同样，右边可以如下计算。

$$\begin{aligned} &reverse(\langle a, b, c\rangle \upharpoonright \{a, b\}) \\ &= reverse\langle a, b\rangle \\ &= \langle b, a\rangle \end{aligned}$$

因此，两边相等。 □

最后一条规则表明，s 的逆置的长度与 s 的长度相同。

规则 10.12　给定任意的序列 s，有

$\#\ reverse\ s = \#\ s$ □

例 10.23

$\#\langle a, b, c\rangle = \#\langle c, b, a\rangle$ □

练习 10.23　证明逆置对 $\frown$ 不满足分配律。 □

练习 10.24　计算下列各题。

1. *reverse id_seq*(3)

2. *reverse tail id_seq*(3)

3. *head reverse id_seq*(3)

4. *reverse*(*id_seq*(10) ↾ {n : ℕ | n is odd})

5. (*reverse id_seq*(3))⌢(*reverse id_seq*(3)) □

10.10 单射序列

回忆一下，我们在第 9 章将单射函数定义为这样的函数：对于目标中的每一个元素，最多有一个源头中的元素映射到它。因此，目标中的每一个元素在值域中只出现一次。例如，下面的 i 是一个单射函数。

$i=\{a\mapsto 1,\ b\mapsto 2,\ c\mapsto 3,\ d\mapsto 4\}$

这里，值域中不存在重复出现的元素。

反之，下面的函数 j 不是单射函数。

$j=\{a\mapsto 1,\ b\mapsto 2,\ c\mapsto 1,\ d\mapsto 2\}$

这里，1 和 2 都在值域中重复出现。

由于序列是特殊的函数，所以同样可以考虑单射序列(injective seguence)的概念。给定序列 s，s 是单射序列当且仅当在 s 中没有重复出现的元素。用 *iseq X* 表示类型为 X 的所有单射序列组成的集合，*iseq X* 的形式定义如下。

$$\mathrm{iseq}\ X=\{s : \mathrm{seq}\ X \mid (\forall x,\ y : \mathrm{dom}\ s \mid x\neq y \bullet sx\neq sy)\}$$[①]

等价地，

$$\mathrm{iseq}\ X=(\mathrm{seq}\ X)\cap(\mathbb{N}\setminus\{0\}\rightarrowtail X)$$

例 10.24 ⟨a，b，c，d⟩是一个单射序列：它包含 4 个都只出现一次的元素。另一方面，⟨a，b，b，a⟩不是单射序列：它包含两个元素，它们各出现两次。 □

例 10.25 对于任意的 $n\in\mathbb{N}$，*id_seq*(n)是单射序列。 □

练习 10.25 下列序列中哪些是单射序列？

1. ⟨⟩

2. *id_seq*(3)

3. *id_seq*(3)⌢*id_seq*(5)

4. *id_seq*(3)⌢(*tail tail tail id_seq*(5)) □

练习 10.26 证明：当且仅当 $\#s=\#\mathrm{ran}\ s$ 时序列 s 是单射序列。 □

10.11 再论递归函数

前一章，我们给出了递归函数的概念。在那里，给出了如下所示的递归函数的一般形式。

$f(n)=x$，若 $n=m$

$\quad=y+f(n')$，其他

另外还介绍了如果递归函数 f 满足下列性质，那么它是良定义的。

1. 存在称为基值的某些值，对于这些值 f 不引用其自身。

2. 每当 f 引用其自身时，函数的自变量都更靠近基值。

序列的结构表明，我们希望定义在序列上的很多函数都可以递归地定义。例如，考虑如下给出的 *reverse* 运算符的定义。

$$reverse\langle\rangle=\langle\rangle$$

$$reverse\langle x\rangle=\langle x\rangle$$

$$reverse\langle x\rangle\frown s=(reverse\ s)\frown\langle x\rangle$$

这里，*reverse* 不仅能够用其自身来定义，而且可以看到有两个基底：空序列和单序列，在这两种情况下，递归终止。另外，这一递归定义将其自变量推向两个基底。因此，*reverse* 不仅是递归函数，而且是良定义的。

① 更严密的定义应为 $\mathrm{iseq}\ X=\{s : \mathrm{seq}\ X \mid (\forall x,\ y\in\mathbb{N} \mid x\in\mathrm{dom}\ s\wedge y\in\mathrm{dom}\ s\wedge x\neq y \bullet sx\neq sy)\}$，这一定义可以处理 $\mathrm{dom}\ s=\varnothing$ 的情况。——译者注

作为下一个例子，我们如下递归地定义长度运算符 $\#$。

$\#\langle\rangle=0$

$\#\langle x\rangle\frown s=1+\#s$

这一函数也是良定义的。

练习 10.27　定义类型为 $\mathrm{seq}\,\mathbb{N}\to\mathbb{N}$ 的递归函数，它返回自然数序列中的最大自然数。假定这一函数在 $\langle\rangle$ 的值为 0。 □

练习 10.28　说明如何利用 μ 运算符来非递归地定义练习 10.27 中的函数。 □

练习 10.29　定义类型为 $\mathrm{seq}\,\mathbb{N}\to\mathrm{seq}\,\mathbb{N}$ 的递归函数，对于给定的自然数序列 s，它返回将 s 中的每一个元素加倍的自然数序列。例如，把这一函数应用于 $\langle 1, 2, 3\rangle$ 时，给出结果 $\langle 2, 4, 6\rangle$。 □

练习 10.30　说明如何利用集合描述来非递归地定义练习 10.29 中的函数。 □

10.12　附加练习

练习 10.31　假设序列 $s\in seq\ X$ 有 $\#s=n$。下列序列有多少个元素？

1. $\langle\rangle$
2. $s\frown s$
3. $tail\ s$
4. $reverse\ s$
5. $s\upharpoonright\varnothing$
6. $s\upharpoonright X$
7. $head\langle s, s\rangle$
8. $tail\langle s, s\rangle$

练习 10.32　下列各项的值是什么？

1. $\#\langle\langle\rangle, \langle m, a, d\rangle, \langle a, m\rangle\rangle$
2. $\mathrm{ran}\langle m, a, d, a, m\rangle$
3. $\mathrm{dom}\langle m, a, d, a, m\rangle$
4. $\langle m, a, d, a, m\rangle\oplus\langle a, d, a, m\rangle$

练习 10.33　$square$ 和 $double$ 的定义如下所示。

$square=\langle 1, 4, 9, 16\rangle$

$double=\langle 2, 4, 6, 8\rangle$

计算下列各题。

1. $square\frown double$
2. $(square\frown double)\upharpoonright\{1, 3, 5, 7, 9\}$
3. $head((square\frown double)\upharpoonright\{1, 3, 5, 7, 9\})$
4. $tail((square\frown double)\upharpoonright\{1, 3, 5, 7, 9\})$
5. $\#((square\frown double)\upharpoonright\{1, 3, 5, 7, 9\})$
6. $reverse((square\frown double)\upharpoonright\{1, 3, 5, 7, 9\})$

练习 10.34　考虑如下给出的序列。

$r=\langle$雅典，巴塞罗那，科隆，迪拜$\rangle$

$s=\langle$雅典，亚特兰大，巴塞罗那$\rangle$

$t=\langle$亚特兰大，波士顿，
　　芝加哥，丹佛$\rangle$

计算下列各题。

1. $r\cap s$
2. $r\cup s$
3. $r\setminus s$
4. $\mathrm{dom}\ t$
5. $\mathrm{ran}\ s$
6. $r^{\frown}$
7. $\{1, 2\}\lhd r$
8. $s\rhd\{$亚特兰大$\}$
9. $(t\oplus s)(\mid\{1, 2\}\mid)$
10. $r\ 3$

练习 10.35　假设给定练习 10.34 中的序列，计算下列各题。

1. $r\frown s$
2. $\#r$
3. $reverse\ s$
4. $r\upharpoonright\mathrm{ran}\ s$
5. $(r\frown s)\upharpoonright\{$亚特兰大，雅典$\}$
6. $head\ t$
7. $tail\ t$

8. *reverse tail t*

9. *head tail t*

10. $\langle head\ t\rangle \frown \langle head\ r\rangle$

练习 10.36 定义一个谓词 $P(s)$，当且仅当序列 s 不含重复元素时它等值于 true。

练习 10.37 定义一个谓词 $P(s, x)$，当且仅当元素 x 出现于 s 时它等值于 true。

练习 10.38 定义一个谓词 $P(s, x, n)$，当且仅当元素 x 出现于 s 的第 n 个位置时它等值于 true。

练习 10.39 定义一个类型为 $X \times \text{seq}\ X \to \mathbb{N}$ 的函数，对于给定的序列 $s \in \text{seq}\ X$ 和元素 $x \in X$，该函数返回出现于 s 中的所有 x 的下标。

练习 10.40 定义一个类型为 $X \times \text{seq}\ X \to \mathbb{N}$ 的函数，对于给定的序列 $s \in \text{seq}\ X$ 和元素 $x \in X$，该函数返回第一次出现于 s 中的 x 的下标。

练习 10.41 定义一个类型为 $X \times \text{seq}\ X \to \mathbb{N}$ 的函数，对于给定的序列 $s \in \text{seq}\ X$ 和元素 $x \in X$，该函数返回最后一次出现于 s 中的 x 的下标。

练习 10.42 定义一个类型为 $X \times \text{seq}\ X \to \mathbb{N}$ 的递归函数，对于给定的序列 $s \in \text{seq}\ X$ 和元素 $x \in X$，该函数返回 x 在 s 中出现的次数。

练习 10.43 给出上面练习所述函数的非递归定义。

练习 10.44 定义一个类型为 $X \times \text{seq}\ X \times \mathbb{N} \to \mathbb{N}$ 的函数，对于给定的序列 $s \in \text{seq}\ X$，元素 $x \in X$ 以及自然数 n，该函数返回第 n 次出现于 s 中的 x 的下标。

练习 10.45 定义一个递归函数 *repeated*，对任意的序列 s，该函数返回重复出现于 s 中的元素的数目。

练习 10.46 考虑有如下运算结果的函数 *flatten*。

$$flatten\langle\langle a, b, c\rangle, \langle\rangle, \langle d, e\rangle\rangle = \langle a, b, c, d, e\rangle$$

而且

$$flatten\langle\langle a\rangle, \langle b\rangle, \langle c\rangle\rangle = \langle a, b, c\rangle$$

给出 *flatten* 的递归定义。

练习 10.47 小汉普顿火车站有 4 个站台。每个站台有一个屏幕，屏幕向乘客通告从该站台出站的下几辆列车。这一信息用函数 *display* 表示，*display* 的类型如下所示。

$$display: Platform \to \text{seq}(Time \times Destination)$$

例如，如果第 2 站台的屏幕目前显示开往诺里奇的 12.20 列车和开往布里斯托尔的 13.30 列车将从该站台出发，那么有

$$display\ p2 = \langle(12.20, \text{诺里奇}), (13.30, \text{布里斯托尔})\rangle$$

1. 写出一个谓词，该谓词表述：出现于所有屏幕上的所有序列都是按时间排列的，最早开出的列车排在序列的开头。

2. 写出一个谓词，该谓词表述：所有屏幕含有唯一的信息：即任意的 $(time, destination)$ 都不能出现于两个以上的屏幕。

3. 给出定义包含当前显示的所有目的地的集合的集合描述。

4. 给出定义包含所有开往斯旺西的列车的开车时间和相关的站台的集合的集合描述。

5. 定义一个函数，该函数把给定 $(time, destination)$ 添加到给定站台的屏幕的最后。

6. 定义一个函数，该函数从给定站台的屏幕删除给定的 $(time, destination)$。

练习 10.48 飞机场的航班表显示如下信息：起飞时间，航班，目的地，登机通道，状态。航班表的示例如下所示。

起飞时间	航班	目的地	登机通道	状态
10.30	BA11	新加坡	29	登机
10.40	KL27	阿姆斯特丹	27	延期
10.45	QA09	墨尔本	13	登机

可以如下形式地描述这样的航班表。

b: seq(起飞时间 × 航班 × 目的地 × 登

机通道×状态)

因此，对于上面的示例，有

$b=\langle(10.30, BA11, 新加坡, 29, 登机), (10.40, KL27, 阿姆斯特丹, 27, 延期), (10.45, QA09, 墨尔本, 13, 登机)\rangle$

另外，假设 $Time$ 上存在一个顺序 $<$，而且有一个变量 $current : Time$，该变量表示目前的时间。因此，假如目前的时间是 10.37，有 $10.30 < current$ 及 $current < 10.40$。

1. 写出一个谓词，它表述：b 中出现的所有多元组都按起飞时间排列。

2. 写出一个谓词，它表述：只有延期的飞机和起飞时间比目前时间晚的飞机出现于 b。

3. 写出一个谓词，它表述：如果航班为 BA11 的飞机出现于 b，那么它一定是飞往新加坡的。

4. 给出生成所有延期的飞机的详细信息的集合描述。

5. 给出生成 b 中所有飞机的目的地的集合描述。

6. 给出生成 b 中所有飞机的(目的地，起飞时间)序偶的集合描述。

7. 给出生成所有延期飞机的(目的地，起飞时间)序偶的集合描述。

10.13 练习解答

10.1 它们都相等。

10.2

1. $\varnothing$
2. $\{1, 2\}$
3. $\{1, 2\}$
4. $\{\{1, 2\}, \{2\}\}$

10.3

1. $[\![2, 2, 3, 3, 4]\!]$
2. $[\![3, 4]\!]$
3. $[\![2]\!]$
4. $[\![2]\!]$
5. $[\![2, 3, 3, 4]\!]$

10.4 它们都不同。

10.5

1. $\varnothing$
2. $\{1, 2\}$
3. $\{1, 2\}$
4. $\{\{1, 2\}, \{2\}\}$

10.6

1. $\mathrm{seq}\ \mathbb{N}$
2. $\mathrm{seq}\ (\mathbb{N} \times \mathbb{N})$
3. $\mathrm{seq}\ \mathbb{F}\mathbb{N}$
4. $\mathrm{seq}\ \mathbb{F}\ (\mathbb{N} \times \mathbb{N})$

10.7

$\mathrm{seq}\ X=\{s : \mathbb{N} \setminus \{0\} \rightarrow X \mid (\exists n : \mathbb{N} \bullet \mathrm{dom}\ s=1..n)\}$

10.8 只有 1 和 5 还是序列。其他不满足序列的条件：2 的类型不正确，3 的定义域包含 0，4 在定义域有“缝隙”。

10.9

1. $\{1, 2, 3, 4\}$
2. {伦敦，爱丁堡，加的夫，贝尔法斯特}
3. {伦敦，爱丁堡}
4. {3 ↦加的夫，4 ↦贝尔法斯特}
5. {曼彻斯特，格拉斯哥，加的夫，贝尔法斯特}
6. {伦敦，加的夫}
7. {伦敦↦1，爱丁堡↦2，加的夫↦3，贝尔法斯特↦4}
8. 加的夫

10.10

1. 4
2. 10
3. 4

10.11 给定序列 s，s 的长度定义为 $\#s$。$id_seq(\#s)$的定义域为自然数 $1..\#s$ 的一个连续子序列。因此，这一序列的长度是 $\#s$。

10.12 1 总是真的，2 不总是真的。例如，

考虑序列$\langle a, a\rangle$。这里，$\#\langle a, a\rangle=2$，而$\#(\mathrm{ran}\langle a, a\rangle)=1$。

10.13 1、2 和 3 是⌒的合法运用，4 不是⌒的合法运用，因为序列的类型不同，5 不是⌒的合法运用，因为第二个自变量不是序列。

10.14 所有陈述通常都不为真。

10.15

1. $\langle a, b, c, d\rangle$
2. $\langle a, b\rangle$
3. $\langle a, b, c, d, e, f\rangle$
4. 500
5. 200
6. 400

10.16

1. {伦敦，爱丁堡，加的夫，贝尔法斯特，曼彻斯特，格拉斯哥}
2. ⟨曼彻斯特，格拉斯哥，加的夫，贝尔法斯特⟩
3. {1 ↦伦敦，2 ↦爱丁堡，3 ↦加的夫，4 ↦贝尔法斯特，1 ↦曼彻斯特，2 ↦格拉斯哥}

10.17 只有 2 有定义。

10.18

1. 2
2. ⟨⟩
3. ⟨⟩
4. ⟨1⟩
5. ⟨1，1⟩
6. 99
7. 198

10.19

1. ⟨茶，茶，茶⟩
2. ⟨咖啡，咖啡⟩
3. ⟨茶，咖啡，茶，茶，咖啡⟩
4. ⟨⟩

10.20

1. b
2. $\langle c\rangle$
3. ⟨⟩

10.21

1. ⟨(吉姆，琼)⟩
2. ⟨⟩
3. ⟨(爱米丽，伊丽莎白)⟩
4. ⟨(吉姆，琼)，(戴夫，邓肯)⟩

10.22

1. $\{n : \mathrm{dom}\ A \cdot n \mapsto (A n).1\}$
2. $\{n : \mathrm{dom}\ A \mid n<3 \cdot n \mapsto (A n).n\}$
3. $\{n : \mathrm{dom}\ tail\ A \cdot n \mapsto (A n).1\}$

10.23 为使 *reverse* 对⌒满足分配律，下面的式子必须成立。

$$reverse(s \frown t)=(reverse\ s) \frown (reverse\ t)$$

考虑 $s=\langle a, b\rangle$和 $t=\langle c, d\rangle$，有

$$reverse(\langle a, b\rangle \frown \langle c, d\rangle)=\langle d, c, b, a\rangle$$

且

$$(reverse\langle a, b\rangle) \frown (reverse\langle c, d\rangle)$$
$$=\langle b, a, d, c\rangle$$

因此，*reverse* 对⌒不满足分配律。

10.24

1. ⟨3，2，1⟩
2. ⟨3，2⟩
3. 3
4. ⟨9，7，5，3，1⟩
5. ⟨3，2，1，3，2，1⟩

10.25 1、2 和 4 是单射序列，3 不是。

10.26 根据定义，s 是单射序列当且仅当 s 的每一个元素都刚好出现一次。因此，$\#\mathrm{dom}\ s=\#\mathrm{ran}\ s$。我们已经看到，对于所有的序列 s，都有$\#s=\#\mathrm{dom}\ s$。由相等的传递性可知，若 s 是单射序列则一定有$\#s=\#\mathrm{ran}\ s$。

10.27

$$f(\langle\rangle)=0$$

$$f(\langle x\rangle \frown s)=\max(x, f(s))$$

其中，$\max(x, y)$是如下定义在自然数 x 和 y 上的函数。

$$\max(x, y)=x\ ,\ 若\ x>y$$
$$=y\ ,\ 其他$$

10.28 这一函数可以如下非递归地定义。

$f(s)=(\mu n : \mathrm{dom}\ s \mid (\forall m : \mathrm{dom}\ s \cdot (s\,n > s\,m) \vee (sn = sm \wedge n < m)) \cdot sn)$○

10.29

$$f(\langle\rangle)=\langle\rangle$$

$$f(\langle x\rangle \frown s)=\langle x\times 2\rangle \frown f(s)$$

10.30 这一函数也可如下定义。

$f(s)=\{n : \mathrm{dom}\ s \cdot n \mapsto 2\times(sn)\}$○

10.31

1. 0
2. $2\times n$
3. $n-1$
4. n
5. 0
6. n
7. n
8. 1

10.32

1. 3
2. $\{m, a, d\}$
3. $\{1, 2, 3, 4, 5\}$
4. $\langle a, d, a, m, m\rangle$

10.33

1. $\langle 1, 4, 9, 16, 2, 4, 6, 8\rangle$
2. $\langle 1, 9\rangle$
3. 1
4. $\langle 9\rangle$
5. 2
6. $\langle 9, 1\rangle$

10.34

1. 〈雅典〉
2. {1↦雅典，2↦巴塞罗那，2↦亚特兰大，3↦科隆，3↦巴塞罗那，4↦迪拜}
3. {2↦巴塞罗那，3↦科隆，4↦迪拜}
4. {1，2，3，4}
5. {雅典，亚特兰大，巴塞罗那}
6. {雅典↦1，巴塞罗那↦2，科隆↦3，迪拜↦4}
7. 〈雅典，巴塞罗那〉
8. {1↦雅典，3↦巴塞罗那}
9. {雅典，亚特兰大}
10. 科隆

10.35

1. 〈雅典，巴塞罗那，科隆，迪拜，雅典，亚特兰大，巴塞罗那〉
2. 4
3. 〈巴塞罗那，亚特兰大，雅典〉
4. 〈雅典，巴塞罗那〉
5. 〈雅典，雅典，亚特兰大〉
6. 亚特兰大
7. 〈波士顿，芝加哥，丹佛〉
8. 〈丹佛，芝加哥，波士顿〉
9. 波士顿
10. 〈亚特兰大，雅典〉

10.36

$P(s) \Leftrightarrow \forall m, n : \mathrm{dom}\ s \cdot m \neq n \Rightarrow s\,m \neq s\,n$○

10.37

$\forall s : \mathrm{seq}\ X;\ x : X \cdot P(s, x) \Leftrightarrow x \in \mathrm{ran}\ s$

10.38

$\forall s : \mathrm{seq}\ X;\ x : X;\ n : \mathbb{N} \cdot P(s, x, n) \Leftrightarrow sn = x$

10.39

$\forall x : X;\ s : \mathrm{seq}\ X \cdot f(x, s) = \{i : \mathrm{dom}\ s \mid si = x\}$⊗

10.40

$\forall x : X;\ s : \mathrm{seq}\ X \cdot f(x, s) = (\mu n : \mathbb{N} \mid sn = x \wedge (\forall m : \mathbb{N} \mid sm = x \cdot n \leqslant m))$

○○ 这一定义对 $s=\langle\rangle$ 的情况不正确，请读者自行加以修改。——译者注

○⊗ 该谓词对 $s=\langle\rangle$ 不正确，请读者自行修改。——译者注

10.41

$\forall x : X;\ s : \mathrm{seq}\ X \bullet f(x,\ s) = (\mu n : \mathbb{N} \mid sn = x \wedge (\forall m : \mathbb{N} \mid sm = x \bullet n \geqslant m))$

10.42

$f(x, \langle\rangle) = 0$

$f(x, \langle y\rangle \frown s) = 1 + f(x, s)$，若 $x = y$

$f(x, \langle y\rangle \frown s) = f(x, s)$，其他

10.43

$\forall x : X;\ s : \mathrm{seq}\ X \bullet f(x,\ s) = \#\{n : \mathbb{N} \mid sn = x\}$

10.44

$\forall x : X;\ s : \mathrm{seq}\ X;\ n : \mathbb{N} \bullet f(x,\ s,\ n) = (\mu m : \mathbb{N} \mid sm = x \wedge \#\{l : \mathbb{N} \mid sl = x \wedge l \leqslant m\} = n)$

10.45

$repeated\langle\rangle = 0$

$repeated\langle x\rangle \frown s = 1 + repeated(s \upharpoonright \mathrm{ran}\ s \setminus \{x\})$，若 $x \in \mathrm{ran}\ s$

$repeated\langle x\rangle \frown s = repeated\ s$，其他

10.46

$flatten\langle\rangle = \langle\rangle$

$flatten\langle x\rangle \frown s = x \frown flatten\ s$

10.47○

1. $\forall s : \mathrm{ran}\ display \bullet \forall p,\ q : \mathrm{dom}\ s \mid p < q \bullet p.1 \leqslant q.1$

2. $\forall s,\ t : \mathrm{ran}\ display \bullet \mathrm{ran}\ s \cap \mathrm{ran}\ t = \varnothing$

3. $\bigcup\{s : \mathrm{ran}\ display \bullet \{p : \mathrm{ran}\ s \bullet p.2\}\}$

4. $\{p : Platform;\ s : \mathrm{seq}(Time \times Destination) \mid (p,\ s) \in display \wedge (\exists x : \mathrm{ran}\ s \bullet x.2 = swansea) \bullet x.1 \mapsto p\}$

5. $\forall d : Platform \rightarrow \mathrm{seq}(Time \times Destination);\ p : Platform;\ q : Time \times Destination \bullet add(d,\ p,\ q) = d \oplus \{p \mapsto (dp) \frown \langle q\rangle\}$

6. $\forall d : Platform \rightarrow \mathrm{seq}(Time \times Destination);\ p : Platform;\ q : Time \times Destination \bullet remove(d,\ p,\ q) = d \oplus \{p \mapsto (dp) \upharpoonright \{q\}\}$

10.48

1. $\forall s,\ t : \mathrm{dom}\ b \bullet s < t \Leftrightarrow (b\,s).1 \leqslant (b\,t).1$

2. $\forall s : \mathrm{ran}\ b \bullet s.5 = 延期 \vee s.1 > current$

3. $\forall s : \mathrm{ran}\ b \bullet s.2 = \mathrm{BA11} \Rightarrow s.3 = 新加坡$

4. $\{s : \mathrm{ran}\ b \mid s.5 = 延期\}$

5. $\{s : \mathrm{ran}\ b \bullet s.3\}$

6. $\{s : \mathrm{ran}\ b \bullet (s.3,\ s.1)\}$

7. $\{s : \mathrm{ran}\ b \mid s.5 = 延期 \bullet (s.3,\ s.1)\}$

○ 所有解答都没有考虑空集的特殊情况，例如，(在晚间)可能已经没有从某个站台出站的列车，这时，对于相应的 s，$\mathrm{dom}\ s = \varnothing$。——译者注

第11章 归 纳 法

到此为止，已经讨论了一系列证明定理的方法。在第3章，我们看到，可以通过真值表、替换、等值推理和自然演绎建立命题逻辑的定理。当考虑谓词逻辑时，我们看到，可以将等值推理和自然演绎的技术从命题逻辑拓广到谓词逻辑。另外，我们还学习了如何使用等值推理来证明集合、关系、函数以及序列的相等关系。综上可知，利用等值推理和自然演绎的技术可以推导并证明我们开发的绝大多数形式描述的性质。

另一方面，当希望证明给定类型的所有序列，或者所有自然数满足的性质时，我们需要一种不同的技术，这一技术称为*归纳法*(induction)。归纳法是一个强有力的数学技术，是许多计算应用的核心技术。而且，在第9章和第10章我们已经看到，它与递归密切相关。事实上，归纳法经常是证明某些类型的归纳算法的正确性的唯一方法。

本章讨论两种归纳法：数学归纳法，利用它可以证明所有自然数满足的性质；结构归纳法，利用它可以证明给定归纳定义的类型的所有元素满足的一些性质(例如，序列就是一种归纳定义的类型)。我们首先考虑数学归纳法。

论述数学归纳法相关内容的两本参考书是[You64]和[Wan80]。前者更多讨论的是数学归纳法，而后者更多讨论的是数学归纳法在程序设计中的应用。

11.1 数学归纳法

在第2章，我们讨论了由Giuseppe Peano最早提出的处理自然数的方法——Peano算术。在那里，我们指出，可以使用5个公理来定义自然数。第2章讨论了其中的前4个公理。这些公理如下所示。

1. 0是自然数。

2. 若 x 是自然数，则 $x+1$ 也是自然数。

3. 不存在使 $z+1=0$ 成立的自然数 z。

4. 给定自然数 x 和 y，若 $x+1=y+1$ 则 $x=y$。

下面介绍第5个公理，其叙述如下。

对于某个性质，如果能证明该性质对某个自然数 x 成立则对 $x+1$ 也成立，而且能证明该性质对0成立，那么该性质对所有自然数成立。

也就是说，给定一个性质 p，如果可以证明 $p(0)$ 成立，而且可以证明 $p(n)\Rightarrow p(n+1)$ 对每个自然数 n 都成立，那么可得出 $p(n)$ 对所有自然数成立的结论。可以用自然演绎的格式来如下书写这一公理。

$$\frac{p(0)\quad \forall n:\mathbb{N}\cdot p(n)\Rightarrow p(n+1)}{\forall n:\mathbb{N}\cdot p(n)}\text{[数学归纳法]}$$

因此，这样的性质对于所有自然数成立的证明工作被分为两个阶段：*基底阶段*，在此证明该性质对于0成立；*归纳阶段*，在此证明若该性质对于 n 成立，那么它对于 $n+1$ 也成立。后一阶段的证明可以如下进行：假设该性质对 n 成立，这称为*归纳假设*(inductive hypothesis)，然后使用这一假设来证明该性质对于 $n+1$ 也成立。可以如下形式地给出归纳法原理(principle of mathematical induction)的陈述。

设有性质 p，它对于每个自然数或者为真或者为假。若

1. *$p(0)$ 为真*

2. *$p(n)\Rightarrow p(n+1)$ 为真*

那么，p 对于所有自然数为真。

下面是一个比较简单的例子。

例 11.1 我们希望证明下面的定理成立。

$\forall n: \mathbb{N} \cdot n < n+1$

为证明每个自然数 n 都小于 $n+1$，需要证明基底阶段和归纳阶段。先考虑基底阶段，在此，由<和+的定义，显然有

$0<0+1$

下面考虑归纳阶段。为此，首先假设定理对某个值 x 为真，即

$x<x+1$

成立。这就是归纳假设。由此出发，需要证明

$(x+1)<(x+1)+1$

成立。由算术可知，若 $m<n$ 则 $m+1<n+1$。由于归纳假设告诉我们 $x<x+1$，所以可以用它来得到

$(x+1)<(x+1)+1$

因为我们证明了基底阶段和归纳阶段，所以得到结论

$\forall n: \mathbb{N} \cdot n<n+1$ □

许多离散数学的教科书以多米诺骨牌来解释数学归纳法。考虑一列一个接一个排列起来的多米诺骨牌，如果骨牌的间距合理，那么推倒第一张骨牌会引发第二张骨牌倒下，它又会引发第三张骨牌倒下，以此类推。这样，所有的骨牌都会被推倒。数学归纳法的工作原理则是：若定理对 0 成立，那么它对 1 也成立；若它对 1 成立，那么它对 2 也成立；若它对 2 成立，那么它对 3 也成立；以此类推。归纳法和多米诺骨牌间的这一类似性对上面的例子显然是正确的。

下面再看一个例子。

例 11.2 使用归纳法可以证明下面的公式对于所有的自然数 n 都成立。

$$\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$$

为确信这一公式的正确性，考虑 $n=3$ 的情况。这时有

$$\sum_{i=1}^{3} i = \frac{3 \times 4}{2} = 6$$

再例如，考虑 $n=5$ 的情况，有

$$\sum_{i=1}^{5} i = \frac{5 \times 6}{2} = 15$$

为证明公式对 $n=0$ 成立，有

$$\sum_{i=1}^{0} i = \frac{0 \times 1}{2} = 0$$

显然，这是真的(注意，若 $m<n$，则 $\sum_{i=n}^{m} i = 0$)。

我们尝试通过假设定理对某个值 x 成立来证明归纳阶段。即假设

$$\sum_{i=1}^{x} i = \frac{x \times (x+1)}{2}$$

这是归纳假设。如果可以从 $\sum_{i=1}^{x} i = \frac{x \times (x+1)}{2}$ 得到

$$\sum_{i=1}^{x+1} i = \frac{(x+1) \times ((x+1)+1)}{2}$$

就可以证明归纳阶段。当 $n=x+1$ 时，等式的左边为

$1+2+3+..+x+(x+1)$

由归纳假设，可以把上式改写成

$$\frac{x \times (x+1)}{2} + (x+1)$$

可以如下继续证明。

$$\frac{x \times (x+1)}{2} + (x+1)$$

$$= (x+1) \times \left(\frac{x}{2} + 1\right)$$

$$= \frac{(x+1) \times (x+2)}{2}$$

$$= \frac{(x+1) \times ((x+1)+1)}{2}$$

这样，就完成了归纳阶段的证明。

由于基底阶段和归纳阶段都成立，所以得到

$$\sum_{i=1}^{n} i = \frac{n \times (n+1)}{2}$$

对所有自然数 n 都成立的结论。 □

练习 11.1 使用数学归纳法证明

$\forall n : \mathbb{N} \cdot (n^2 + n)$ 为偶数 □

为真。

11.2 结构归纳法

数学归纳法是证明特定性质对所有自然数都成立的技术。结构归纳法(structural induction)是证明特定性质对某个归纳定义[1]的类型的所有元素都成立的技术。到此为止，我们考虑的归纳定义的对象类型只有序列。为了证明确实可以归纳地定义序列，考虑 seq $\mathbb{N}$(所有自然数序列的集合)的如下定义。

1. $\langle\rangle$ 是 seq $\mathbb{N}$ 的元素。
2. 对于任意的 $s \in \text{seq } \mathbb{N}$ 及任意的 $x \in \mathbb{N}$，$\langle x \rangle \frown s$ 是 seq $\mathbb{N}$ 的元素。

以上足以定义 seq $\mathbb{N}$。

例 11.3 $\langle\rangle$ 是自然数的序列。因为 $\langle\rangle \in \text{seq } \mathbb{N}$，可知 $\langle 0 \rangle \in \text{seq } \mathbb{N}$、$\langle 1 \rangle \in \text{seq } \mathbb{N}$ 及 $\langle 2 \rangle \in \text{seq } \mathbb{N}$，等等。进而，因为 $\langle 0 \rangle \in \text{seq } \mathbb{N}$，可知 $\langle 0, 0 \rangle \in \text{seq } \mathbb{N}$、$\langle 0, 1 \rangle \in \text{seq } \mathbb{N}$ 及 $\langle 0, 2 \rangle \in \text{seq } \mathbb{N}$，等等。因此，自然数的任意序列都可以通过这一过程来定义。 □

练习 11.2 给定集合 Car，给出所有汽车序列的集合的归纳定义。 □

与数学归纳法相同，使用结构归纳法的证明也包括两个子证明。为证明性质 p 对类型为 X 的序列成立，需要证明以下两种情况。

1. 基底阶段：$p(\langle\rangle)$。
2. 归纳阶段：$\forall x : X; s : \text{seq } X \cdot p(s) \Rightarrow p(\langle x \rangle \frown s)$。

即，需要证明 p 对于空序列成立，而且能够从 $p(s)$ 得到 $p(\langle x \rangle \frown s)$。第二个阶段通常包括使用归纳假设 $p(s)$ 为证明 $p(\langle x \rangle \frown s)$ 的假设。

可以使用自然演绎规则的风格如下表示这一证明。

$$\frac{p(\langle\rangle) \quad \forall s : \text{seq } X; x : X \cdot p(s) \Rightarrow p(\langle x \rangle \frown s)}{\forall s : \text{seq } X \cdot p(s)}$$

[结构归纳法]

这里，证明 $p(\langle\rangle)$ 是基底阶段，证明

$\forall s : \text{seq } X; x : X \cdot p(s) \Rightarrow p(\langle x \rangle \frown s)$

是归纳阶段。

对于序列，可以如下形式地陈述结构归纳法原理。

设有性质 p，它对类型 X 的每个序列或者为真或者为假。若

1. $p(\langle\rangle)$ 为真；
2. $\forall s : \text{seq } X; x : X \cdot p(s) \Rightarrow p(\langle x \rangle \frown s)$ 为真。

那么 p 对所有类型 X 的序列为真。

练习 11.3 假设我们希望证明汽车的所有序列的某个性质 p。陈述该证明的基底阶段和归纳阶段。 □

正如我们在第 10 章所看到的，有些序列的性质对所有序列都成立，不论序列是什么类型的。例如，

$$\#(s \frown t) = (\# s) + (\# t)$$

对所有序列 s 和 t 成立，不论它们的类型为 seq $\mathbb{N}$ 还是 seq Car，唯一的条件是 s 和 t 必须是相同类型的。

这样的性质也可用结构归纳法证明。

例 11.4 设有类型 X，使得 $s \in \text{seq } X$ 且 $t \in \text{seq } X$。为了证明

$\forall s, t : \text{seq } X \cdot \#(s \frown t) = (\# s) + (\# t)$

成立，需要证明两个子情况成立。首先，这一性质对空序列成立，即有

---

[1] 更多的参考书使用递归定义(recursively defined)一词。——译者注

$\forall t: \text{seq } X \cdot \#(\langle\rangle \frown t) = (\#\langle\rangle) + (\#t)$

然后，需要证明：若这一性质对某个序列 s 成立，那么它对 $\langle x\rangle \frown s$ 也成立，其中 x 是 X 的任意元素，即

$\forall s, t: \text{seq } X;\ x: X \cdot$

$\#(s \frown t) = (\#s) + (\#t) \Rightarrow \#((\langle x\rangle \frown s) \frown t) = (\#(\langle x\rangle \frown s)) + (\#t)$

成立。

首先考虑基底阶段。在此，有

$\#(\langle\rangle \frown t)$

$= \#t$

$= 0 + \#t$

$= \#\langle\rangle + \#t$

下面考虑归纳阶段。在此，要证明

$\#((\langle x\rangle \frown s) \frown t) = \#(\langle x\rangle \frown s) + (\#t)$

可由 $\#(s \frown t) = (\#s) + (\#t)$ 得到。

首先，考虑

$\#((\langle x\rangle \frown s) \frown t)$

根据 $\frown$ 的结合律，这与下式相同。

$\#(\langle x\rangle \frown (s \frown t))$

由 $\#$ 的定义，这等于

$1 + \#(s \frown t)$

由归纳假设，这等于

$1 + (\#s) + (\#t)$

最后，由 $\#$ 的定义，这等于

$\#(\langle x\rangle \frown s) + (\#t)$ □

练习 11.4 证明：对任意的序列 s，有

$reverse(reverse\ s) = s$ □

成立。

11.3 附加练习

练习 11.5 用数学归纳法证明

$\forall n: \mathbb{N} \mid n \geqslant 1 \cdot 1^2 + 2^2 + 3^2 + .. + n^2 = \dfrac{n \times (n+1) \times (2n+1)}{6}$

为真。注意，这是关于所有大于 0 的自然数的性质。因此，基底阶段是 1 而非 0 的情况。

练习 11.6 用数学归纳法证明

$\forall n: \mathbb{N} \mid n \geqslant 1 \cdot 1 + 4 + 7 + .. + (3n-2) = \dfrac{n \times (3n-1)}{2}$

为真。

练习 11.7 用数学归纳法证明

$\forall n: \mathbb{N} \mid n \geqslant 4 \cdot n! \geqslant 2^n$

为真。注意，n 的阶乘，记作 $n!$，定义为：若 $n > 1$ 则

$n! = n \times (n-1)!$

若 $n = 0$ 或 $n = 1$ 则

$n! = 1$

练习 11.8 用数学归纳法证明

$\forall n: \mathbb{N} \mid n \geqslant 3 \cdot n^2 \geqslant 2n + 1$

为真。

练习 11.9 用结构归纳法证明：对于任意的类型 X，有

$\forall s: \text{seq } X;\ A: \mathbb{P} X \cdot (s \frown t) \upharpoonright A = (s \upharpoonright A) \frown (t \upharpoonright A)$

为真。

练习 11.10 用结构归纳法证明，对于任意的类型 X，有

$\forall s: \text{seq } X;\ A: \mathbb{P} X \cdot reverse(s \upharpoonright A) = (reverse\ s) \upharpoonright A$

为真。

11.4 练习解答

11.1 先进行基底阶段的证明。为此，有

$0^2 + 0 = 0$

我们知道，0 是偶数，因此，此性质对 0 成立。

下面，进行归纳阶段的证明。为此，假设对某个值 x，$x^2 + x$ 是偶数。考虑 $x+1$，有

$$(x+1)^2 + (x+1) = x^2 + 2x + 1 + (x+1) = x^2 + x + 2x + 2$$

由归纳假设，x^2+x 是偶数。而且，对于任意的 x，$2x+2$ 显然是偶数。由于偶数的和总是偶数，所以 $x^2+x+2x+2$ 一定是偶数。

因此，由数学归纳法，有

$$\forall n:\mathbb{N}\cdot n^2+n \text{ 是偶数}$$

为真。

11.2

1. $\langle\rangle$是 seq Car 的元素。

2. 对于任意的 $s\in \text{seq } Car$ 和任意的 $c\in Car$，$\langle c\rangle\frown s$ 是 seq Car 的元素。

11.3 基底阶段由 $p(\langle\rangle)$给出，归纳阶段由下式给出。

$$\forall s:\text{seq } Car;\ c:Car\cdot p(s)\Rightarrow p(\langle c\rangle\frown s)$$

11.4 基底阶段可以如下证明。

$$\begin{aligned}&reverse(reverse\langle\rangle)\\=&\ reverse\langle\rangle && [reverse \text{ 的定义}]\\=&\ \langle\rangle && [reverse \text{ 的定义}]\end{aligned}$$

归纳阶段可以如下证明。

$$\begin{aligned}&reverse(reverse\langle x\rangle\frown s)\\=&\ reverse((reverse\ s)\frown\langle x\rangle)\\&[reverse \text{ 的定义}]\\=&\ \langle x\rangle\frown(reverse\ reverse\ s)\\&[reverse \text{ 的定义}]\\=&\ \langle x\rangle\frown s\quad [\text{归纳假设}]\end{aligned}$$

因此、有

$$reverse(reverse\ s)=s$$

对于所有序列成立。

11.5 先证明基底阶段。在此，有

$$1^2=\frac{1\times2\times3}{6}$$

因此，基底阶段成立。

下面，证明归纳阶段。为此，假设

$$\begin{aligned}&1^2+2^2+3^2+..+x^2\\=&\frac{x\times(x+1)\times(2x+1)}{6}\end{aligned}$$

成立。

考虑 $x+1$，有

$$\begin{aligned}&1^2+2^2+..+x^2+(x+1)^2\\=&\frac{x\times(x+1)\times(2x+1)}{6}+(x+1)^2\\=&\frac{x\times(x+1)\times(2x+1)+6(x+1)^2}{6}\\=&\frac{(x+1)\times((2x^2+x)+(6x+6))}{6}\\=&\frac{(x+1)\times(2x^2+7x+6)}{6}\\=&\frac{(x+1)\times(x+2)\times(2x+3)}{6}\\=&\frac{(x+1)\times(x+2)\times(2(x+1)+1)}{6}\end{aligned}$$

因此，由数学归纳法，有

$$\begin{aligned}&\forall n:\mathbb{N}\cdot 1^2+2^2+3^2+..+n^2\\=&\frac{n\times(n+1)\times(2n+1)}{6}\end{aligned}$$

为真。

11.6 先进行基底阶段的证明。在此，等式的左边等于 1，右边由下式给出。

$$\begin{aligned}\frac{1\times(3-1)}{2}&=\frac{1\times2}{2}\\&=1\end{aligned}$$

因此，等式的两边相等。

然后，进行归纳阶段的证明。为此，假设

$$1+4+7+..+(3x-2)=\frac{x\times(3x-1)}{2}$$

成立。

考虑 $x+1$，有

$$\begin{aligned}&1+4+7+..+(3(x+1)-2)\\=&\frac{x\times(3x-1)}{2}+(3x+1)\\=&\frac{x\times(3x-1)+2\times(3x+1)}{2}\\=&\frac{3x^2+5x+2}{2}\\=&\frac{(x+1)\times(3x+2)}{2}\\=&\frac{(x+1)\times(3(x+1)-1)}{2}\end{aligned}$$

因此，由数学归纳法，有

$$\forall n : \mathbb{N} \mid n \geq 1 \cdot 1+4+7+\ldots+(3n-2) = \frac{n \times (3n-1)}{2}$$

为真。

11.7 首先考虑基底阶段，其中 $n=4$。在此，有

$$4! = 24$$

及

$$2^4 = 16$$

因此，基底阶段成立。

下面考虑归纳阶段。根据归纳假设，有

$$x! \geq 2^x$$

因为 $x+1>0$，有

$$(x+1) \times x! \geq (x+1) \times 2^x$$

由阶乘！的定义，它等价于

$$(x+1)! \geq (x+1) \times 2^x$$

由于 $x+1>2$，可得

$$(x+1)! \geq 2 \times 2^x$$

由此，有

$$(x+1)! \geq 2^{x+1}$$

因此，归纳阶段成立。

由上可知，由数学归纳法，有

$$\forall n : \mathbb{N} \mid n \geq 4 \cdot n! \geq 2^n$$

为真。

11.8 首先考虑 $n=3$ 的基底阶段。在此，有

$$3^2 \geq (2 \times 3)+1$$

由此，基底阶段成立。

下面考虑归纳阶段。首先对 $(x+1)^2$ 做如下展开。

$$(x+1)^2 = x^2+2x+1$$

由归纳假设，有

$$x^2+2x+1 \geq (2x+1)+2x+1$$

由于 $(2x+1)+2x+1=4x+2$，上式可写成

$$x^2+2x+1 \geq 4x+2$$

最后，因为 $x \geq 3$，所以有

$$x^2+2x+1 \geq 2(x+1)+1$$

因此，归纳阶段成立。

由数学归纳法，有

$$\forall n : \mathbb{N} \mid n \geq 3 \cdot n^2 \geq 2n+1$$

成立。

11.9 先考虑基底阶段。在此，有

$$\begin{aligned}(\langle\rangle \frown t) \upharpoonright A &= t \upharpoonright A \\ &= \langle\rangle \frown (t \upharpoonright A) \\ &= (\langle\rangle \upharpoonright A) \frown (t \upharpoonright A)\end{aligned}$$

因此，基底阶段成立。

现在考虑归纳阶段。归纳假设是

$$(s \frown t) \upharpoonright A = (s \upharpoonright A) \frown (t \upharpoonright A)$$

由此，需要证明下式成立。

$$\begin{aligned}&((\langle x\rangle \frown s) \frown t) \upharpoonright A \\ &= ((\langle x\rangle \frown s) \upharpoonright A) \frown (t \upharpoonright A)\end{aligned}$$

为了证明上式成立，需要证明下列两个子情况。

$$\begin{aligned}&((\langle x\rangle \frown s) \frown t) \upharpoonright A \\ &= \langle x\rangle \frown ((s \frown t) \upharpoonright A)，若\ x \in A \\ &= (s \frown t) \upharpoonright A，其他\end{aligned}$$

设 $x \in A$。在此，由归纳假设有

$$\begin{aligned}\langle x\rangle \frown ((s \frown t) \upharpoonright A) &= \langle x\rangle \frown (s \upharpoonright A) \frown (t \upharpoonright A) \\ &= ((\langle x\rangle \frown s) \upharpoonright A) \frown (t \upharpoonright A)\end{aligned}$$

现在设 $x \notin A$。由归纳假设，有

$$\begin{aligned}(s \frown t) \upharpoonright A &= (s \upharpoonright A) \frown (t \upharpoonright A) \\ &= ((\langle x\rangle \frown s) \upharpoonright A) \frown (t \upharpoonright A)\end{aligned}$$

因此，归纳阶段成立。由此，有

$$\begin{aligned}&\forall s : \mathrm{seq}\ X;\ A : \mathbb{F}\ X \cdot (s \frown t) \upharpoonright A \\ &= (s \upharpoonright A) \frown (t \upharpoonright A)\end{aligned}$$

对任意给定的类型 X 成立。

11.10 先考虑基底阶段。在此，有

$$\begin{aligned}reverse(\langle\rangle \upharpoonright A) &= reverse\langle\rangle \\ &= (reverse\langle\rangle) \upharpoonright A\end{aligned}$$

现在考虑归纳阶段。归纳假设为

$$reverse(s \upharpoonright A) = (reverse\ s) \upharpoonright A$$

由此，需要证明

$$\begin{aligned}&reverse((\langle x\rangle \frown s) \upharpoonright A) \\ &= (reverse\langle x\rangle \frown s) \upharpoonright A\end{aligned}$$

它的证明如下。首先，有

$$reverse((\langle x\rangle \frown s) \upharpoonright A)$$

$=reverse(\langle x\rangle \frown (s \upharpoonright A))$，若 $x \in A$

$=reverse(s \upharpoonright A)$，其他

设 $x \in A$。由归纳假设，有

$$
\begin{aligned}
& reverse(\langle x\rangle \frown (s \upharpoonright A)) \\
=& reverse(s \upharpoonright A) \frown \langle x\rangle \\
=& ((reverse\ s) \upharpoonright A) \frown \langle x\rangle \\
=& (reverse\langle x\rangle \frown s) \upharpoonright A
\end{aligned}
$$

现在设 $x \notin A$。再由归纳假设，有

$$
\begin{aligned}
reverse(s \upharpoonright A) &= (reverse\ s) \upharpoonright A \\
&= (reverse\langle x\rangle \frown s) \upharpoonright A
\end{aligned}
$$

因此，归纳阶段成立。因此，有

$$
\begin{aligned}
& \forall s : \mathrm{seq}\ X;\ A : \mathbb{F} X \bullet reverse(s \upharpoonright A) \\
& = (reverse\ s) \upharpoonright A
\end{aligned}
$$

对任意给定的类型 X 为真。

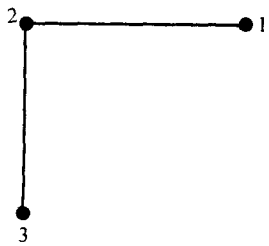
第12章 图 论

在第1章中我们就说过，对计算专业的高校学生进行离散数学的教学的目的主要有两个。其一是讲授支撑形式描述技术的数学理论，如Z(参看[Spi92]或[WD96])或CSP(参看[Hoa85]或[Ros97])。其二是讲授支撑理论计算机科学的数学理论。本章的论题(图论)和下一章的论题(组合数学)都属于第二个目的。在这一章里，将对图论的基本概念给出必要的简单介绍。如果想要了解有关这个主题的更深入的知识，有兴趣的读者可以参考[Wes95]、[Wil96]或[Mer00]。对于图论的附加练习，有兴趣的读者可以参考[Bal97]。

图以及图论是计算机科学的很多领域的核心。例如，将许多计算机连接在一起形成网络，这样的网络的结构可以表示为一个图。再例如，在高级程序设计中经常遇到的基础数据结构——二叉树结构，其实是图的一个特例。从更广义的层面来讲，图可以形成关系的可视化表示。下面，从解释到底什么是图来开始这一章的学习。

12.1 图

考虑下图。

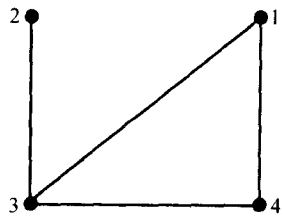


图中含有3个顶点(vertex)(或者称作结点(node)或点(point)，这三种称呼是一样的)：1、2和3。另外，图中还有两条边(edge)(或称作线(line)，这两种称呼也是一样的)，其中，第一条边连接顶点1和顶点2，第二条边连接顶点2和顶点3。注意，每一条边都是从一个顶点出发并且在一个顶点结束。

形式地讲，把图看成是顶点的非空集合和(可能空的)边的集合，并且每条边都是从一个顶点出发且终止于另一个顶点。

称一个顶点与另一个顶点是相邻的(adjacent)，当且仅当这两个顶点之间有边。因此，在上例中，顶点1与顶点2相邻(反之亦然)，顶点2与顶点3相邻(反之亦然)。

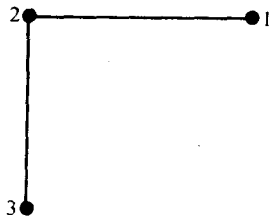
练习 12.1 下图中哪些顶点对是相邻的？



□

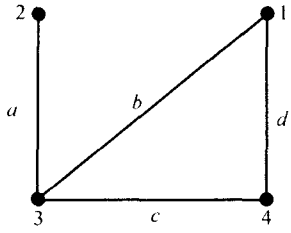
称一条边关联于(incident)两个顶点，当且仅当这两个顶点出现在这条边的两端。

例 12.1 在下面的图中只有两条边，其中一条边关联于顶点1和顶点2，而另一条边关联于顶点2和顶点3。



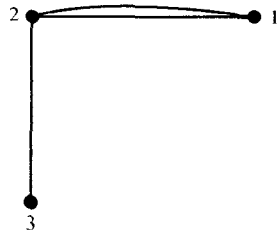
□

可以为边和顶点标上标记。例如，考虑练习12.1中的图，可以把边分别记作 a，b，c，d。其表示如下所示。



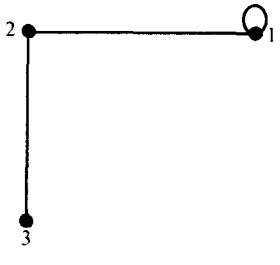
练习 12.2 上图有 4 条边。对于每条边，给出它所关联的顶点。 □

在一个图的一对顶点之间，可以有多条边，例如考虑下面的图。



这里，在顶点 1 和顶点 2 之间有两条平行的边(parallel edge)。这也是完全合法的图。

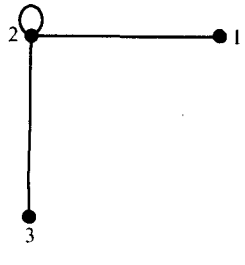
另外，一个图可以包含环(loop)，也就是说，起点和终点都是相同顶点的边。例如，下图在顶点 1 处包含一个环。



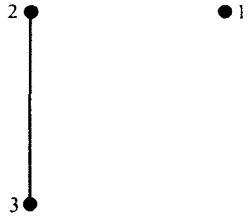
称一个图为简单图(simple graph)，当且仅当它既不含环也不含平行边。

练习 12.3 下列各图中哪些是简单图？

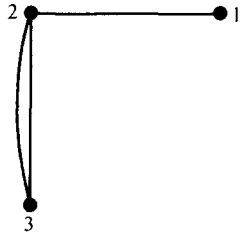
1.



2.



3.



□

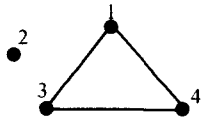
练习 12.4 没有边的图是简单图吗？

□

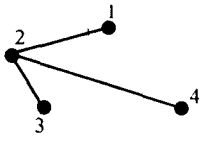
练习 12.5 考虑含有 n 个顶点的图，如果它是简单图，那么最多能包含多少条边？ □

给定一个简单图 G，G 的补(complement)是一个简单图，它与 G 含有相同的顶点，并且任意两个顶点在 G 的补中相邻当且仅当它们在 G 中不相邻。我们把 G 的补记作 $\overline{G}$。

例 12.2 考虑下面的简单图。



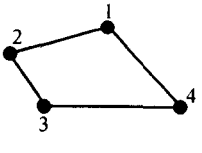
这个图的补如下所示。



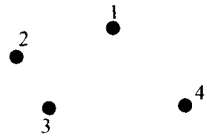
□

练习 12.6 给出下列各简单图的补。

1.



2.

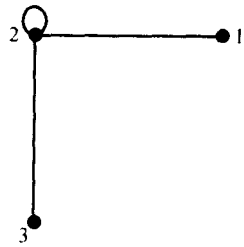


□

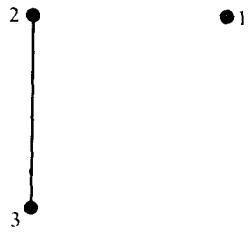
对于图中的任何顶点 v，可以求 v 的度(degree)，记作 $deg(v)$。v 的度是与这一顶点相关联的边的数目。注意，如果顶点 v 与一个环相关联，那么把这个环看成与两条边等价。

例 12.3　再次考虑如下所示的练习 12.3 中的图。

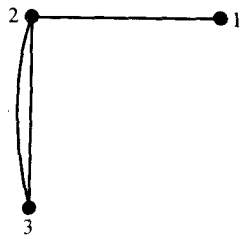
1.



2.



3.



在第一个图中，有

$deg(1)=1$

$deg(2)=4$

$deg(3)=1$

在第二个图中，有

$deg(1)=0$

$deg(2)=1$

$deg(3)=1$

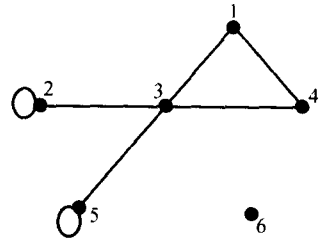
在第三个图中，有

$deg(1)=1$

$deg(2)=3$

$deg(3)=2$　□

练习 12.7　计算下图中各顶点的度。



□

对于任意的顶点 v，如果 $deg(v)=0$，则称 v 为孤立顶点(isolated vertex)。

练习 12.8　在练习 12.7 的图中，哪些顶点是孤立的？　□

读者可能已经注意到，图中所有顶点的度的总和与边数间存在着某种关系，的确，前者是后者的两倍；这一关系形式地表述如下。

$$\left(\sum_{v\in V} deg(v)\right)=2\times \# E$$

其中，V 表示图的顶点集合，E 表示图的边的集合。

练习 12.9　验证下式对练习 12.7 中的图成立。

$$\left(\sum_{v\in V} deg(v)\right)=2\times \# E$$　□

关于顶点的度的第二个定理是握手引理(handshaking lemma)，引理的内容如下。

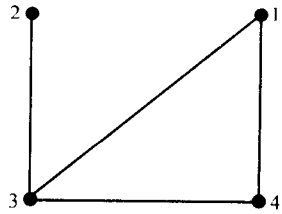
在任何一个图中，度为奇数的顶点的数目是偶数。

练习 12.10　验证练习 12.7 中的图满足握手引理。　□

一个图称为是连通的(connected)，当

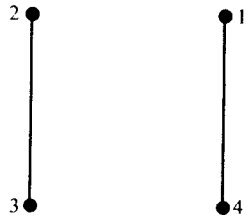
且仅当从任一个顶点出发，如有必要经过一个或多个中间顶点，能够到达其余的任意顶点[1]。

例 12.4 下面的图是一个连通图。



在某种意义上，这个图是“混为一体的”。 □

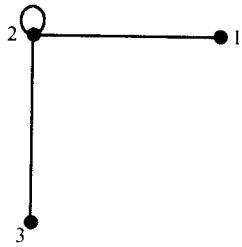
例 12.5 另一方面，下面的图不是连通的：因为无法从顶点 1 到达顶点 2(反之亦然)。



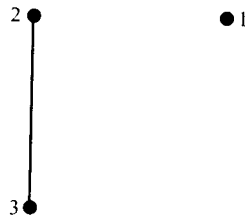
□

练习 12.11 下列各图哪些是连通的？

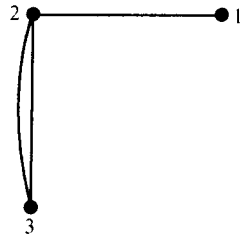
1.



2.



3.



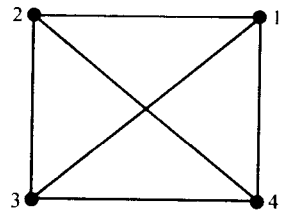
□

练习 12.12 考虑带有 n 个顶点的图。如果这个图是连通的，那么它最少包含多少条边？

□

一个图称为是完全的(complete)，当且仅当每个顶点都与其他顶点相邻。

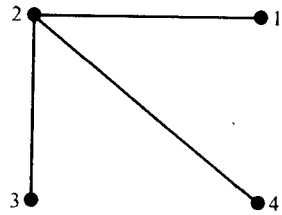
例 12.6 考虑下面的图。



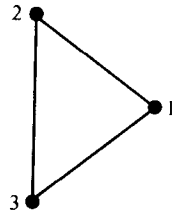
这个图是完全的，因为每一个顶点都可以通过一条边与其他顶点相连。然而，如果从这个图中去掉任何一条边，那么剩下的图就不是完全图了。 □

练习 12.13 下列各图哪些是完全的？

1.

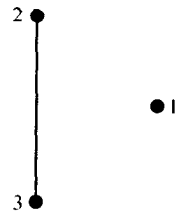


2.



[1] 在本章的后面，将考虑连通性的更加技术性的定义。然而，在此这一定义已经足够了。

3.



□

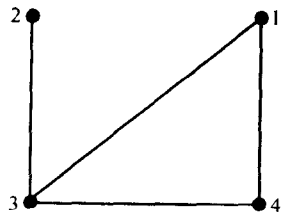
练习 12.14　考虑带有 n 个顶点的图，如果它是完全的，那么最少包含多少条边？ □

练习 12.15　证明：如果一个图是完全的，那么它一定是连通的。 □

12.2　图的集合和元包表示

可以使用集合和元包(bag)形式地表示图。回忆一下，图是由一些顶点和边组成的，而且边连接着顶点，可以使用顶点的非空集合和边(顶点的序偶)的(可能空的)元包来表示图。把顶点的集合记作 V，边的元包记作 E。给定一个图 G，其顶点的集合为 V，边的元包为 E，可以写作 $G=(V, E)$。

再一次考虑下图。



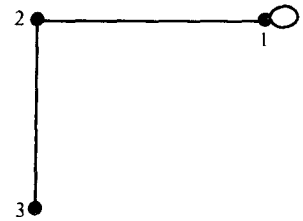
可以按如下方式表示这个图。

$V=\{1, 2, 3, 4\}$

$E=[\![(1, 3), (1, 4), (2, 3), (3, 4)]\!]$

值得注意的是，对于顶点的集合 V，除了 V 不能为空集外，对 V 的大小没有任何限制。另外，对于出现于边的元包 E 中的顶点对，除了它们必须是类型为 $V\times V$ 的元素之外，没有其他限制。特别地，在 E 中可以存在形如(v, v)的元素，表示图中包含环(loop)。

例 12.7



这个图可以如下形式化地表示。

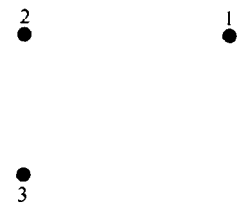
$V=\{1, 2, 3\}$

$E=[\![(1, 1), (1, 2), (2, 3)]\!]$

注意，E 中的顶点对$(2, 3)$表明顶点 2 与顶点 3 相邻，这可以等价地表示成$(3, 2)$。这是因为，到目前为止，所讨论的图仅仅是无向图。同时还要注意的是，图中出现的每条边只记录一次。 □

元包 E 可以是空的。

例 12.8　考虑下图。



这个图可以形式地表示如下。

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

$E=[\![\ \ ]\!]$ □

练习 12.16　为什么使用元包而不是使用集合来表示 E？ □

练习 12.17　画出下列各图。

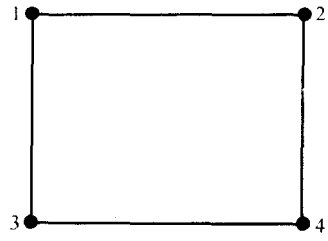
1. $V=\{1, 2, 3, 4\}$，$E=[\![\ \ ]\!]$

2. $V=\{1, 2, 3, 4\}$，$E=[\![(1, 1), (2, 2), (3, 3)]\!]$

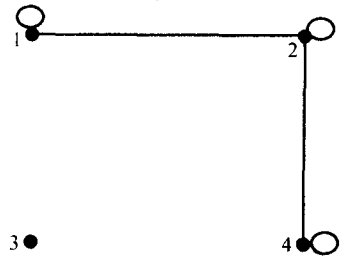
3. $V=\{1, 2, 3, 4\}$，$E=[\![(1, 1), (1, 2), (3, 4), (4, 4)]\!]$ □

练习 12.18　使用集合和元包表示下列各图。

1.



2.



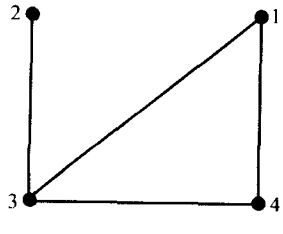
□

12.3 图的矩阵表示

如果希望使用高级编程结构(high level programming structure)的方式来表示图，也就是说，如果希望图能在计算机中表示，并且能利用计算机对图进行各种操作，那么除了用可视的方法来表示图以外，还需要其他表示图的方法。其中一种方法就是邻接矩阵(adjacency matrix)，这是适合于计算机操作的最常用的表示图的方法。

考虑图 $G=(V, E)$，且 $\#V=n$。假设 V 中的顶点可以被标记为(有序化为) v_1，$v_2, \ldots, v_n$。这时，G 的邻接矩阵就是一个 $n \times n$ 的矩阵，其中，位置 (i, j) 处的项表示顶点 v_i 和 v_j 之间的边的数目。

例如，考虑如下给出的图。



这个图有 4 个顶点，因此，表示这个图的邻接矩阵是一个 4×4 的矩阵。进而，为了简化起见，我们假设 $v_1=1$，$v_2=2$，$v_3=3$，$v_4=4$。另外，这个图有 4 条边：(1, 3)，(1, 4)，(2, 3)，(3, 4)。因此，在矩阵中与这些边相对应的位置的值是 1，没有边的位置的值是 0。假定左上角的位置对应于(1, 1)，其下一个位置代表(1, 2)，以此类推，那么这个图的邻接矩阵如下所示。

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

这个矩阵表示，顶点 1 通过一条边与顶点 3 相连(反之亦然)，顶点 1 通过一条边与顶点 4 相连(反之亦然)，顶点 2 通过一条边与顶点 3 相连(反之亦然)，顶点 3 通过一条边与顶点 4 相连(反之亦然)。

如果要从这个图中去掉连接顶点 3 和顶点 4 的边，那么邻接矩阵将变为

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

另一方面，如果想在原图的基础上，在顶点 3 和顶点 4 之间再加上一条平行的边，那么邻接矩阵将变为

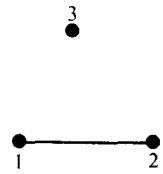
$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 2 \\ 1 & 0 & 2 & 0 \end{pmatrix}$$

回想一下，一个图被称为简单图，当且仅当这个图既没有平行边也没有环。因此，简单图的邻接矩阵表示将有两个特殊的性质。第一，矩阵中从左上到右下的对角线上的所有项都是 0(这是图中无环的性质)；第二，矩阵中所有项不是 0 就是 1(这是图中没有平行边的性质)。如上所示的图是简单图，因此，它的邻接矩阵满足这样的性质。

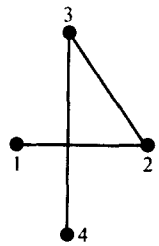
在顶点 3 和顶点 4 之间再加上第二条边就导致一个非简单图，因此，它的邻接矩阵不再满足第二个性质。

练习 12.19　给出下列各图的邻接矩阵。

1.



2.



□

练习 12.20　画出下列邻接矩阵所表示的图。

1. $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$

□

练习 12.21　下列邻接矩阵中哪些表示简单图？

1. $\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

3. $\begin{pmatrix} 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

□

由于我们所讨论的是无向图(将在 12.9 节中讨论有向图)，因此，邻接矩阵(i, j)位置上的项总与这一矩阵的(j, i)位置上的项相等。这是因为，这两个项都表示顶点 v_i 和 v_j 之间的边的数目。显然，这是一种对称性：对于无向图，如果顶点 v_i 相邻于 v_j，那么 v_j 也相邻于 v_i。正因如此，当我们处理无向图时，只需要考虑邻接矩阵的左下半部：这和整个矩阵所提供的信息是同样的。

例如，考虑下面的矩阵。

$$\begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

这也可以表示成如下的形式。

$$\begin{pmatrix} 0 & & & \\ 0 & 0 & & \\ 1 & 1 & 0 & \\ 1 & 0 & 1 & 0 \end{pmatrix}$$

第二种方法的主要优点体现在计算机实现上：当存储一个图的信息时，后者只用了前者一半略多一点的空间，当处理巨型的图时，这样做可以节省大量的空间。

练习 12.22　使用邻接矩阵表示练习 12.19 中的图，其中邻接矩阵只使用它的左下半部来表示。　□

练习 12.23　假设 $v_1=1$，$v_2=2$，$v_3=3$，$v_4=4$，画出下列邻接矩阵表示的图。

1. $\begin{pmatrix} 0 & & & \\ 1 & 0 & & \\ 1 & 1 & 0 & \\ 0 & 0 & 1 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 0 & & & \\ 0 & 0 & & \\ 0 & 1 & 0 & \\ 1 & 0 & 1 & 0 \end{pmatrix}$

□

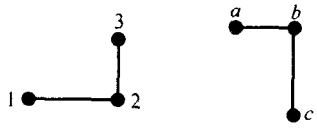
练习 12.24 如果一个图有 n 个顶点。那么，使用左下半部所表示的邻接矩阵中包含多少个项？ □

12.4 图的同构

回想一下第 5 章中讲述的布尔代数的同构的概念。在那一章中已经表明，如果两个布尔代数 A 和 B 具有相同的结构，那么可以认为它们是同构的。证明两个布尔代数同构包括给出我们后来学到的全双射函数，$f: A \twoheadrightarrow B$，使得这个函数将 A 中的元素映射到 B 中，且"保持"A 的运算(一个函数是双射意味着它既是单射又是满射)。例如，可以证明，集合论的布尔代数与命题逻辑的布尔代数是同构的。按照同样的方法，也可以推断出两个图是否同构。另外，两个图的同构的定义与两个布尔代数的同构的定义类似。

考虑两个图 $G_1=(V_1, E_1)$，$G_2=(V_2, E_2)$。形式地，为了证明 G_1 和 G_2 同构，需要证明存在全双射 $f: V_1 \twoheadrightarrow V_2$，使得对于任意的顶点 u，$v \in V_1$，u 和 v 在 G_1 中相邻当且仅当 $f(u)$ 和 $f(v)$ 在 G_2 中相邻。用非形式的语言来描述的话，对于两个图 G_1 和 G_2，如果可以将 G_1 的所有顶点的名字改成 G_2 的顶点的名字，而且可以使用改变后的名字重新安排 G_1 的形状使其与 G_2 的形状吻合，那么这两个图就是同构的。

例 12.9 下面的两个图是同构的。

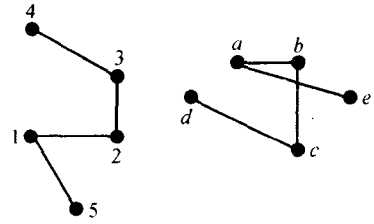


对于这两个图，可以给出如下所示的全双射 f。

$f=\{1 \mapsto c,\ 2 \mapsto b,\ 3 \mapsto a\}$

另外，在第一个图中，1 和 2 是相邻的，且 2 和 3 是相邻的，而在第二个图中，c 和 b 是相邻的，且 b 和 a 是相邻的。 □

例 12.10 下面的两个图是同构的。



对于这两个图，可以给出如下所示的全双射 f。

$f=\{1 \mapsto c,\ 2 \mapsto b,\ 3 \mapsto a,\ 4 \mapsto e,\ 5 \mapsto d\}$

在第一个图中，下面的顶点对是相邻的：1 和 2，1 和 5，2 和 3，3 和 4。在第二个图中，可以看出 $f(1)$ 和 $f(2)$ 是相邻的，$f(1)$ 和 $f(5)$ 是相邻的，$f(2)$ 和 $f(3)$ 是相邻的，$f(3)$ 和 $f(4)$ 是相邻的。另外，在第二个图中没有其他顶点是相邻的。因此，这两个图是同构的。 □

练习 12.25 考虑两个图 G_1 和 G_2，G_1 有 m 个顶点，G_2 有 n 个顶点，并且 $m \neq n$，那么 G_1 和 G_2 有可能同构吗？ □

练习 12.26 考虑两个图 G_1 和 G_2，G_1 有 m 条边，G_2 有 n 条边，并且 $m \neq n$，那么 G_1 和 G_2 有可能同构吗？ □

练习 12.27 考虑如下的顶点集合 V_1 和 V_2。

$V_1=\{a, b, c, d\}$

$V_2=\{1, 2, 3, 4\}$

对于下列各元包 E_1，分别定义元包 E_2，使得 (V_1, E_1) 和 (V_2, E_2) 同构。

1. $E_1=[\![(a, a), (a, b), (c, d)]\!]$

2. $E_1=[\![(a, d), (c, d), (b, c), (b, d)]\!]$ □

练习 12.28 下列各图中哪些是同构的？

$V_1=\{1, 2, 3, 4\}$；$E_1=[\![(1, 3), (1, 4), (2, 4), (2, 2)]\!]$

$V_2=\{a, b, c, d\}$；$E_2=[\![(a, a), (c, b), (d, b), (c, d)]\!]$

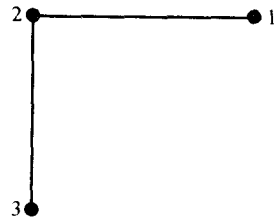
$V_3=\{1, 2, 3, 4\}$；$E_3=[\![(4, 2), (4, 3), (1, 3), (1, 1)]\!]$

$V_4=\{a, b, c, d, e\}$；$E_4=[\![(a, b), (c, d), (a, b)]\!]$

$V_5=\{1, 2, 3, 4, 5\}$；$E_5=[\![(1, 2), (3, 4), (1, 3)]\!]$ □

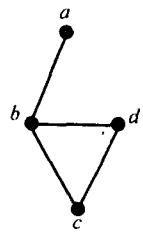
12.5 路径

考虑下图。这里，1 和 2 相连，2 和 3 相连。如果把这个图的边想象成商业航班的路线，那么可以从 1 通过 2 飞到 3，虽然不能直接从 1 飞到 3。这就是路径(path)的例子。



可以形式地定义路径如下。一条长度为 n 的路径○ 是一个顶点的序列 $\langle v_0, v_1, \ldots, v_n\rangle$，使得对于每个满足 $0\leqslant i\leqslant n-1$ 的 v_i 都与 v_{i+1} 相邻。

例 12.11　考虑下面的图。

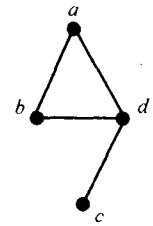


这个图包含若干路径。例如，$\langle a, b, d\rangle$ 和 $\langle a, b, c\rangle$ 都是长度为 2 的路径。注意，对于路径中访问给定顶点的次数并没有限制，所以 $\langle a, b, d, c, b\rangle$ 也是一条路径。另外，$\langle a\rangle$ 也是一条路径，它的长度为 0。 □

简单路径(simple path)是在其中任何一个顶点最多出现 1 次的路径。

例 12.12　参看例 12.11 中的图，$\langle a, b, d\rangle$ 是简单路径，而 $\langle a, b, d, c, b\rangle$ 不是简单路径。 □

练习 12.29　列举出下图中所有长度为 2 的路径。



□

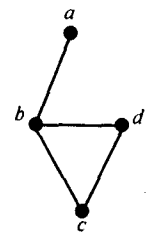
练习 12.30　列举出上图中所有长度为 2 的简单路径。 □

练习 12.31　画出有顶点 a、b、c 和 d 的图，且这个图由下列简单路径组成。

$\langle a, b, c\rangle$，$\langle a, b, d\rangle$，$\langle c, b, a\rangle$，$\langle c, b, d\rangle$，$\langle d, b, a\rangle$，$\langle d, b, c\rangle$ □

回路(circuit)定义为使得 $v_0=v_n$ 的路径 $\langle v_0, v_1, \ldots, v_n\rangle$。

例 12.13　考虑下面的图。



这个图包含下列长度为 3 的回路。

$\langle b, c, d, b\rangle$，$\langle b, d, c, b\rangle$，$\langle c, b, d, c\rangle$，$\langle c, d, b, c\rangle$，$\langle d, b, c, d\rangle$，$\langle d, c, b, d\rangle$ □

练习 12.32　考虑练习 12.29 中的图，列举出所有不含重复边的回路。 □

○ 可是，此处的路径概念与其他教科书多少有些不同。应该注意的是，这并不意味着这一定义或其他定义是不正确的；只是图论中的某些术语没有标准定义。这一问题同样发生在后面的环和回路的定义上。

现在回顾一下一个图是连通图的含义，并给出连通图的形式定义。为此，可以做出如下定义：一个图 $G=(V, E)$ 是连通的，当且仅当对所有的顶点对 u，$v\in V$，G 中都存在一条从 u 到 v 的路径。

练习 12.33 下列各图中哪些是连通的？

1. $V=\{1, 2, 3\}$；$E=[\![(1, 2), (1, 3)]\!]$

2. $V=\{1, 2, 3, 4\}$；$E=[\![(1, 2), (2, 3), (3, 4), (4, 1)]\!]$

3. $V=\{1, 2, 3, 4, 5\}$；$E=[\![(1, 3), (2, 4), (2, 5), (3, 4)]\!]$ □

练习 12.34 对于下面给出的各顶点集合，说明以这些集合为顶点集合且不连通的简单图的边的最大数目。

1. $V=\{1, 2, 3\}$

2. $V=\{1, 2, 3, 4\}$

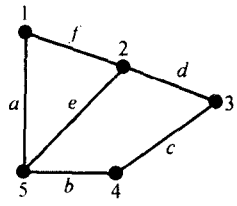
3. $V=\{1, 2, 3, 4, 5\}$

4. $V=\{1, 2, 3, 4, 5, 6\}$ □

现在，考虑两类特殊的路径，**欧拉路径**(Eulerian path)[1]和**哈密尔顿路径**(Hamiltonian path)[2]。我们先讨论前一类路径。

连通图 $G=(V, E)$ 中的**欧拉路径**是图中的每一条边 $e\in E$ 刚好通过一次的路径。相应地，**欧拉回路**(Eulerian circuit)是同时是回路的欧拉路径。

例 12.14 考虑下面的图。

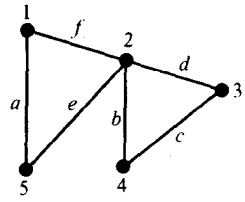


如下所示，这个图中有 4 条欧拉路径。

$\langle 2, 5, 4, 3, 2, 1, 5\rangle$，$\langle 2, 5, 1, 2, 3, 4, 5\rangle$，$\langle 5, 1, 2, 3, 4, 5, 2\rangle$，$\langle 5, 4, 3, 1, 5, 2\rangle$ □

例 12.14 中的图有 4 条欧拉路径，但是没有欧拉回路。包含欧拉路径但不包含欧拉回路的连通图，称为**半欧拉图**(semi-Eulerian)。含有欧拉回路的连通图叫做**欧拉图**。

例 12.15 下面的图是欧拉图，因为它是连通图，且包含欧拉回路。



□

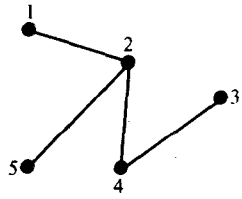
练习 12.35 列举出下面图中的所有欧拉路径。

$V=\{1, 2, 3, 4\}$

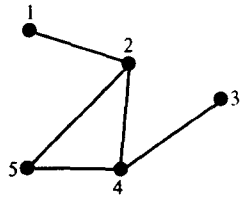
$E=[\![(1, 2), (1, 3), (1, 4), (2, 3), (3, 4)]\!]$ □

练习 12.36 对于下列各图，说明它们是欧拉图、半欧拉图还是两者都不是。

1.



2.

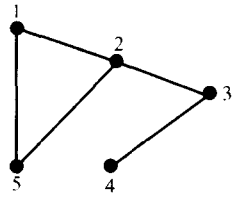


---

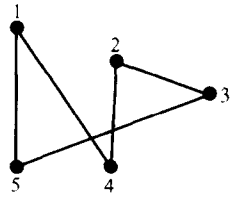
[1] 以 Leonhard Euler(1707—1783)命名。关于欧拉的工作参看[Dun99]。

[2] 以 Sir William Hamilton(1805—1865)命名。

3.



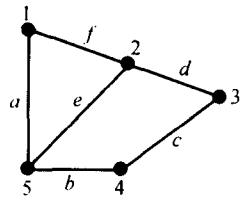
4.



□

连通图 $G=(V, E)$ 中的哈密尔顿路径(Hamiltonian path)是图中的每个顶点 $v\in V$ 都恰好被访问一次的路径(不包括哈密尔顿回路(Hamiltonian circuit)，在哈密尔顿回路中 $v_0=v_n$)。

例 12.16　再一次考察下面的图。



这个图包含若干条哈密尔顿路径。例如，路径〈1，2，3，4，5〉、〈1，2，5，4，3〉和〈1，5，2，3，4〉都满足哈密尔顿路径的性质。另外，该图还包含若干条哈密尔顿回路，例如〈1，2，3，4，5，1〉就是一条哈密尔顿回路。包含哈密尔顿路径的连通图称为哈密尔顿图。　□

练习 12.37　列举出练习 12.35 的图中所包含的所有哈密尔顿路径。　□

练习 12.38　再次考虑练习 12.36 中的各图，对于每一个图，说明它是否是哈密尔顿图。　□

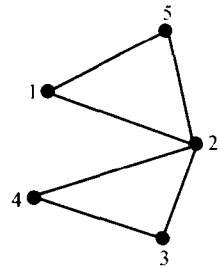
12.6　循环

上节讨论了路径的概念，本节讨论称为循环(cycle)的一种特殊路径。循环具有下列性质。

1. 它至少有一条边。
2. 它没有重复的边。
3. 它只有两个重复的顶点：第一个和最后一个。

我们认为循环是只有一个重复顶点的回路。

例 12.17　考虑下面的图。



在这个图中，有若干循环，包括〈2，1，5，2〉和〈2，3，4，2〉。　□

注意，在上例中，〈2，3，4，2，1，5，2〉不是循环，因为顶点 2 被访问了三次。然而，它是一条回路。同时也要注意，我们认为循环〈1，5，2，1〉和〈2，1，5，2〉等价。更一般地，当用序列和边表示回路时，若两条回路 c_1 和 c_2 满足 $\text{ran}\ c_1=\text{ran}\ c_2$，我们认为 c_1 和 c_2 等价。

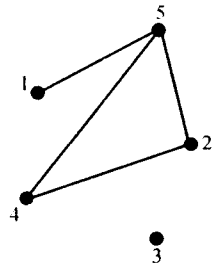
练习 12.39　在图中，是否可能有包含两条边的循环？　□

练习 12.40　在图中，是否可能有包含一条边的循环？　□

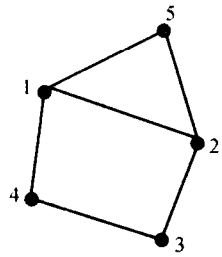
练习 12.41　回想一下简单图是既没有环(loop)也没有平行边的图。带有一个循环的简单图的边的最小数目是多少？　□

练习 12.42　列举出下列各图所包含的所有循环。

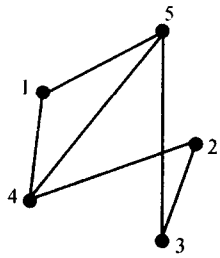
1.



2.



3.



□

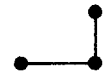
12.7 树

树(tree)是一种特殊的图，它在计算学科中的很多场合出现。例如，在第 13 章，我们将介绍树图(tree diagram)的概念，这种结构可以图示一系列事件的不同组合。另外，在本章的后面我们将介绍二叉树的概念，二叉树是高级程序设计中的一种基本数据结构。

本质上，树是没有循环的连通图。

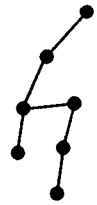
例 12.18 下面 4 个图都是树，因为它们满足树的性质。

1.

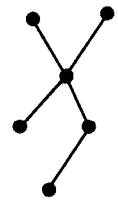


2.

3.

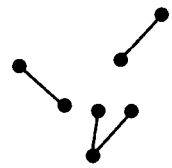


4.



□

如果一个图不含循环但不是连通的，那么称之为森林(forest)。下面就是一个森林。



本质上，森林是由若干棵树组成的图。

树具有下面的性质。

对于任意的树 T，如果 T 有 n 个顶点，那么 T 有 $n-1$ 条边。

通过考察上面所给出的树的例子，可以很容易地验证这一性质。

练习 12.43 考虑图 $G=(V, E)$，使得 G 是树。如果 $\#V=4$，那么 $\#E$ 是多少？

□

练习 12.44 下列各图哪些是树？

1. $V=\{1, 2, 3, 4\}$；$E=[\![(1, 2), (2, 3), (2, 4)]\!]$

2. $V=\{1, 2, 3, 4\}$; $E=[\![(1, 2), (2, 3), (2, 4), (3, 4)]\!]$

3. $V=\{1, 2, 3, 4\}$; $E=[\![(1, 2), (1, 3), (3, 4)]\!]$

4. $V=\{1, 2, 3, 4\}$; $E=[\![(1, 2), (1, 3), (2, 4), (3, 4)]\!]$ □

练习 12.45 证明：对于任意给定的树 T，从 T 中移除一条边但不移动任何顶点，这将产生一个不连通的图。 □

练习 12.46 证明：对于任意给定的树 T，给 T 增加一条新边但不增加任何新顶点，这将产生一个含有循环的图。 □

练习 12.47 考虑图 $G=(V, E)$，使得 G 是树。任意给定两个顶点 v_1，$v_2 \in V$，v_1 和 v_2 之间有多少条路径？ □

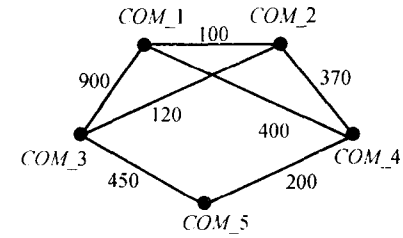
12.8 带权图

在本章的开始，通过说明图论是计算机科学许多领域的核心，阐明了学习图论的动机。计算机的网络化就是一个这样的例子。如果要把多台计算机连接起来，那么我们希望直接或间接地把每一台计算机与其他每一台计算机连接在一起；希望计算机的连接方式具有连通图的性质。

在许多实际情况中，例如计算机网络，仅仅表述机器的连通方法还不够；在某些情况下，需要考虑采用不同的网络配置的成本，然后按照成本选择最适宜的配置(一般是成本最低的)。

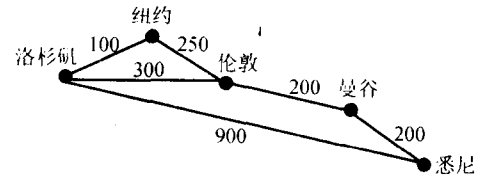
为了考虑那些诸如成本、距离等因素起重要作用的问题，需要考虑带权图(weighted graph)。带权图为每一条边上都有一个相关的权(weight)的简单图。

例 12.19 假设有分布在不同建筑物中的五台计算机 COM_1、COM_2、COM_3、COM_4、COM_5。连接每个机器对的成本如下图所示。



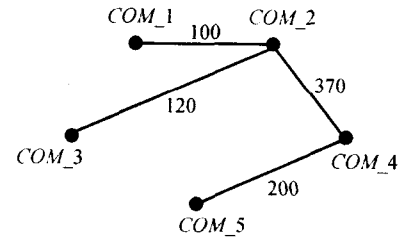
□

例 12.20 一家旅游公司提供五个城市间的航班，这几个城市分别是曼谷、悉尼、伦敦、洛杉矶和纽约。每个城市对间的飞行成本如下图所示。



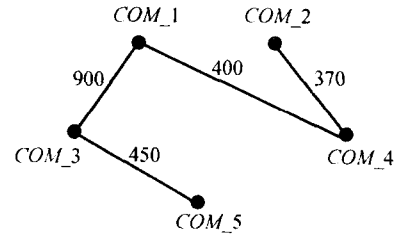
□

在例 12.19 中，说明了计算机连接的可能方式及每种连接方式的成本。当然，实际中不会同时安装所有的连接；只需要安装必要的连接，保障所有的机器都相连就可以了。例如，下图所表示的安装方案就足够了(同时也是成本最低的安装方案)。



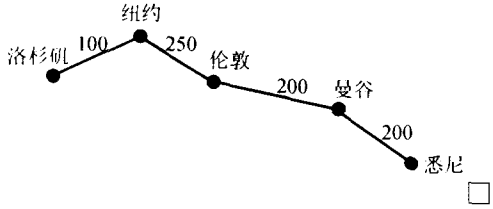
上面的图满足树的性质：它是连通图且没有循环。另外，它与原始图具有相同的顶点，其边的集合是原始图的边的集合的子集。称这样的树为生成树(spanning tree)。

对于计算机网络这个图的另外一棵生成树如下所示。



这个图也满足生成树的性质。然而，与第一棵生成树相比较，它不具有优越性：第一棵生成树的成本为 790，而第二棵生成树的成本为 2120。因此，从成本的角度看，第一棵生成树比第二棵更好，因为它更便宜。而且在计算机网络中，没有比第一棵生成树更便宜的生成树，这样的生成树称为最小生成树(minimal spanning tree)。

例 12.21 例 12.20 中的图的最小生成树如下所示。

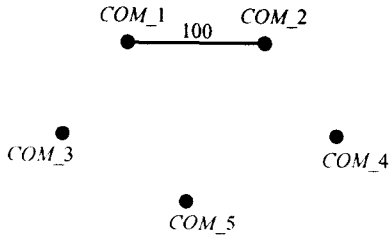


□

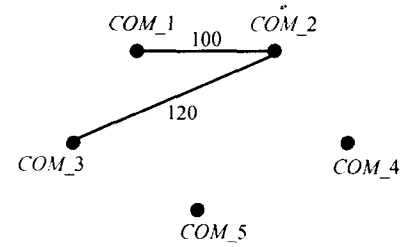
下面介绍求最小生成树的一种方法。从任意一个顶点出发，选择与这个顶点相连接的边中权最小的边。标记这条边及通过这条边连接在一起的两个顶点。下一步，在连接已被标记的顶点和没有被标记的顶点的边中，选择一条权最小的边，标记这条边以及通过这条边连接起来的两个顶点。重复这一过程，直到所有的顶点都被标记，这时，所有被标记的边形成一棵生成树。这个过程就是普里姆算法(Prim's algorithm)。

例 12.22 可以对例 12.19 中的带权图运用普里姆算法，其过程如下所示。

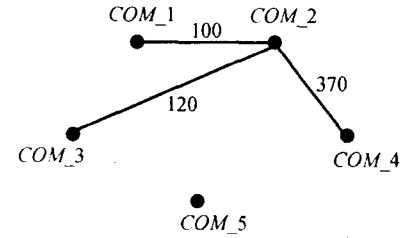
首先，不标记任何边。从 COM_1 开始，找到最便宜的边是连接 COM_1 和 COM_2 的边。因此，树变成如下形式。



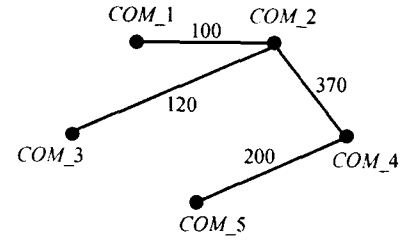
连接 COM_1 和 COM_2 之一与 COM_3、COM_4 和 COM_5 之一的最便宜的边是连接 COM_2 和 COM_3 的边。因此，把这条边加入到树中。



连接 COM_1、COM_2 和 COM_3 之一与 COM_4 和 COM_5 之一的最便宜的边是连接 COM_2 和 COM_4 的边。因此，把这条边加入到树中。



最后，连接 COM_1、COM_2、COM_3 和 COM_4 之一与 COM_5 的最便宜的边是连接 COM_4 和 COM_5 的边。因此，把这条边加入到树中。



这就是我们要求的最小生成树。 □

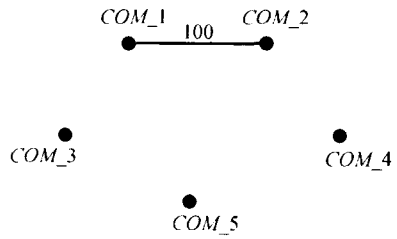
练习 12.48 使用普里姆算法求例 12.20 中的图的最小生成树。 □

另一个计算最小生成树的算法是克鲁斯卡尔算法(Kruskal's algorithm)。这个算法的步骤如下所示。首先，选择一条成本最低的边，标记这条边。然后，选择下一条成本最低的边，标记这条边。其次，选择下一条成本最低的边，如果由这条边以及前面已经

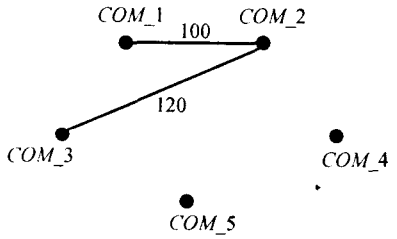
标记的边组成的图含有循环，那么就忽略这条边；否则标记这条边。持续这一过程，直到被标记的边形成一棵生成树。

例 12.23　可以对例 12.19 中的图运用克鲁斯卡尔算法，其过程如下所示。

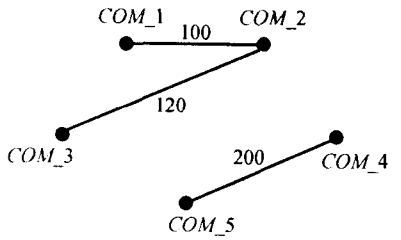
首先，最便宜的边是连接 *COM_1* 和 *COM_2* 的边。因此，树变成如下形式。



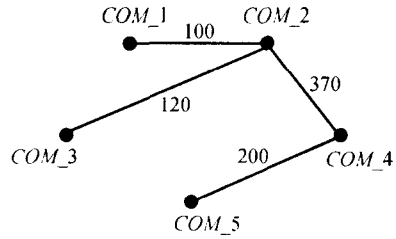
下一条最便宜的边是连接 *COM_2* 和 *COM_3* 的边。而且，这条边不生成循环。因此，把这条边加入到树中。



下一条最便宜的边是连接 *COM_4* 和 *COM_5* 的边。而且，这条边不生成循环。因此，把这条边加入到树中。



下一条最便宜的边是连接 *COM_2* 和 *COM_4* 的边。而且，这条边不生成循环。因此，把这条边加入到树中。



这就是我们要求的最小生成树。 □

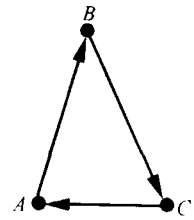
练习 12.49　运用克鲁斯卡尔算法求例 12.20 中的图的最小生成树。 □

12.9　有向图

在很多实际应用中，使用无向图（即到目前为止我们所研究的图）可以颇为完美地解决很多问题。然而，在某些应用中，顶点间的连接关系只是单方向的。例如，在三个城市 *A*、*B* 和 *C* 间的飞行航线网络中，飞机可以从 *A* 直接飞到 *B*，从 *B* 直接飞到 *C*，从 *C* 直接飞到 *A*，而不能往相反的方向飞行。用无向图定义这一网络的表示无法完全提供我们所需要的信息：它只能表示 *A* 和 *B* 是连通的，*B* 和 *C* 是连通的，*C* 和 *A* 是连通的。相反，有向图（directed graph）则能帮助我们表示这样的网络[○]。有兴趣的读者可以参见[Jen00]以获得更多有向图应用的介绍。

形式地，可以用与表示无向图同样的方法表示有向图的成分：用一个顶点的集合和一个边的元包，与无向图不同的是，在有向图中所有的边都是有向的，在有向图的图示中用箭头来表明边的方向。

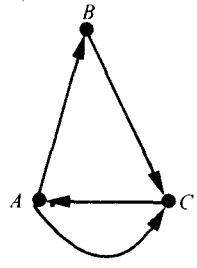
例 12.24　例如，下图表示上述的飞行航线网络。



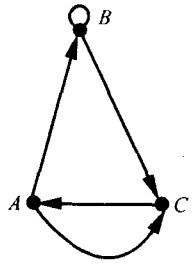
○ 在某些教科书中，有向图（directed graph）写作 digraph。

在这里，用从 A 指向 B 的箭头来表示从 A 到 B 有一条有向边。类似地，从 B 到 C 和从 C 到 A 各有一条边。 □

当然，有向图并非意味着两个顶点之间不能有相反方向的边。例如，假设航空公司除了有从 C 到 A 的航线外，又增设了从 A 飞到 C 的航线，那么可以用下图表示整个航线。



另外，如下图所示，有向图可以拥有环(loop)。



当然，可以用关系和矩阵来表示有向图。例如，上图的关系表示如下。

$V=\{A, B, C\}$

$E=[\![(A, C), (A, B), (B, B), (B, C), (C, A)]\!]$

另外，上图的矩阵表示如下所示(假设 A 是顶点 v_1，B 是顶点 v_2，C 是顶点 v_3)。

$$\begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix}$$

与无向图的矩阵表示不同，表示有向图的矩阵不是对称的。因此，对于无向图的矩阵表示适用的“左下”矩阵表示对于表示有向图是不适用的。

另外，当考虑用集合和元包的有向图表示时，E 中元素的成分的顺序是重要的：在无向图中，a 和 b 相连接可以写作 $(a, b)\in E$ 或者 $(b, a)\in E$，这没有区别。而在有向图中，$(a, b)\in E$ 表示存在从 a 到 b 的一条有向边，而 $(b, a)\in E$ 表示存在从 b 到 a 的一条有向边，这是完全不同的两条边。

练习 12.50 画出下列有向图。

1. $V=\{a, b, c, d, e\}$；$E=[\![(a, b), (b, d), (d, e), (a, e)]\!]$

2. $V=\{a, b, c, d, e\}$；$E=[\![(b, c), (b, b), (c, b)]\!]$

3. $V=\{a, b, c, d, e\}$；$E=[\![(a, b), (b, c), (c, d), (d, e), (e, a)]\!]$ □

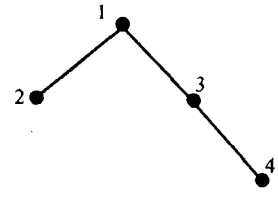
练习 12.51 构造练习 12.50 中的各有向图的矩阵表示。假设 $v_1=a$，$v_2=b$，$v_3=c$，$v_4=d$，$v_5=e$。 □

12.10 二叉树

二叉树(binary tree)是计算机科学和离散数学中最基础的结构。本质上，二叉树是满足一定性质的树○。

二叉树有一个被称为根节点(root node)的特殊元素，并且根节点一般画在树的顶端(这与树的直观形式相反)。另外，二叉树的每一个节点(在叙述有关二叉树的问题时，常称其顶点为节点(node))，包括根节点，最多有两个后继节点(descendant)或子节点(child)。

考虑如下给出的二叉树。



○ 注意，在某些教科书中，二叉树被称为带根树(rooted tree)。

这里，根节点是节点 1。另外，根节点有两个子节点：节点 2 和节点 3。节点 2 称为节点 1 的*左孩子*(left child)，而节点 3 称为节点 1 的*右孩子*(right child)。相反，节点 1 称为节点 2 和节点 3 的*双亲节点*(parent)。以此类推，节点 3 有一个孩子：节点 4。节点 2 和节点 4 没有孩子，没有孩子的节点称为*叶节点*(leaf node)。

注意二叉树的画法：根节点画在最上层，它的孩子放在下一层。根节点的后继节点的子节点按此规律一层一层地向下排列。值得注意的是，根节点的左边分支的所有节点也形成一棵树，同样根节点的右边分支的所有节点也形成一棵树。称根节点左边分支上的所有节点组成的树为根节点的*左子树*(left subtree)。类似地，称根节点右边分支上的所有节点组成的树为根节点的*右子树*(right subtree)。这样，节点 3 有一个右子树，没有左子树。节点 2 和节点 4 是叶节点，它们没有子树。

回忆第 8 章介绍的偏序(partial ordering)，显然，二叉树给出了偏序的一种图形化表示：自反性、反对称性和传递性都成立。

首先，作为一个示例，设 $x \leqslant y$ 表示节点 x 是节点 y 的后继节点或者就是节点 y 本身。显然，自反性成立，对于树中任意的节点 x 都有 $x \leqslant x$ 成立。其次，传递性成立，因为若节点 x 是节点 y 的后继，节点 y 是节点 z 的后继，则节点 x 是节点 z 的后继。最后，反对称性成立，因为若对任意的节点 x 和 y，$x \leqslant y$ 和 $y \leqslant x$ 同时成立，则一定有 $x=y$ 成立。

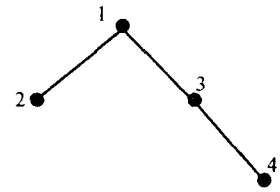
练习 12.52　画出下面的二叉树。

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

$E=[\![(d, e), (d, f), (b, a), (a, d), (b, c)]\!]$　□

二叉树的*高度*(height)是指从根节点到最低层的某个叶节点的路径的长度。特别地，空二叉树的高度为 0。

例 12.25　再次考虑下面的二叉树。



这棵树的高度为 2。　□

练习 12.53　练习 12.52 中的二叉树的高度是多少？　□

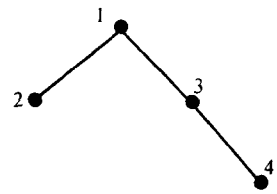
练习 12.54　带有 n 个节点的二叉树的最大高度是多少？　□

练习 12.55　带有 n 个节点的二叉树的最小高度是多少？　□

可以使用三种方法对二叉树进行*遍历*(traverse)，也就是说，从根节点出发访问每一个节点：*前序*(pre-order)方法，*中序*(in-order)方法，*后序*(post-order)方法。通过遍历二叉树可以列举出树中出现的所有节点，不同的遍历方法给出的节点顺序是不同的。下面依次讨论这三种遍历方法。

使用前序方法遍历一棵二叉树时，首先访问它的根节点，接着遍历其左子树，最后遍历其右子树。这种模式遍访整棵树，依次列举出根节点、左子树中的节点、右子树中的节点。

例 12.26　再一次考虑下面的二叉树。



用前序方法遍历这棵二叉树得到的节点顺序为 1，2，3，4。　□

用中序方法遍历一棵二叉树时，首先遍历其左子树，接着访问它的根节点，最后遍历其右子树。这一模式遍访整棵二叉树，依次列出左子树中的节点、根节点、右子树中

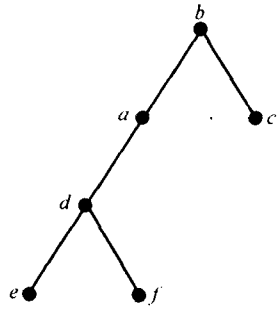
的节点。

例 12.27 中序遍历上面的二叉树得出的节点顺序为 2，1，3，4。 □

最后，用后序方法遍历一棵二叉树时，首先遍历其左子树，接着遍历其右子树，最后访问它的根节点。这一模式遍访整棵二叉树，依次列出左子树中的节点、右子树中的节点、根节点。

例 12.28 后序遍历上面的二叉树得出的节点顺序为 2，4，3，1。 □

练习 12.56 考虑下面的二叉树。



用下列方法列出这棵二叉树的节点。

1. 前序方法。
2. 中序方法。
3. 后序方法。 □

12.11 附加练习

练习 12.57 画出下列各(无向)图。

1. $V=\{a, b, c, d, e\}$；$E=[\![(a, b), (b, e), (c, e), (d, e)]\!]$

2. $V=\{a, b, c, d, e\}$；$E=[\![(a, c), (a, e), (b, c), (b, d), (b, e)]\!]$

练习 12.58 给出练习 12.57 中的图的矩阵表示，假设 $v_0=a$，$v_1=b$，等等。

练习 12.59 使用左下矩阵法给出练习 12.57 中的图的邻接矩阵。

练习 12.60 证明下列各图间互不同构。

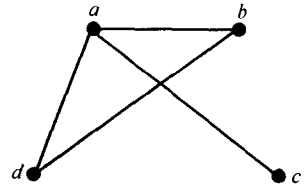
$V_1=\{a, b, c, d\}$；$E_1=[\![(a, b), (c, d), (a, d)]\!]$

$V_2=\{x, y, z\}$；$E_2=[\![(x, y), (y, z), (x, z)]\!]$

$V_3=\{1, 2, 3, 4\}$；$E_3=[\![(1, 3), (2, 3), (3, 4)]\!]$

$V_4=\{a, b, c, d\}$；$E_4=[\![(a, b), (c, d)]\!]$

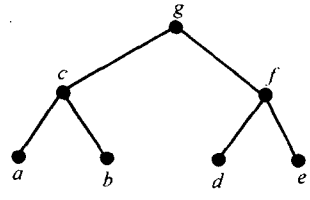
练习 12.61 列举出下面图中所含有的所有欧拉路径。



练习 12.62 列举出练习 12.61 的图中所含有的所有哈密尔顿路径。

练习 12.63 练习 12.61 的图中有多少个循环？

练习 12.64 考虑如下给出的二叉树。



使用下列各方法遍历这棵二叉树。

1. 前序方法。
2. 中序方法。
3. 后序方法。

练习 12.65 练习 12.64 中的树的高度是多少？

练习 12.66 考虑如下给出的顶点集合。

$V=\{a, b, c\}$

1. 由 V 中一个元素构成的二叉树有多少棵？

2. 由 V 中两个元素构成的二叉树有多少棵？

3. 由 V 中三个元素构成的二叉树有多

少棵？

12.12 练习解答

12.1 下列顶点对是相邻的：1 和 3，1 和 4，2 和 3，3 和 4。

12.2 边 a 关联于顶点 2 和 3；边 b 关联于顶点 1 和 3；边 c 关联于顶点 3 和 4；边 d 关联于顶点 1 和 4。

12.3 1 不是简单图，因为它有环；2 是简单图，因为它没有环，也没有平行边；3 不是简单图，因为它有平行边。

12.4 是的，因为如果一个图没有边，那么它当然也没有环和平行边。因此，它一定是简单图。

12.5 答案由如下公式给出。

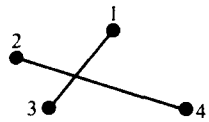
$$(n-1)+(n-2)+(n-3)+..+1$$

它等价于下式。

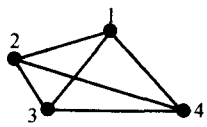
$$\frac{n\times(n-1)}{2}$$

12.6

1.



2.



12.7

$deg(1)=2$

$deg(2)=3$

$deg(3)=4$

$deg(4)=2$

$deg(5)=3$

$deg(6)=0$

12.8 只有顶点 6 是孤立顶点；其余各顶点的度至少为 1。

12.9 可以如下证明这一等式成立。

$$2+3+4+2+3+0=14$$
$$=2\times7$$

12.10 这一引理成立，因为有 2 个顶点(顶点 2 和顶点 5)的度是奇数。

12.11 1 和 3 是连通图。

12.12 它最少有 $n-1$ 条边。

12.13 图 2 是完全图；其余两个图则不是。

12.14 这个图一定至少有

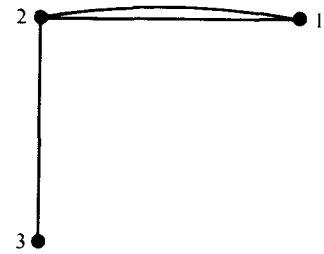
$$(n-1)+(n-2)+..+1$$

条边，它等价于

$$\frac{n\times(n-1)}{2}$$

12.15 假设图 G 是完全的。根据定义，这个图的每一个顶点都与其余顶点相邻。因此，可以从每一个顶点移动到另一个顶点。所以 G 是连通的。

12.16 E 是由元包而非集合定义的，这样，可以讨论平行边。考虑下图。



这里存在平行边，如果 E 是集合而不是元包，那么 E 的形式如下所示。

$$E=\{(1,2),(2,3)\}$$

另一方面，由于用元包来如下定义 E，所以可以看到顶点 1 和顶点 2 之间有 2 条边。

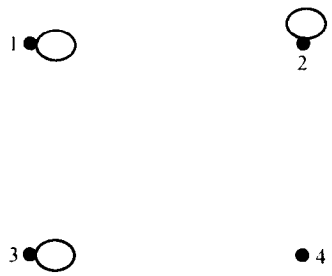
$$E=[\![(1,2),(1,2),(2,3)]\!]$$

当然，如果只考虑简单图，那么，把 E 表示成集合也完全可以。

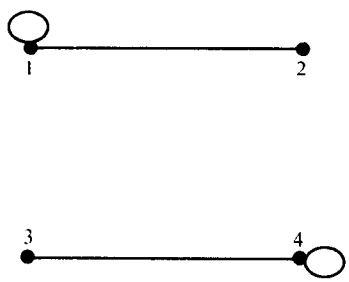
12.17

1.

2.



3.



12.18

1. $V=\{1, 2, 3, 4\}$, $E=[\![(1, 2), (1, 3), (2, 4), (3, 4)]\!]$

2. $V=\{1, 2, 3, 4\}$, $E=[\![(1, 1), (1, 2), (2, 2), (2, 4), (4, 4)]\!]$

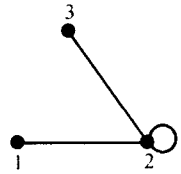
12.19

1. $\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$

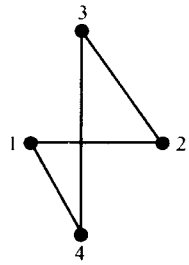
2. $\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$

12.20

1.



2.



12.21 1 表示一个简单图：它没有边（因此，既没有环也没有平行边）。2 不表示简单图，因为它（在顶点 v_1 处）含有一个环。最后，3 不表示简单图，因为它（在顶点 v_1 和 v_2 之间）含有平行边。

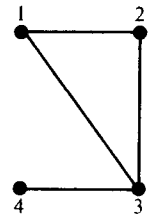
12.22

1. $\begin{pmatrix} 0 & & \\ 1 & 0 & \\ 0 & 0 & 0 \end{pmatrix}$

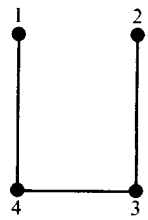
2. $\begin{pmatrix} 0 & & & \\ 1 & 0 & & \\ 0 & 1 & 0 & \\ 0 & 0 & 1 & 0 \end{pmatrix}$

12.23

1.



2.



12.24　有$(n^2+n)/2$个项。

12.25　不能同构。为使G_1和G_2同构，需要存在全双射$f: V_1 \to V_2$。然而，如果集合V_1和V_2的势不同，那么不可能存在这样的函数。所以，G_1和G_2不可能同构。

12.26　不能同构。如果G_1和G_2是同构的，那么我们需要证明：u和v在G_1中是相邻的当且仅当$f(u)$和$f(v)$在G_2中是相邻的。然而，如果E_1和E_2的势不相同，上面的性质不可能成立。因此，G_1和G_2不可能同构。

12.27[○]

1. $E_2=[\![(1, 1), (1, 2), (3, 4)]\!]$

2. $E_2=[\![(1, 4), (3, 4), (2, 3), (2, 4)]\!]$

12.28　(V_1, E_1)和(V_3, E_3)是同构的，相关的全双射f如下所示。

$f=\{1\mapsto 4, 3\mapsto 2, 4\mapsto 3, 2\mapsto 1\}$

其余的图都不相互同构。

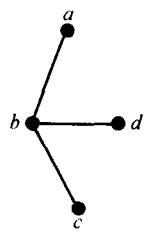
12.29　路径如下所示。

$\langle a, b, d\rangle$，$\langle a, b, a\rangle$，$\langle a, d, a\rangle$，$\langle a, d, b\rangle$，$\langle a, d, c\rangle$，$\langle b, a, b\rangle$，$\langle b, a, d\rangle$，$\langle b, d, a\rangle$，$\langle b, d, b\rangle$，$\langle b, d, c\rangle$，$\langle c, d, c\rangle$，$\langle c, d, a\rangle$，$\langle c, d, b\rangle$，$\langle d, a, b\rangle$，$\langle d, a, d\rangle$，$\langle d, b, a\rangle$，$\langle d, b, d\rangle$，$\langle d, c, d\rangle$

12.30　路径如下所示。

$\langle a, b, d\rangle$，$\langle a, d, b\rangle$，$\langle a, d, c\rangle$，$\langle b, a, d\rangle$，$\langle b, d, a\rangle$，$\langle b, d, c\rangle$，$\langle c, d, a\rangle$，$\langle c, d, b\rangle$，$\langle d, a, b\rangle$，$\langle d, b, a\rangle$

12.31



12.32　回路如下所示。

$\langle a, b, d, a\rangle$，$\langle a, d, b, a\rangle$，$\langle b, a, d, b\rangle$，$\langle b, d, a, b\rangle$，$\langle d, a, b, d\rangle$，$\langle d, b, a, d\rangle$

12.33　它们都是连通的。

12.34

1. 1

2. 3

3. 6

4. 10

12.35　这一图中有如下所示的8条欧拉路径。

$\langle 1, 2, 3, 1, 4, 3\rangle$，$\langle 1, 2, 3, 4, 1, 3\rangle$，$\langle 1, 3, 2, 1, 4, 3\rangle$，$\langle 1, 3, 4, 1, 2, 3\rangle$，$\langle 3, 2, 1, 3, 4, 1\rangle$，$\langle 3, 2, 1, 4, 3, 1\rangle$，$\langle 3, 4, 1, 3, 2, 1\rangle$，$\langle 3, 4, 1, 2, 3, 1\rangle$

12.36　3是半欧拉图，4是欧拉图。

12.37　这一图中有如下给出的12条哈密尔顿路径。

$\langle 1, 2, 3, 4\rangle$，$\langle 1, 4, 3, 2\rangle$，$\langle 2, 1, 4, 3\rangle$，$\langle 2, 1, 3, 4\rangle$，$\langle 2, 3, 1, 4\rangle$，$\langle 2, 3, 4, 1\rangle$，$\langle 3, 2, 1, 4\rangle$，$\langle 3, 4, 1, 2\rangle$，$\langle 4, 1, 2, 3\rangle$，$\langle 4, 1, 3, 2\rangle$，$\langle 4, 3, 1, 2\rangle$，$\langle 4, 3, 2, 1\rangle$

12.38　2、3和4是哈密尔顿图，1不是。

12.39　是的，如果两个顶点间的两条边是平行边。

12.40　是的，如果边是一个环。

12.41　3条。

12.42

1. 有1个循环：$\langle 2, 4, 5, 2\rangle$。

2. 有3个循环，它们分别是$\langle 1, 2, 5, 1\rangle$、$\langle 1, 2, 3, 4, 1\rangle$和$\langle 1, 5, 2, 3, 4, 1\rangle$。

3. 有3个循环，它们分别是$\langle 1, 4, 5, 1\rangle$、$\langle 1, 4, 2, 3, 5, 1\rangle$和$\langle 2, 3, 5, 4, 2\rangle$。

12.43

$\neq E=3$

12.44　1和3是树，2和4不是树，因为它们包含循环。

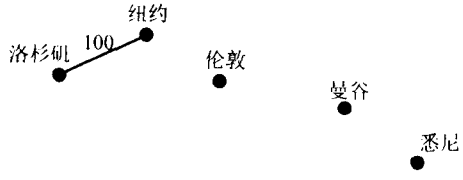
○ 答案不唯一。　译者注

12.45 假设 T 有 n 个顶点。因为 T 是树，T 有 $n-1$ 条边。从 T 中移走一条边会产生一个有 n 个顶点和 $n-2$ 条边的图。新图中边的数目不足以使其成为连通图。

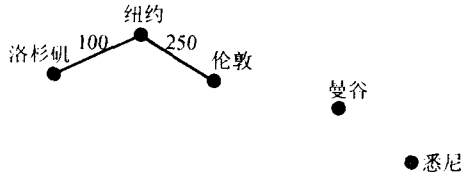
12.46 假设 T 有 n 个顶点。因为 T 是树，T 有 $n-1$ 条边。在 T 中增加一条边意味着产生一个带有 n 个顶点和 n 条边的图。新图中边的数目和顶点的数目相同，所以其中一定存在循环。

12.47 树的任意两个顶点间总是恰好有一条路径。

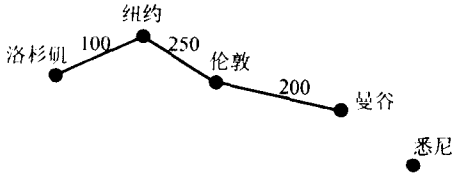
12.48 最初，没有边被标记。从洛杉矶开始，可以发现最便宜的边是连接洛杉矶到纽约的边，因此，树变成如下形式。



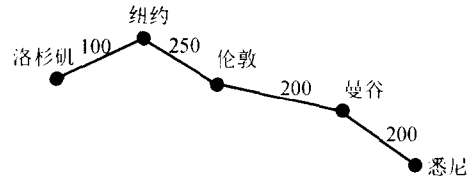
连接纽约和洛杉矶之一与伦敦、悉尼和曼谷之一的最便宜的边是连接纽约和伦敦的边。因此，把这条边加入到树中，得到如下的树。



连接纽约、洛杉矶和伦敦之一与悉尼和曼谷之一的最便宜的边是连接伦敦和曼谷的边。因此，把这条边加入到树中，得到如下的树。

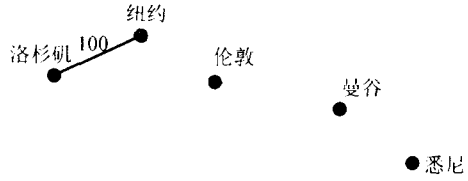


最后，连接纽约、洛杉矶、伦敦和曼谷之一与悉尼的最便宜的边是连接曼谷和悉尼的边。因此，把这条边加入到树中，得到如下的树。

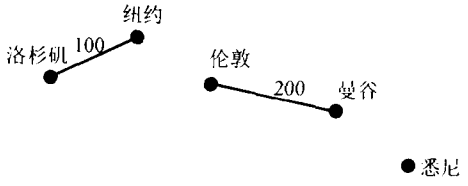


这棵树就是我们要求的最小生成树。

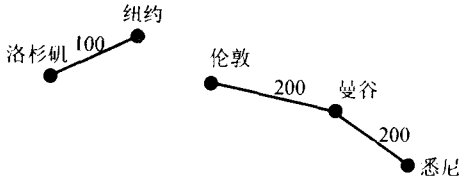
12.49 首先，最便宜的边是连接洛杉矶和纽约的边。因此，树变成如下形式。



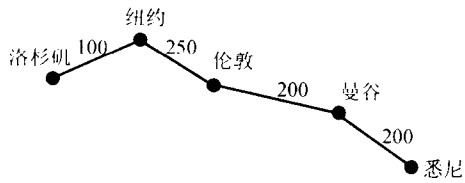
下一条最便宜的边是连接伦敦与曼谷的边或者连接曼谷与悉尼的边，这两条边中的任何一条加入到树中都不会产生循环，我们选择前者。因此，把这条边加入到树中，得到如下所示的树。



下一条最便宜的边是连接曼谷与悉尼的边，这条边加入到树中不会产生循环。因此，把这条边加入到树中，得到如下所示的树。



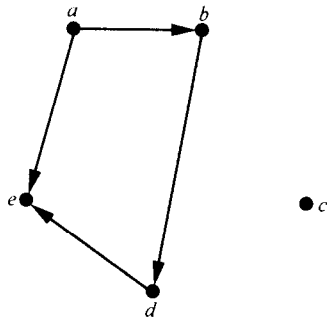
最后，下一条最便宜的边是连接伦敦和纽约的边，这条边加入到树中不会产生循环。因此，把这条边加入到树中，得到如下所示的树。



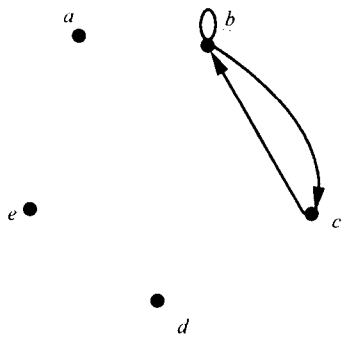
这就是我们要求的最小生成树。

12.50

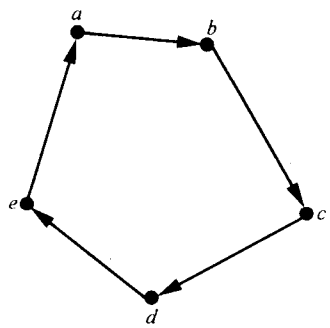
1.



2.



3.



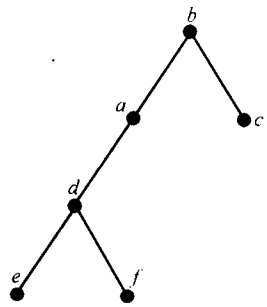
12.51

1. $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$

3. $\begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

12.52　存在若干可能的解，下面给出其中的一个。



12.53　树的高度是 3。

12.54　含有 n 个节点的二叉树的最大高度为 $n-1$。

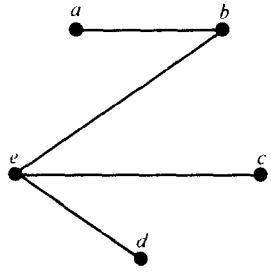
12.55　含有 n 个节点的二叉树的最小高度为 $\lceil \log_2(n+1) \rceil - 1$。

12.56

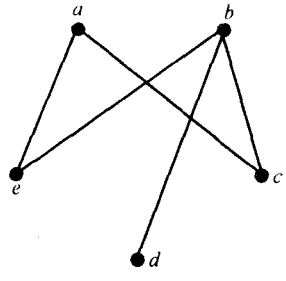
1. b，a，d，e，f，c
2. e，d，f，a，b，c
3. e，f，d，a，c，b

12.57

1.



2.



12.58

1. $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$

12.59

1. $\begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 0 & 0 & & \\ 0 & 0 & 0 & 0 & \\ 0 & 1 & 1 & 1 & 0 \end{pmatrix}$

2. $\begin{pmatrix} 0 & & & & \\ 0 & 0 & & & \\ 1 & 1 & 0 & & \\ 0 & 1 & 0 & 0 & \\ 1 & 1 & 0 & 0 & 0 \end{pmatrix}$

12.60 (V_2, E_2)表示的图有 3 个顶点，而其余图都有 4 个顶点，因此，这个图不可能与其余图同构。(V_4, E_4)表示的图有 2 条边，而其余图有 3 条边，因此，这个图不可能与其余图同构。最后，不存在从 V_1 到 V_3 的全双射 f，满足$(u, v) \in E_1$ 当且仅当$(f(u), f(v)) \in E_3$。

12.61 有如下所示的 4 个欧拉路径。

$\langle c, a, b, d, a \rangle$，$\langle c, a, d, b, a \rangle$，$\langle a, b, d, a, c \rangle$，$\langle a, d, b, a, c \rangle$

12.62 有如下所示的 4 个哈密尔顿路径。

$\langle c, a, b, d \rangle$，$\langle c, a, d, b \rangle$，$\langle b, d, a, c \rangle$，$\langle d, b, a, c \rangle$

12.63 有一个循环：$\langle a, b, d, a \rangle$。

12.64

1. g, c, a, b, f, d, e
2. a, c, b, g, d, f, e
3. a, b, c, d, e, f, g

12.65 树的高度是 2。

12.66

1. 有三棵：每一个可能的根节点构成一棵这样的树。

2. 根节点有三种可能性：a、b 或 c。如果 a 是根节点，那么它或者有左孩子，或者有右孩子(但不能同时既有左孩子又有右孩子)。子节点可以是 b，也可以是 c。因此，如果 a 是根节点，有 4 种可能的二叉树。同样，如果 b 或 c 是根节点，那么分别也有 4 种可能的二叉树。因此，共有 12 种可能的二叉树。

3. 根节点有三种可能性：a、b 或 c。如果 a 是根节点，那么它或者有两个孩子，或者只有左孩子，或者只有右孩子。如果 a 有一个孩子，那么 a 的孩子还可以有左孩子或者右孩子。因此，如果 a 是根节点，那么可能的二叉树有 5 种形状。因为在每种形状的树中，b 可以占两个空位中的任意一个，然后 c 占剩下的空位。所以，如果 a 是根节点，那么有 10 种可能的二叉树。同样，以 b 和 c 为根节点的二叉树也分别有 10 种。所以，共有 30 种可能的二叉树。

第13章 组合数学

本章讨论组合数学的话题。事实上，本章所讨论的内容是如何判定在特定场景下可能结局的数目。这种情况在计算中很常见。例如，我们可能希望知道有多少种计算机联网的方法，或者可能希望知道一组特定的运算会产生多少种可能的结果，或者可能希望通过给定算法所需的运算数目来确定这一算法的效率。

限于篇幅，本章只给出组合数学的简单介绍；读者可以从[LL92]获取更多的练习题和例子，也可以从[Bol86]、[Eri96]、[Bru99]或[Gri99]获得组合数学的更深入的介绍。

我们首先考虑组合数学学习中的基础数学函数：阶乘函数。

13.1 阶乘函数

对于本章内容的理解至关重要的阶乘函数(factorial function)的定义如下所示。对于任意的自然数 n，n 的阶乘(记作 $n!$)定义为

$$n! = 1\times 2\times 3\times \ldots \times n$$

n 的阶乘就是从 1 到 n 的所有自然数相乘的结果。另外，对于特例 $n=0$，我们约定 $0! = 1$。

当然，也可以给出如下所示的阶乘函数的递归定义。

$$0! = 1$$

$$1! = 1$$

$$n! = n\times(n-1)!，若\ n>1$$

例 13.1 3! 可以计算如下。

$$\begin{aligned} 3! &= 3\times 2! \\ &= 3\times 2\times 1! \\ &= 3\times 2\times 1 \\ &= 6 \end{aligned}$$ □

阶乘函数有若干令人愉快的性质，在以后的学习中起着重要作用的一个性质如下所示。

规则 13.1 对于任意的自然数 n，有

$$\frac{(n+1)!}{n!} = n+1$$ □

例 13.2

$$\frac{4!}{3!} = 4$$ □

这一规则的更一般的形式如下所示。

规则 13.2 对于任意的自然数 m 和 n，有

$$\frac{(n+m)!}{n!} = \prod_{i=n+1}^{n+m} i$$ □

例 13.3 如果 $n=7$，$m=3$，那么可以得到如下结果。

$$\frac{10!}{7!} = \prod_{i=8}^{10} i$$ □

练习 13.1 计算下列各题。

1. $1!$
2. $3!$
3. $5!$
4. $-1!$
5. $\frac{1001!}{1000!}$
6. $\frac{20!}{18!}$
7. $\frac{\left(\frac{20!}{19!}\right)!}{18!}$ □

练习 13.2 考虑练习 12.34。使用“!”写出一个计算含有 n 个顶点($n>2$)的非连通图的边(包括环)的最大数目的公式。假定图中没有平行边。 □

13.2 二项式系数

对于自然数 n 和 r，使得 $r \leqslant n$，二项式系数 $\binom{n}{r}$（读作 n 选 r）的定义如下所示。

$$\binom{n}{r}=\frac{n!}{(n-r)! \times r!}$$

本质上，二项式系数 $\binom{n}{r}$ 描述了从 n 个对象中取出 r 个对象时的选取种类。

例如，考虑一次包含 3 个奖品的竞赛，奖品分别是汽车、房子和驴。如果需要从这 3 个奖品中选择 2 个奖品，那么有三种可能的组合。它们是：

1. 汽车和驴
2. 房子和驴
3. 汽车和房子

因此，这一例子的二项式系数的计算如下所示。

$$\begin{aligned}\binom{3}{2}&=\frac{3!}{(3-2)! \times 2!}\\&=\frac{6}{2}\\&=3\end{aligned}$$

另一方面，如果需要从这 3 个奖品中选出 1 个奖品，那么这时的二项式系数的计算如下所示。

$$\begin{aligned}\binom{3}{1}&=\frac{3!}{(3-1)! \times 1!}\\&=\frac{6}{2}\\&=3\end{aligned}$$

再举一个例子，从 4 个奖品中选出 2 个奖品的可能组合数为

$$\begin{aligned}\binom{4}{2}&=\frac{(4-2)!}{(4-2)! \times 2!}\\&=\frac{24}{4}\\&=6\end{aligned}$$

下面是二项式系数的一个有用的性质。

规则 13.3 对于任意的自然数 n 和 r，有

$$\binom{n}{r}=\binom{n}{n-r} \qquad \square$$

例 13.4 考虑下式。

$$\binom{20}{15}$$

根据二项式系数的定义，这与下式相等。

$$\frac{20!}{(20-15)! \times 15!}$$

由算术可知，它与下式相等。

$$\frac{20!}{(20-5)! \times 5!}$$

最后，根据二项式系数的定义，它与下式相等。

$$\binom{20}{5} \qquad \square$$

下一个规则说明，只有一种从 n 个元素中选取 n 个元素的办法。

规则 13.4 对于任意的自然数 n，有

$$\binom{n}{n}=1 \qquad \square$$

例 13.5 如果从 3 个奖品中选取 3 个奖品的话，那么只有一种选取的方法：获得全部 3 个奖品。 □

接下来，只有一种从 n 个元素中选取 0 个元素的方法。

规则 13.5 对于任意的自然数 n，有

$$\binom{n}{0}=1 \qquad \square$$

例 13.6 如果要从 3 个奖品中选取 0 个奖品，那么也只有一种可能：没有得到奖品。 □

最后，二项式系数满足下面的性质。

规则 13.6 对于任意的自然数 n 和 r，使得 $n \geqslant 1$，有

$$\binom{n}{r}=\binom{n-1}{r-1}+\binom{n-1}{r} \qquad \square$$

例 13.7 考虑 $n=5$ 和 $r=3$，这里，我们希望证明下式成立。

$$\binom{5}{3}=\binom{4}{2}+\binom{4}{3}$$

由二项式系数的定义，上式的左边等于

$$\frac{5!}{2!\times 3!}$$

它又等于

$$\frac{120}{12}$$

所以，结果为 10。

现在考虑上面等式的右边。根据二项式系数的定义，它等于

$$\frac{4!}{2!\times 2!}+\frac{4!}{1!\times 3!}$$

上式又等于

$$\frac{24}{4}+\frac{24}{6}$$

上式的结果为 6+4。

因此，等式的两边相等。 □

练习 13.3 计算下列各题。

1. $\binom{0}{0}$

2. $\binom{4}{4}$

3. $\binom{4}{3}$

4. $\binom{4}{1}$

5. $\binom{4}{2}$

6. $\binom{1001}{1000}$ □

练习 13.4 证明规则 13.3。 □

练习 13.5 证明规则 13.4。 □

练习 13.6 证明规则 13.5。 □

练习 13.7 帕斯卡三角形（Pascal's triangle）[1]是一种表示二项式系数的方法，它基于规则 13.6。帕斯卡三角形的前三行如下所示。

```
      1
    1   1
  1   2   1
```

第一行表示 $n=0$，第二行表示 $n=1$，第三行表示 $n=2$。另外，每一行的第一个元素表示 $r=0$，第二个元素表示 $r=1$，第三个元素表示 $r=2$，以此类推。因此，这一表中的第一个条目就表示"从 0 个中取 0 个"，第二行的各条目分别表示"从 1 个中取 0 个"和"从 1 个中取 1 个"。另外，可以看到从两个元素中取一个元素有两种方法。这一条目可以由上面相邻两个元素相加得到（注意，这一三角形的结构是基于规则 13.6 的）。除了每一行的两个端点之外，这个规则对所有条目都适用，端点的条目总为 1（这些条目分别表示 $r=0$ 和 $r=n$）。因此，例如，在这一三角形上再加上一行，将得到下面的帕斯卡三角形。

```
        1
      1   1
    1   2   1
  1   3   3   1
```

继续给出帕斯卡三角形的下三行。 □

练习 13.8 考虑练习 13.7 中的帕斯卡三角形，对于下列各 n 和 r 的数对，给出帕斯卡三角形中的相应条目。

1. $n=3$，$r=2$
2. $n=4$，$r=2$
3. $n=5$，$r=3$
4. $n=6$，$r=3$ □

练习 13.9 参考练习 13.7 中的帕斯卡三角形，计算下列各题。

1. $\binom{7}{2}$

---

[1] 以 Blaise Pascal(1623—1662)的名字命名。

2. $\binom{7}{3}$

3. $\binom{7}{4}$

4. $\binom{7}{5}$ □

13.3 计数

计数的基本原理就是假设有一系列事件，每个事件都有若干种不同的产生方式。例如，假设考虑三个事件 e_1、e_2 和 e_3。e_1 有 n_1 种不同的产生方式，e_2 有 n_2 种不同的产生方式，e_3 有 n_3 种不同的产生方式。那么，事件 e_1、e_2、e_3 的序列可以有 $n_1 \times n_2 \times n_3$ 种不同的产生方式。

例 13.8 假设要投掷硬币，然后再投骰子，然后再从一摞牌中随机抽取一张牌。第一个事件有 2 种可能结果，第二个事件有 6 种可能结果，第三个事件有 52 种可能结果。因此，这个事件序列的可能结果有 $2 \times 6 \times 52 = 624$ 种。 □

值得注意的是，这里事件彼此独立这一事实是十分重要的：掷出一个正面并不会改变投出 6 点骰子的可能性，同样投出 3 点骰子也不会增加抽出红心 Q 的可能性。

计数原理与笛卡儿积的元素数目有着密切的关系。回想一下，给定两个集合 X 和 Y，笛卡儿积 $X \times Y$ 的定义如下所示。

$$X \times Y = \{x : X;\ y : Y \cdot (x, y)\}$$

再回想一下，这样的笛卡儿积的势由下式给出。

$$\#(X \times Y) = (\#X) \times (\#Y)$$

例如，给定下列集合。

$Bread = \{$棕色，白色$\}$

$Filling = \{$火腿，吉士，吉士火腿$\}$

有

$$\#(Bread \times Filling) = 2 \times 3 = 6$$

因此，存在 6 种可能的结果。

在更一般的意义下，给定集合 X_1，X_2，…，X_n，有

$$\#(X_1 \times X_2 \times \cdots \times X_n) = \prod_{i=1}^{n} (\#X_i)$$

练习 13.10 假设在一个特定的计算机系统中，用户的默认电子邮箱密码是以 DDMMYY 的形式给出的用户的生日。这个系统能够生成的不同密码的最大数目是多少？ □

练习 13.11 再次考虑练习 13.10 中的系统，如果知道所有用户都是在某个四年间出生的，那么这个系统能够生成多少个不同的密码？ □

练习 13.12 假设在另一个计算机系统中，用户的默认电子邮箱的密码是 8 位随机字符串，字符可以是 $A..Z$、$a..z$、$0..9$ 中的一个，而且开头的字符不能是数字。那么，这个系统能够生成的密码的最大数目是多少？ □

练习 13.13 一家三明治店提供三种不同类型的夹馅：鸡蛋、鸡肉和金枪鱼，还提供三种不同类型的面包：全麦面包、黑麦面包和麦壳面包。这家三明治店能卖出多少种不同类型的三明治？ □

练习 13.14 参考练习 13.13 中的三明治店，如果一个人不吃动物肉类食品，那么他能吃的三明治有多少种(假设金枪鱼是动物)？ □

练习 13.15 在一次网球联赛中，有 64 名选手参加男子比赛，64 名选手参加女子比赛。那么有多少种冠军的组合？ □

13.4 排列

上节所述的计数所关心的是确定一系列事件的不同产生方式的数目，其中，假设事件彼此相互独立。本节我们讨论排列，这里不再假设事件相互独立，主要关心有一定关

联的对象或事件有多少种排序方法。

例如，假设有如下定义的集合 *Colour*。

Colour={红色，黄色，绿色}

如果要考虑这些元素有多少种排序方法，也就是说，关心的是 *Colour* 的元素的排列，那么有下列可能的排列结果。

红色，黄色，绿色
红色，绿色，黄色
黄色，红色，绿色
黄色，绿色，红色
绿色，红色，黄色
绿色，黄色，红色

这里，显然不满足事件独立的原理：如果选取红色为第一位，那么它将影响下面的选取，即接下来就不能选择红色了。类似地，如果选取黄色为第一位，那么黄色就不能出现在第二或第三个位置上。因此，可能的排序数由下式给出。

$$3\times2\times1=6$$

显然，结果就是 3!。

给定 n 个对象的集合，这些对象的任意编排称为排列(permutation)。排列的第一个对象是 n 个对象中的一个；第二个对象是 $n-1$ 个对象中的一个；第三个对象是 $n-2$ 个对象中的一个；以此类推。因此，排列的所有可能的数目由下式给出。

$$n\times(n-1)\times(n-2)\times\cdots\times1$$

因此，n 个对象的排列数目是 $n!$。

给定 n 个对象的集合 X，可以使用序列形式地定义 X 的排列的集合。

$$\{s : \mathrm{iseq}\ X \mid \#s=\#X\}$$

例 13.9 假设有如下给出的集合 *Friends*。

Friends={理查德，乔治，约翰}

那么，*Friends* 的排列的集合如下所示。

{⟨理查德，乔治，约翰⟩，⟨理查德，约翰，乔治⟩，⟨乔治，理查德，约翰⟩，⟨乔治，约翰，理查德⟩，⟨约翰，理查德，乔治⟩，⟨约翰，乔治，理查德⟩}

这里，有 6 种可能的排列(即 3! =6)。 □

任何 $r\leqslant n$ 个对象的任意编排称为 r-排列。n 个对象的 r-排列的数目记作 $P(n, r)$，其定义如下。

$$P(n,\ r)=\frac{n!}{(n-r)!}$$

同样，可以使用序列如下形式地定义集合 X 的 r-排列的集合。

$$\{s : \mathrm{iseq}\ X \mid \#s=r\}$$

例 13.10 再次考虑如下给出的集合 *Colour*。

Colour={红色，黄色，绿色}

如果希望考虑 *Colour* 中的两种颜色的排列，那么将得到如下结果。

{⟨红色，黄色⟩，⟨红色，绿色⟩，⟨黄色，红色⟩，⟨黄色，绿色⟩，⟨绿色，红色⟩，⟨绿色，黄色⟩}

这里，有

$$\begin{aligned}P(3,\ 2)&=\frac{3!}{(3-2)!}\\&=\frac{6}{1}\\&=6\end{aligned}$$

□

例 13.11 下式给出含有 10 个元素的集合的 3-排列的数目。

$$\begin{aligned}P(10,\ 3)&=\frac{10!}{(10-3)!}\\&=\frac{10!}{7!}\\&=10\times9\times8\\&=720\end{aligned}$$

□

练习 13.16 字母 A、B、C、D、E 的排列有多少种？ □

练习 13.17

1. 字母 A、B、C、D、E 中的 2 个字母的排列有多少种？

2. 字母 A、B、C、D、E 中的 3 个字母的排列有多少种？

3. 字母 A、B、C、D、E 中的 4 个字母的排列有多少种？ □

练习 13.18 一次宴会上有 12 瓶酒，要喝 8 瓶，那么有多少种排列？ □

练习 13.19 一场赛马中有 10 匹马参加比赛，前三名有奖，获奖的排列有多少种？ □

练习 13.20 一次宴会上有 10 把椅子可用，假设有 5 个人参加宴会，那么座次的编排有多少种排列？ □

练习 13.21 8 个选手参加竞赛，只有 3 个人能获得不同的奖，那么获奖者的排列有多少种？ □

13.5 组合

在 13.4 节中，我们看到集合{红色，黄色，绿色}有 6 种排列，在那里，顺序是很重要的，例如，〈红色，黄色，绿色〉与〈绿色，黄色，红色〉是不同的排列。另一方面，给定集合的元素的组合(combination)是一种与顺序无关的对象的选取：一个组合就是一组对象，与对象的顺序无关。例如，红色、黄色和黄色、红色是取自集合{红色，黄色，绿色}的等价组合。组合是本节的主题。

n 个对象的集合的 r-组合是该集合的一个 r-子集。n 个对象的集合的 r-组合的数目记作 $C(n, r)$，其定义如下所示。

$$C(n, r)=\binom{n}{r}$$

与使用序列定义集合 X 的排列一样，可以使用集合如下定义 r-组合。

$$\{s : \subseteq X \mid \#s=r\}$$

例 13.12 集合{红色，黄色，绿色}的 2-组合的集合如下所示。

{{红色，黄色}，{红色，绿色}，{黄色，绿色}}

另外，有

$$C(3, 2)=\binom{3}{2}$$

$$=\frac{3!}{1!\times 2!}$$

$$=3$$ □

例 13.13 含有 10 个元素的集合的 3-组合的数目的计算如下。

$$C(10, 3)=\binom{10}{3}$$

$$=\frac{10!}{7!\times 3!}$$

$$=\frac{10\times 9\times 8}{3!}$$

$$=120$$ □

练习 13.22 证明下式成立。

$$C(n, r)=\frac{P(n, r)}{r!}$$ □

练习 13.23 一副标准的扑克牌有 52 张牌，从这样的一副牌中选取下列牌数时，各有多少种组合？

1. 51
2. 50
3. 49 □

练习 13.24 一个口袋中有 6 个不同颜色的球。如果从中取出 3 个球，那么有多少种组合？ □

练习 13.25 考虑字母的集合{A，B，C，D，E}。

1. 这个集合的 2 个字母的组合有多少种？
2. 这个集合的 3 个字母的组合有多少种？
3. 这个集合的 4 个字母的组合有多少种？ □

练习 13.26 一次宴会上有 12 瓶酒，要喝 8 瓶，有多少种组合？ □

练习 13.27 一次赛马中有 10 匹马参加比赛，前三名有奖，获奖的组合有多少种？ □

练习 13.28 一次宴会上有 10 把椅子，有 5 个人参加宴会，就座情况有多少种组合？ □

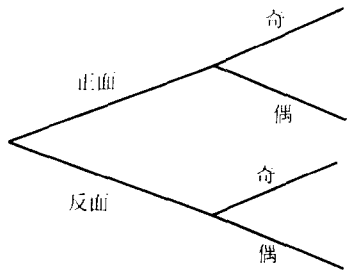
练习 13.29 有 8 个人参加竞赛，只有 3 个人能获得不同的奖，有多少种获奖者的

组合？ □

13.6 树形图

树形图可以使我们形象地列举出一系列事件产生的不同可能性(假设每个事件都有有限种产生方式)。

例如，考虑一个掷硬币并选取一个非零自然数的游戏，在此所关心的是所选取的自然数的奇偶性。这一游戏的可能结果如下所示。



这里，有四种可能的结果：掷出正面且选取奇数；掷出正面且选取偶数；掷出反面且选取奇数；掷出反面且选取偶数。这些结果可以用树形图来表示。

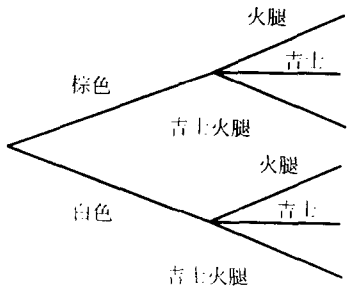
注意，我们并不关心这些不同结果间的相似性，只是列举出不同的可能性。还要注意，我们只对有限个事件构造树形图，而且每个事件都有有限种产生方式。

例 13.14 集合 *Bread* 和 *Filling* 的定义如下。

Bread={棕色，白色}

Filling={火腿，吉士，吉士火腿}

不同的面包和不同的夹馅产生不同的三明治组合，如下所示。



□

利用计数可以确定一系列相互独立的事件的可能结果的数目，而利用树形图能够可视化地表示这些结果。

练习 13.30 考虑一个从一副标准的扑克牌中抽取一张牌，然后再掷一枚硬币的游戏。如果抽取的牌是红心或黑桃而且掷出的是硬币的正面，那么玩家就获胜，否则玩家就输掉。用树形图表示可能的结果。 □

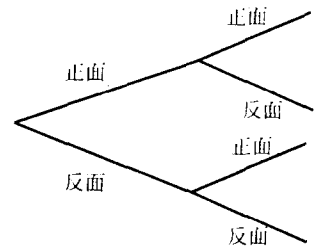
练习 13.31 考虑一个别致的双人四脚游戏。如果一个选手双腿领先，那么他就获胜。还有，如果比分是 2-2，那么平局。用树形图来列举这一游戏的可能结果。 □

练习 13.32 戴夫周五晚上或者去参加派对，或者待在家里。另外，周六晚上，他或者去看电影，或者去下饭馆，或者待在家里。用树形图来表示戴夫周末活动的可能性。

□

13.7 取样

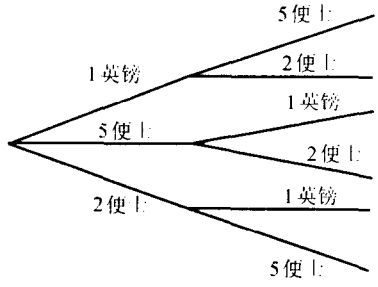
当考虑树形图时，我们倾向于考虑一系列相互独立的事件。例如，在三明治的例子中，夹馅的选取不受面包选取的影响。另外，掷硬币的结果也不影响抽取扑克牌的结果(也不受抽取扑克牌的结果的影响)。这些例子是计算机建模中比较典型的。然而，在某些场合，我们或许希望考虑一系列事件，这些事件中的某一个事件可以被一次又一次地重复。练习 13.31 的游戏就是这样的一个例子。下面再举一个例子，考虑下面的树形图。



这里，掷硬币这一事件发生两次。当然，第二次掷硬币的结果与第一次掷硬币的结果是

独立的。

现在我们考虑一个游戏：玩家从 1 英镑硬币、5 便士硬币、2 便士硬币中选取两枚硬币。可以如下表示这一游戏的可能结果。



这里，尽管第二个事件(随机选取硬币)与第一个事件相同，但是第二个事件的可能结果直接受到第一个事件的结果的影响。例如，第一次选取了 1 英镑硬币，那么第二次就不能再选取这枚硬币。

一系列事件的可能结果的集合称为事件的取样空间(sample space)。有两种取样形式：放回取样(sampling with replacement)和非放回取样(sampling without replacement)。下面用一个例子来说明它们之间的区别。

考虑一个含有三个球的口袋：一个红球(记作 r)，一个黄球(记作 y)，一个绿球(记作 g)。如果从口袋中取出第一个球，然后在不放回第一个球的情况下再取出第二个球，那么有六种可能结果：rg，ry，yr，yg，gr，gy。这是非放回取样的一个例子，可能结果的数目由 $P(n, r)$给出。另一种计算这种可能性的方法是，考虑第一个球有三种取法，第一个球确定后，第二个球就只有两种可能取法了，因此，有 $3\times2=6$ 种可能的取法。显然，这与 $P(3, 2)$相等。

现在考虑相同的场景，但是，在取出第一个球之后要将其放回到口袋中。这里有 9 种可能的结果：rr，rg，ry，gr，gg，gy，yr，yg，yy。这就是放回取样的一个例子，可能结果的数目由 n^r 给出。当然，这等价于考虑第一个球有三种可能的取法，第二个球也有三种可能的取法，因此有 $3\times3=9$ 种可能的取法。

因此，给定 n 个对象的取样空间，如果在非放回取样的情况下进行 r 次取样，那么可能结果的数目就是 $P(n, r)$。另一方面，如果在放回取样的情况下进行 r 次取样，那么可能结果的数目就是 n^r。

例 13.15 在放回取样的情况下，如果从一副牌中随机抽取三张牌，那么可能结果的数目由下式给出。

$$52\times52\times52=140\ 608$$

另一方面，在非放回取样的情况下，如果从一副牌中随机抽取三张牌，那么可能结果的数目由下式给出。

$$52\times51\times50=132\ 600$$ □

练习 13.33 一个委员会由一位主席、一位副主席、一位财务总管、一位秘书组成。现在，要从 30 个人中抽取这些职务的人选，每个人只能担任一个职位，那么，这一委员会的组成有多少种可能性？ □

练习 13.34 假设有练习 13.33 所描述的委员会。现在允许兼任，但是主席不能兼任，那么这一委员会的组成有多少种可能性？ □

练习 13.35 考虑上面练习 13.33 中的委员会。现在假设任何人都可以兼任，那么这一委员会的组成有多少种可能性？ □

练习 13.36 考虑字母的集合$\{A, B, C, D, E\}$。

1. 在非放回取样的情况下，从这一集合中选取 2 个字母，有多少种方法？

2. 在非放回取样的情况下，从这一集合中选取 3 个字母，有多少种方法？

3. 在非放回取样的情况下，从这一集合中选取 4 个字母，有多少种方法？ □

练习 13.37 考虑字母的集合$\{A, B, C, D, E\}$。

1. 在放回取样的情况下，从这一集合

中选取 2 个字母，有多少种方法？

2. 在放回取样的情况下，从这一集合中选取 3 个字母，有多少种方法？

3. 在放回取样的情况下，从这一集合中选取 4 个字母，有多少种方法？ □

13.8 附加练习

练习 13.38 计算下列各题。

1. $\binom{6}{2}$

2. $\binom{10}{8}$

3. $\binom{17}{14}$

练习 13.39 在练习 13.7 中，我们考虑了 0～6 行的帕斯卡三角形，写出第 7、8 和 9 行的值。

练习 13.40 考虑集合 X、Y 和 Z，其中，$\#X=4$，$\#Y=5$，$\#Z=10$，计算 $X\times Y\times Z$ 的势。

练习 13.41 考虑一把由四位数字组成的密码锁，每个数字的取值范围是 0～9。这把密码锁能提供多少个密码？

练习 13.42 假设某个国家的车牌号是由一个字母后面跟随 3 个数字后面再跟随 3 个字母组成的。这种类型的车牌号有多少种？

练习 13.43 假设某个国家的邮政编码是由两个字母后面跟随两个数字后面再跟随两个字母组成的。这种类型的邮政编码有多少种？

练习 13.44 计算下列各题。

1. $P(8, 4)$

2. $P(10, 4)$

3. $P(12, 4)$

练习 13.45 下列单词中的字母可以有多少种组织方式？

1. BASE

2. FOOT

3. BALL

4. FOOTBALL

5. BASEBALL

练习 13.46 计算下列各题。

1. $C(8, 4)$

2. $C(10, 4)$

3. $C(12, 4)$

练习 13.47 一次运动会有 6 个队参加，把这 6 个队分成两组，每组 3 个队，有多少种组合？

练习 13.48 一次测验有 35 个问题，而试题库中共有 50 个问题，问有多少种问题组合？

练习 13.49 一次测验有 35 个问题，而试题库中共有 50 个问题，问有多少种问题排列？

练习 13.50 集合 *menu* 的组成是：两道开胃菜(汤和面包)；两道主食(意大利面条和鱼)；两道甜点(冰淇淋和奶油蛋糕)。画出表示所有可能的食谱组合的树形图。

练习 13.51 一个性格测试要求应试者从 10 种性格特点中选择 3 种最适合自己的性格特点，有多少种组合？

练习 13.52 一个性格测试要求应试者从 10 种性格特点中选择 3 种最适合自己的性格特点，有多少种排列？

练习 13.53 考虑数字 4、7 和 9。

1. 在放回取样的情况下，这些数字可以形成多少种两位数？

2. 在非放回取样的情况下，这些数字可以形成多少种两位数？

13.9 练习解答

13.1

1. 1

2. 6

3. 120

4. 未定义

5. 1001

6. 380

7. 380

13.2 首先，考虑可能的环的数目。如果图有 n 个顶点，那么它最多只能有 n 个环。

其次，考虑一个非连通图的边的最大数目。为了得到这一数目，我们可能会想到“最连通的非连通图”，这是一个除了一个孤立的顶点以外，所有其余顶点对都相邻的图。例如，如果 $n=4$，那么有 3 个顶点是彼此相邻的(共有 3 条边)。另外，如果 $n=5$，有 4 个顶点是彼此相邻的(共有 6 条边)。因此，公式如下所示。

$$\frac{(n-1)!}{2}+n$$

13.3

1. $\binom{0}{0}=1$

2. $\binom{4}{4}=1$

3. $\binom{4}{3}=\frac{4!}{1!\times 3!}$
$=4$

4. $\binom{4}{1}=\frac{4!}{3!\times 1!}$
$=4$

5. $\binom{4}{2}=\frac{4!}{2!\times 2!}$
$=6$

6. $\binom{1001}{1000}=\frac{1001!}{1!\times 1000!}$
$=1001$

13.4 考虑 $\binom{n}{r}$，它与下式相等。

$$\frac{n!}{(n-r)!\times r!}$$

因为乘法满足交换律，因此，它与下式相等。

$$\frac{n!}{r!\times(n-r)!}$$

而上式与下式相等。

$$\frac{n!}{(n-(n-r))!\times(n-r)!}$$

显然，它与下式相等。

$$\binom{n}{n-r}$$

13.5 考虑某个自然数 n。在这里，有

$$\binom{n}{n}=\frac{n!}{(n-n)!\times n!}$$
$$=\frac{n!}{0!\times n!}$$
$$=\frac{n!}{1\times n!}$$
$$=1$$

因此，有

$$\binom{n}{n}=1$$

13.6 考虑某个自然数 n。在这里，有

$$\binom{n}{0}=\frac{n!}{(n-0)!\times 0!}$$
$$=\frac{n!}{n!\times 1}$$
$$=1$$

因此，有

$$\binom{n}{0}=1$$

13.7

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
1 6 15 20 15 6 1

13.8

1. 3

2. 6

3. 10

4. 20

13.9

1. 21

2. 35

3. 35

4. 21

13.10 假设每个月最多有31天，每年有12个月，而年的取值范围在00～99，那么有

$$31\times12\times100=37\ 200$$

种可能的组合。

13.11 假设在四年中进行选择，那么有

$$31\times12\times4=1\ 488$$

种可能的组合。

13.12 可能的组合数为

$$52\times62^7=183\ 123\ 959\ 522\ 816$$

13.13 可能的组合数为

$$3\times3=9$$

13.14 可能的组合数为

$$0\times3=0$$

13.15 可能的组合数为

$$64\times64=4\ 096$$

13.16 可能的组合数为

$$P(5,\ 5)=\frac{5!}{(5-5)!}=\frac{120}{1}=120$$

13.17

1. $P(5,\ 2)=\frac{5!}{(5-2)!}=\frac{120}{6}=20$

2. $P(5,\ 3)=\frac{5!}{(5-3)!}=\frac{120}{2}=60$

3. $P(5,\ 4)=\frac{5!}{(5-4)!}=\frac{120}{1}=120$

13.18

$$P(12,\ 8)=\frac{12!}{(12-8)!}=\frac{12!}{4!}=19\ 958\ 400$$

13.19

$$P(10,\ 3)=\frac{10!}{(10-3)!}=\frac{10!}{7!}=720$$

13.20

$$P(10,\ 5)=\frac{10!}{(10-5)!}=\frac{10!}{5!}=30\ 240$$

13.21

$$P(8,\ 3)=\frac{8!}{(8-3)!}=\frac{8!}{5!}=336$$

13.22

$$C(n,\ r)=\binom{n}{r}=\frac{n!}{(n-r)!\times r!}=\frac{n!}{(n-r)!}\times\frac{1}{r!}=P(n,\ r)\times\frac{1}{r!}=\frac{P(n,\ r)}{r!}$$

13.23

1. $C(52,\ 51)=\binom{52}{51}$

$$=\frac{52!}{1!\times 51!}$$
$$=52$$

2. $C(52,\ 50)=\binom{52}{50}$

$$=\frac{52!}{2!\times 50!}$$
$$=\frac{52\times 51}{2}$$
$$=1\ 326$$

3. $C(52,\ 49)=\binom{52}{49}$

$$=\frac{52!}{3!\times 49!}$$
$$=\frac{52\times 51\times 50}{6}$$
$$=22\ 100$$

13.24

$$C(6,\ 3)=\binom{6}{3}$$
$$=\frac{6!}{3!\times 3!}$$
$$=\frac{720}{36}$$
$$=20$$

13.25

1. $C(5,\ 2)=\binom{5}{2}$

$$=\frac{5!}{3!\times 2!}$$
$$=10$$

2. $C(5,\ 3)=\binom{5}{3}$

$$=\frac{5!}{2!\times 3!}$$
$$=10$$

3. $C(5,\ 4)=\binom{5}{4}$

$$=\frac{5!}{1!\times 4!}$$
$$=5$$

13.26

$$C(12,\ 8)=\binom{12}{8}$$
$$=\frac{12!}{4!\times 8!}$$
$$=495$$

13.27

$$C(10,\ 3)=\binom{10}{3}$$
$$=\frac{10!}{7!\times 3!}$$
$$=120$$

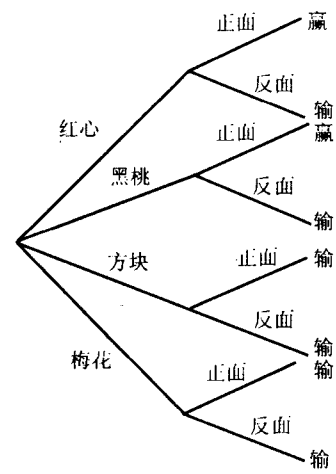
13.28

$$C(10,\ 5)=\binom{10}{5}$$
$$=\frac{10!}{5!\times 5!}$$
$$=252$$

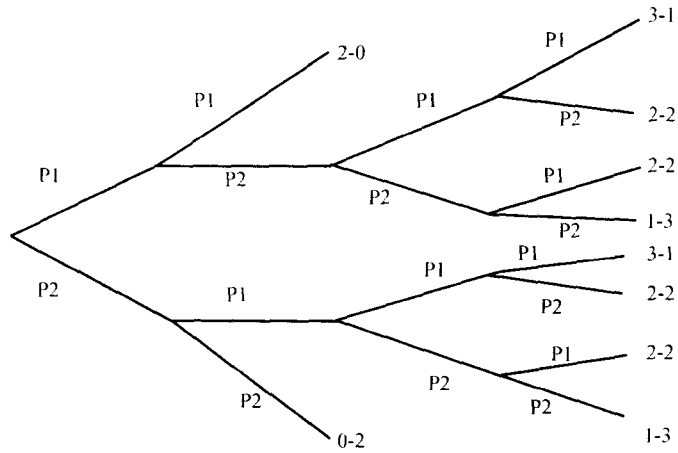
13.29

$$C(8,\ 5)=\binom{8}{5}$$
$$=\frac{8!}{3!\times 5!}$$
$$=56$$

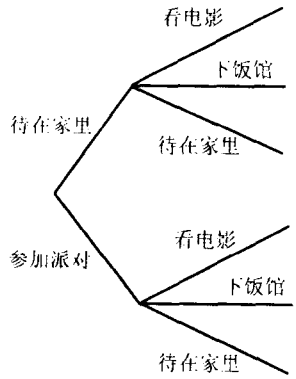
13.30



13.31



13.32



13.33

$30\times29\times28\times27=657\ 720$

13.34

$30\times29\times29\times29=731\ 670$

13.35

$30\times30\times30\times30=810\ 000$

13.36

1. $5\times4=20$
2. $5\times4\times3=60$
3. $5\times4\times3\times2=120$

13.37

1. $5\times5=25$
2. $5\times5\times5=125$
3. $5\times5\times5\times5=625$

13.38

1. $\binom{6}{2}=\frac{6!}{4!\times2!}$
$=15$

2. $\binom{10}{8}=\frac{10!}{2!\times8!}$
$=45$

3. $\binom{17}{14}=\frac{17!}{3!\times14!}$
$=680$

13.39 帕斯卡三角形的第 7 行如下所示。

1 7 21 35 35 21 7 1

帕斯卡三角形的第 8 行如下所示。

1 8 28 56 70 56 28 8 1

帕斯卡三角形的第 9 行如下所示。

1 9 36 84 126 126 84 36 9 1

13.40 $X\times Y\times Z$ 的势由 $4\times5\times10$ 给出，它等于 200。

13.41 可能的密码数为

$10^4=10\ 000$

13.42 可能的车牌号数目为

$26\times10^3\times26^3=456\ 976\ 000$

13.43 可能的邮政编码的数目为

$26^2\times10^2\times26^2=45\ 697\ 600$

13.44

1. $P(8,\ 4)=\frac{8!}{(8-4)!}$

$$=\frac{8!}{4!}$$
$$=8\times7\times6\times5$$
$$=1\ 680$$

2. $P(10,\ 4)=\frac{10!}{(10-4)!}$
$$=\frac{10!}{6!}$$
$$=10\times9\times8\times7$$
$$=5\ 040$$

3. $P(12,\ 4)=\frac{12!}{(12-4)!}$
$$=\frac{12!}{8!}$$
$$=12\times11\times10\times9$$
$$=11\ 880$$

13.45

1. 有 4! 种组织这些字母的方法，它等于 24。

2. 有 4! 种组织这些字母的方法，但是它们中两两一样(交换两个 O 的位置对所生成的字母序列不产生影响)，因此，这些字母有

$$\frac{4!}{2!}=12$$

种不同的组织方法。

3. 同样，这些字母有 12 种不同的组织方法。

4. 这些字母有 8! 种组织方法。但是，这些字母中有两个 L 和两个 O，交换两个 L 或者交换两个 O 的位置对所生成的字母序列不产生影响。因此，这些字母有

$$\frac{8!}{2!\times2!}=10\ 080$$

种不同的组织方法。

5. 这些字母有 8! 种组织方法。但是，这些字母中有两个 B、两个 A 和两个 L，交换两个 B、两个 A 或者两个 L 的位置对所生成的字母序列不产生影响。因此，这些字母有

$$\frac{8!}{2!\times2!\times2!}=5\ 040$$

种不同的组织方法。

13.46

1. $C(8,\ 4)=\binom{8}{4}$
$$=\frac{8!}{4!\times4!}$$
$$=70$$

2. $C(10,\ 4)=\binom{10}{4}$
$$=\frac{10!}{6!\times4!}$$
$$=210$$

3. $C(12,\ 4)=\binom{12}{4}$
$$=\frac{12!}{8!\times4!}$$
$$=495$$

13.47 不同组合的数目由下式给出。

$$\binom{6}{3}=\frac{6!}{3!\times3!}$$
$$=\frac{720}{36}$$
$$=20$$

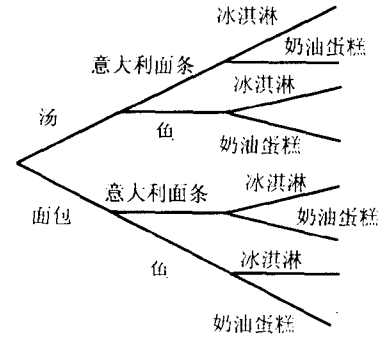
13.48 不同组合的数目由下式给出。

$$\binom{50}{35}=\frac{50!}{15!\times35!}$$

13.49 不同组合的数目由下式给出。

$$\frac{50!}{(50-35)!}=\frac{50!}{15!}$$

13.50



13.51 组合数由下式给出。

$$C(10,\ 3)=\binom{10}{3}$$

$$=\frac{10!}{7!\times 3!}$$

$$=\frac{10\times 9\times 8}{3!}$$

$$=120$$

13.52 排列数由下式给出。

$$P(10,3)=\frac{10!}{7!}$$

$$=10\times 9\times 8$$

$$=720$$

13.53

1. 在放回取样的情况下，有 $3\times 3=9$ 种可能性。

2. 在非放回取样的情况下，有 $3\times 2=6$ 种可能性。

第14章 应用实例

本章使用若干实例来展示如何将前面所述的技术应用于各种类型的系统的建模以及推理。这些例子大小、复杂度不同，但是都利用了"抽象"的威力。我们将重点放在进行建模所需考虑的方方面面。

14.1 程序变量的建模

在高级程序设计语言中，变量用于保存变化的对象的值。例如，在特定的应用中 *current_temperature* 的值也许被初始化为21，但是这个值可能随时发生变化；当然，如果程序认为当前的温度足够重要，那么我们就会希望变量的值根据实际的值而变化。而如果这一程序的可靠运行依赖于当前温度的精确表示，那么我们就会希望随时更新它的值。

可以考虑包含所有可能的变量名的类型 *VARIABLE*。为简单起见，这里只考虑类型为$\mathbb{N}$的变量，即$\mathbb{N}$包含变量可以取的所有可能值。那么，可以如下定义函数 *value*。

value: $VARIABLE \rightarrow \mathbb{N}$

我们假设，只有所考虑的程序中声明了 *VARIABLE* 的元素才出现在 *value* 的定义域中。

另外，我们还假定所有变量都初始化为0。

一种程序设计语言

在此，我们使用含有以下结构的一种小型程序设计语言。

- *declare*：可以使用这一语句声明变量，每个变量在使用之前必须加以声明。例如，*declare*(x)声明变量 x。
- *write*：使用特定值来更新特定变量的内容。例如，*write*(x, 4)表明用4来更新变量 x 的内容。
- *read*：读取特定变量的值。例如，*read*(x)返回 x 所关联的值。
- *swap*：交换两个变量的内容。例如，*swap*(x, y)交换 x 和 y 的内容。

另外，我们分别用 *begin* 和 *end* 来标识程序的开始和结尾。

练习 14.1 函数 *value* 的初始值是什么？ □

练习 14.2 形式地说明语句 *declare*(x)对 *value* 所产生的作用(假设在此之前没有声明 x)。 □

练习 14.3 下面的语句对于 *value* 可能产生什么影响？

1. *write*(x, 4)
2. *write*(x, *read*(y))
3. *swap*(x, y) □

练习 14.4 给出函数 *value* 在下面程序的每个语句后的值。

1. *begin*
2. *declare*(x)
3. *declare*(y)
4. *declare*(z)
5. *write*(x, 1)
6. *write*(y, 2)
7. *write*(z, 3)
8. *swap*(x, y)
9. *swap*(y, z)
10. *end* □

练习 14.5 使用命题逻辑表示下列陈述。

1. x 的值与 y 的值相同。

2. 若 x 的值为 3，那么 y 的值为 2。

3. 若 x 的值和 y 的值均为 0，那么一定没有声明过 z。

4. 若只声明了 x 和 y，那么 y 的值不能大于 x 的值。

5. x 和 y 相等当且仅当它们的值都为 0。 □

练习 14.6 使用谓词逻辑表示下列陈述。

1. 所有值不为 0 的已声明变量的值都至少为 10。

2. 所有声明了的变量都有不同的值。

3. 至少有一个值为 0 的已声明变量。

4. 刚好有一个值大于 x 的已声明变量。

5. 若存在一个值不为 0 的已声明变量，那么所有已声明变量的值都不为 0。 □

练习 14.7 通过公理定义来定义一个序列 *ordered*，*ordered* 列出程序中使用的所有变量，使它们按在程序中的取值成递减顺序排列。 □

练习 14.8 定义递归函数 *add*，*add* 给出 *ordered* 中的所有变量的值的和。 □

练习 14.9 定义递归函数 *double*，*double* 产生一个序列，该序列的元素依次为 *ordered* 的各元素的值的两倍。因此，例如，若

$$value=\{x\mapsto 3,\ y\mapsto 2,\ z\mapsto 1\}$$

那么

$$double\langle x,\ y,\ z\rangle=\langle 6,\ 4,\ 2\rangle$$ □

练习 14.10 使用结构归纳法证明下式成立。

$$add\ (double\ \ ordered)\ =\ 2\ \times\ (add\ \ ordered)$$ □

14.2 元搜索引擎

对于所有使用万维网的人来说，搜索引擎是众所周知的概念：目前流行的 Yahoo、Lycos、Alta Vista 等网站接受用户的查询，处理查询，然后返回包含几百、几千甚至是几百万个与查询相关的万维网网页的链接列表。元搜索引擎则做的更多：元搜索引擎接受查询，将查询提交给一系列搜索引擎，整理各搜索引擎返回的结果，对这些结果按某些标准进行排列，并将元搜索的结果提交给用户。这样的系统是理想的形式建模系统。

下面，我们假设 *URL* 表示所有网站地址的集合，而 *ENGINE* 表示所有搜索引擎的集合。

14.2.1 使用集合建模

考虑下面的情景。元搜索引擎将查询提交给了三个不同的搜索引擎：*S1*、*S2* 和 *S3*，并得到下面的结果。

S1	S2	S3
www. xyz. com	www. abc. com	www. xyz. com
www. lmn. com	www. fgh. com	www. lmn. com
www. abc. com	www. xyz. com	
www. fgh. com		

当然可以用集合来表示每个搜索结果。搜索引擎 *S1* 返回 4 个结果，*S2* 返回 3 个结果，*S3* 返回 2 个结果。可以如下用集合来表示这一信息。

$S1$ = { www. xyz. com， www. lmn. com， www. abc. com， www. fgh. com}

$S2$ = { www. abc. com， www. fgh. com， www. xyz. com}

$S3$={www. xyz. com， www. lmn. com}

现在考虑将什么提交给用户。可以决定只提交列在同时出现于 *S1*、*S2* 和 *S3* 中的网址，这是三个集合的广义交。也可以决定提交至少出现于 *S1*、*S2* 和 *S3* 中的一个集合的网址，这是三个集合的广义并。

练习 14.11 计算 *S1*、*S2* 和 *S3* 的广义交。 □

练习 14.12 计算 *S1*、*S2* 和 *S3* 的广义并。 □

14.2.2 使用序列建模

可以假定 $S1$、$S2$ 和 $S3$ 所表示的结果的顺序很重要：第一个结果与用户的查询最匹配，第二个结果与用户的查询次匹配，等等。为此，也许用序列来表示这一信息更好。因此，下面给出的信息表示比上述的信息表示更加具体。

$S1 = \langle$ www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com $\rangle$

$S2 = \langle$ www.abc.com, www.fgh.com, www.xyz.com $\rangle$

$S3 = \langle$ www.xyz.com, www.lmn.com $\rangle$

练习 14.13 $S1$ 是单射序列吗？$S2$ 和 $S3$ 又如何呢？ □

练习 14.14 计算下列各题。

1. $reverse\ S2$
2. $\#\ tail\ S3$
3. $head\ tail\ S1$
4. $(S1 \frown S2) \upharpoonright (\mathrm{ran}\ S3)$
5. $ran((S1 \frown S2) \upharpoonright (\mathrm{ran}\ S3))$ □

练习 14.15 假设有序列 $S1$、$S2$ 和 $S3$，使用谓词逻辑写出下列陈述。

1. 若一个网址出现于 $S1$，那么它一定出现于 $S2$。
2. 没有同时出现于 $S1$ 和 $S2$ 的网址。
3. 若在 $S1$ 中，网址 u 出现于网址 v 之前，而且 u 和 v 都出现于 $S2$，那么在 $S2$ 中 u 同样出现于 v 之前。
4. 若一个网址出现于 $S1$ 的位置 1，那么它同样出现于 $S2$ 和 $S3$ 的相同位置。 □

14.2.3 使用关系建模

到此为止，我们把 $S1$、$S2$ 和 $S3$ 看成相互独立的对象，不论它们是用集合还是用序列表示的。通过使用关系，可以把各搜索引擎的信息组合在一起来决定元搜索引擎的综合结果。

假设有类型 $ENGINE$（使得如 $S1 \in ENGINE$）以及类型 URL（使得如 www.xyz.com $\in URL$），可以如下定义关系 $result \in ENGINE \leftrightarrow \mathrm{seq}\ URL$。

$result = \{ S1 \mapsto \langle$ www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com $\rangle$, $S2 \mapsto \langle$ www.abc.com, www.fgh.com, www.xyz.com $\rangle$, $S3 \mapsto \langle$ www.xyz.com, www.lmn.com $\rangle \}$

练习 14.16 假设 $result$ 是上面所给的关系，计算下列各题。

1. $\mathrm{dom}\ result$
2. $\mathrm{ran}\ result$
3. $\bigcup \{ s : \mathrm{ran}\ result \bullet \mathrm{ran}\ s \}$
4. $result^{\sim}$
5. $result(\mid \{S2\} \mid)$
6. $\{S1\} \lhd result$
7. $result \rhd \{ s : \mathrm{seq}\ URL \mid \text{www.fgh.com} \in \mathrm{ran}\ s \}$ □

练习 14.17 $result$ 是同类关系还是异类关系？ □

练习 14.18 说明如何通过集合描述从 $result$ 构造下列集合。

1. $\{S2, S3\}$
2. $\{ S1 \mapsto \langle$ www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com $\rangle \}$
3. $\{ S1 \mapsto$ www.xyz.com, $S2 \mapsto$ www.abc.com, $S3 \mapsto$ www.xyz.com $\}$
4. $\{$ www.xyz.com $\mapsto S1$, www.abc.com $\mapsto S2$, www.xyz.com $\mapsto S3 \}$ □

练习 14.19 $result$ 是函数吗？ □

练习 14.20 计算下列各题。

1. $result\ S1$
2. $result \oplus \{ e : \mathrm{dom}\ result \bullet e \mapsto tail\ result\ e \}$

3. $result \oplus \{e: \mathrm{dom}\ result \bullet e \mapsto tail(result\ e \upharpoonright \{\text{www.xyz.com}, \text{www.abc.com}\})\}$ □

练习 14.21 *result* 是单射吗？是满射吗？ □

14.2.4 继续讨论

从各个搜索引擎得到结果后，元搜索引擎的下一个任务是决定将哪些结果按什么顺序(当我们关心顺序时)提交给用户。有两个途径可以考虑：一个途径是使用集合来定义，另一个途径是使用序列来定义。

第一个途径

第一个途径不关心提交给用户的结果的顺序，只考虑搜索引擎提供了什么信息，然后根据给定的标准决定哪些信息应该提交给用户。需要提交的结果可以通过若干规则获得。而且，这些规则可以用集合描述来形式地表示。

例如，我们也许希望考虑那些所有选定搜索引擎的结果中都出现的网址。这一规则可以如下描述。

$meta = \{u: URL \mid (\forall e: \mathrm{dom}\ result \bullet u \in \mathrm{ran}\ e)\}$

假设 *result* 为如下所给的函数。

$result = \{S1 \mapsto \langle \text{www.xyz.com}, \text{www.lmn.com}, \text{www.abc.com}, \text{www.fgh.com} \rangle, S2 \mapsto \langle \text{www.abc.com}, \text{www.fgh.com}, \text{www.xyz.com} \rangle, S3 \mapsto \langle \text{www.xyz.com}, \text{www.lmn.com} \rangle\}$

那么，这将导致取值为{www.xyz.com}的集合 *meta*。

练习 14.22 用集合描述刻画下列规则。

1. 只考虑 S1 和 S2 同时返回的网址。
2. 只考虑 S1、S2 和 S3 中至少一个返回的网址。
3. 只考虑这样的网址，它是至少一个搜索引擎的结果中的第一个网址。
4. 只考虑这样的网址，它是至少两个搜索引擎的结果中的第一个网址。
5. 只考虑 S1 的结果中的前三个网址。
6. 只考虑 S1、S2 和 S3 的结果中的前三个网址。 □

练习 14.23 设 *result* 有如下值。

$result = \{S1 \mapsto \langle \text{www.xyz.com}, \text{www.lmn.com}, \text{www.abc.com}, \text{www.fgh.com} \rangle, S2 \mapsto \langle \text{www.abc.com}, \text{www.fgh.com}, \text{www.xyz.com} \rangle, S3 \mapsto \langle \text{www.xyz.com}, \text{www.lmn.com} \rangle\}$

给出练习 14.22 各题中的 *meta*。 □

第二个途径

第一个途径只考虑向用户提供哪些网址；进一步精化的途径要求我们考虑提交给用户的结果的顺序。因此，我们要求结果是网址的序列，而非网址的集合。

例如，可以根据每个搜索引擎返回的位置为各网址设置一个分数。回头再看上面给出的例子。

S1	S2	S3
www.xyz.com	www.abc.com	www.xyz.com
www.lmn.com	www.fgh.com	www.lmn.com
www.abc.com	www.xyz.com	
www.fgh.com		

这里，关于 S1，www.xyz.com 可能得到 1 点，www.lmn.com 可能得到 2 点，以此类推。没被该搜索引擎返回的网址则得到某个默认值；在此假设默认值为 100。可以定义一个函数 *position_scores* 来完成这一工作。*position_scores* 的类型是 $URL \times ENGINE \rightarrow \mathbb{N}$。

$$\forall u: URL; e: ENGINE \bullet$$
$$u \in \operatorname{ran} result(e) \Rightarrow position\_scores(u, e) = (\mu n: \mathbb{N} \mid (result\ e)n = u)$$
$$\wedge$$
$$u \notin \operatorname{ran} result(e) \Rightarrow position\_scores(u, e) = 100$$

例如，

position_scores(www. abc. com，S1)＝3

以及

position_scores(www. xyz. com，S2)＝3

其中，

position_scores(www. fgh. com，S3)＝100

练习 14.24 计算下列各题。

1. *position_scores*(www. abc. com，S1)
2. *position_scores*(www. abc. com，S2)
3. *position_scores*(www. abc. com，S3)

□

练习 14.25 在 *position_scores* 的定义中有一个隐含的假设，就是每个搜索引擎返回的都是结果的单射序列。给出 *position_scores* 的一个新定义，这一定义不依赖于这一假设，而且 *position_scores* 将给定网址的位置与它在给定序列的第一次出现相关联。 □

函数 *position_scores* 以搜索引擎为基础来表示搜索引擎的网址的分数总和。为排列元搜索引擎返回的网址，我们需要一个类型为 $URL \rightarrow \mathbb{N}$ 的函数来将这些分数加起来。可以如下定义这样的函数。

$$scores: URL \rightarrow \mathbb{N}$$
$$\forall u: URL \bullet scores(u) = \sum_{e \in ENGINE} position\_scores(u, e)$$

使用上面的函数和如下所给的 *result*，可得下面的结果。

result＝{S1 ↦⟨www. xyz. com，
www. lmn. com，
www. abc. com，
www. fgh. com⟩，
S2 ↦⟨www. abc. com，
www. fgh. com，
www. xyz. com⟩，
S3 ↦⟨www. xyz. com，
www. lmn. com⟩}

scores＝{www. xyz. com ↦ 5，
www. lmn. com ↦ 104，
www. abc. com ↦ 104，
www. fgh. com ↦ 106}

练习 14.26 假设 *result* 为如下结果。

result＝{S1 ↦⟨www. abc. com，
www. fgh. com，
www. lmn. com，
www. xyz. com⟩，
S2 ↦⟨www. fgh. com，
www. xyz. com，
www. rst. com，
www. lmn. com⟩，
S3 ↦⟨www. xyz. com，
www. rst. com，
www. abc. com，
www. lmn. com⟩，
S4 ↦⟨www. fgh. com，
www. lmn. com，
www. rst. com⟩}

对此 *result*，计算 *scores*。 □

函数 *scores* 和适当定义的集合描述 *meta* 使我们得以使用最终的序列向用户提供我们希望提供的结果。可以确保只有那些符合由 *meta* 定义的标准的网址才能出现于该序列。而且，网址出现的顺序由 *scores* 的值决定。通过公理定义，如下给出这样的一个函数的定义。

$$final: \operatorname{iseq} URL$$
$$\operatorname{ran} final = meta$$
$$\wedge$$
$$\forall m, n: \operatorname{dom} final \bullet m < n \Rightarrow scores\ final\ m \leqslant scores\ final\ n$$

练习 14.27 再次考虑如下给出的函数 *result*。

result = {S1 ↦ ⟨www.xyz.com,
www.lmn.com,
www.abc.com,
www.fgh.com⟩,
S2 ↦ ⟨www.abc.com,
www.fgh.com,
www.xyz.com⟩,
S3 ↦ ⟨www.xyz.com,
www.lmn.com⟩}

再考虑如下给出的 *scores*。

scores = {www.xyz.com ↦ 5,
www.lmn.com ↦ 104,
www.abc.com ↦ 104,
www.fgh.com ↦ 106}

对于下列各 *meta* 的定义，确定相应的 *final* 的值。

1. 只考虑 S1 和 S2 同时返回的网址。
2. 只考虑 S1、S2 和 S3 中至少一个返回的网址。
3. 只考虑这样的网址，它是至少一个搜索引擎的结果中的第一个网址。
4. 只考虑这样的网址，它是至少两个搜索引擎的结果中的第一个网址。
5. 只考虑 S1 的结果中的前三个网址。
6. 只考虑 S1、S2 和 S3 的结果中的前三个网址。 □

练习 14.28 再次考虑如下给出的练习 14.26 中的 *result*。

result = {S1 ↦ ⟨www.abc.com,
www.fgh.com,
www.lmn.com,
www.xyz.com⟩,
S2 ↦ ⟨www.fgh.com,
www.xyz.com,
www.rst.com,
www.lmn.com⟩,
S3 ↦ ⟨www.xyz.com,
www.rst.com,
www.abc.com,
www.lmn.com⟩,
S4 ↦ ⟨www.fgh.com,
www.lmn.com,
www.rst.com⟩}

以及相应的 *scores*。

scores = {www.abc.com ↦ 204,
www.fgh.com ↦ 104,
www.lmn.com ↦ 13,
www.rst.com ↦ 108,
www.xyz.com ↦ 107}

对应于练习 14.27 所给的各 *meta* 的定义，确定相应的 *final* 的值。 □

14.3 用于栈和队列的序列

所有计算机科学专业的学生都应熟悉的两个数据结构是栈和队列。这两个结构都可以用序列来建模。在讨论使用序列为这两个结构建模之前，先来概述一下这两个数据结构。

栈是按后进先出的顺序维护数据的数据结构：最后加入数据结构的项最先离开该数据结构。之所以如此称呼该数据结构，是因为它令人想起餐馆中的一摞(a stack of)盘子。在那里，盘子被一个一个地摞起来；当在一摞盘子上再加入一个盘子时，它被放在这摞盘子的上面；当拿走盘子时，从这摞盘子的顶端拿走盘子。

本质上，栈有 4 个基本运算符：*pop*、*push*、*is_empty* 和 *top*。运算符 *pop* 取走栈顶的元素；运算符 *push* 向栈顶压入一个元素；运算符 *is_empty* 返回表明栈是否为空的布尔值；运算符 *top* 返回栈顶元素的值，它不取走该元素。

例如，假设有如下所示的自然数栈。

7
9

11

对该栈运用 *pop* 取走它的栈顶元素，作用后的栈如下所示。

9

11

注意，运算 *pop* 只对非空栈有定义。

向栈中压入 5 得到下面的结果。

5

9

11

对此栈运用 *is_empty* 将生成值 false。进而，对此栈运用 *top* 将生成值 5。与 *pop* 相同，运算符 *top* 只对非空栈有定义。

练习 14.29 考虑下面的栈。

10

12

14

给出对此栈运用下面各运算的结果。

1. *pop*

2. *push*(8)

3. *top*

4. *is_empty* □

队列是按先进先出的顺序维护数据的数据结构：最先加入队列的项最先离开该队列。之所以如此称呼这一数据结构，是因为它令人想起车站的队列。在那里，人们一个挨着一个地排队：新来的人排在队列的尾部；当汽车到站时，人们从前依次上车，从而离开队列。

与栈相同，队列有 4 个基本运算符：*dequeue*、*enqueue*、*is_empty* 和 *first*。运算符 *dequeue* 取走队列的第一个元素；运算符 *enqueue* 在队列的尾部加入一个元素；运算符 *is_empty* 返回表示队列是否为空的布尔值；运算符 *first* 返回队列的第一个元素的值，它不取走该元素。

作为例子，假设有如下自然数队列。

7，9，11

对此队列运用 *dequeue* 取走它的第一个元素，留下下面的队列。

9，11

在队列中加入 5 得到下面的结果。

9，11，5

对此队列运用 *is_empty* 将生成值 false。进而，对此队列运用 *first* 生成值 9。

练习 14.30 考虑下面的队列。

10，12，14

给出对此队列做下列运算的结果。

1. *dequeue*

2. *enqueue*(8)

3. *first*

4. *is_empty* □

可以使用序列给出栈和队列的形式表示。首先考虑栈的形式表示。

考虑某类型 X 的栈集合 $Stack[X]$。在此，假设

$$Stack[X] = \mathrm{seq}\ X$$

因此，例如⟨7，9，11⟩表示如下栈。

7

9

11

使用此形式描述，运算符 *pop* 的定义如下所示。

$$\forall s: Stack[X] \mid s \neq \langle\rangle \bullet pop\ s = tail\ s$$

因此，例如，

$$pop\langle 7, 9, 11\rangle = \langle 9, 11\rangle$$

我们如下定义 *push*。

$$\forall s: Stack[X];\ x: X \bullet push(s, x) = \langle x\rangle \frown s$$

对于上面的例子，这一定义给出下面的结果。

$$push(\langle 9, 11\rangle, 5) = \langle 5, 9, 11\rangle$$

下面，使用下面的谓词定义 *is_empty*。

$$\forall s: Stack[X] \bullet is\_empty(s) \Leftrightarrow s = \langle\rangle$$

最后，函数 *top* 的定义如下所示。

$$\forall s: Stack[X] \mid s \neq \langle\rangle \bullet top\ s = head\ s$$

例如，

$$top\langle 5, 9, 11\rangle = 5$$

练习 14.31 使用结构归纳法证明下式成立。

$\forall s: Stack[X]; x: X \cdot pop(push(s, x)) = s$ □

练习 14.32 使用等值推理证明练习 14.32 中的定理。 □

练习 14.33 使用等值推理证明下述定理。

$is\_empty(push(pop(s), top(s))) \Leftrightarrow is\_empty(s)$ □

下面考虑队列的建模。用 $Queue[X]$ 表示类型 X 的队列的集合，使得

$Queue[X] = \text{seq } X$

因此，例如，序列 $\langle 7, 9, 11 \rangle$ 表示下面的队列。

7，9，11

运算符 *dequeue* 的定义如下所示。

$$\forall q: Queue[X] \mid q \neq \langle\rangle \cdot dequeue\ q = tail\ q$$

例如，

$dequeue\langle 7, 9, 11 \rangle = \langle 9, 11 \rangle$

enqueue 的定义如下所示。

$$\forall q: Queue[X]; x: X \cdot enqueue(q, x) = q \frown \langle x \rangle$$

例如，

$enqueue(\langle 9, 11 \rangle, 5) = \langle 9, 11, 5 \rangle$

使用谓词如下定义 *is_empty*。

$\forall q: Queue[X] \cdot is\_empty(q) \Leftrightarrow q = \langle\rangle$

最后，*first* 的定义如下所示。

$$\forall q: Queue[X] \mid q \neq \langle\rangle \cdot first\ q = head\ q$$

例如，

$first\langle 9, 11, 5 \rangle = 9$

练习 14.34 下面的陈述是定理吗？

$\forall q: Queue[X]; x: X \cdot dequeue(enqueue(q, x)) = q$ □

练习 14.35 使用等值推理证明下述定理。

$\forall x: X \cdot dequeue(enqueue(\langle\rangle, x)) = \langle\rangle$ □

练习 14.36 使用等值推理证明下述定理。

$\forall q: Queue[X]; x, y: X \cdot is\_empty(q) \Rightarrow \# enqueue(enqueue(q, x), y) = 2$ □

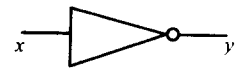
14.4 数字电路

布尔代数的典型应用当然是数字电路(digital circuit)。本节讨论布尔代数在数字电路的应用。关于本话题，感兴趣的读者可以参考[Men70]，那里有许多本话题的问题及解答。

数字计算机中所用的二进制数制含有两个符号：0 和 1。数字计算机是通过传送信号来工作的，传送的信号只能取两种值；通常使用二进制数制的两个符号来表示这两种值。

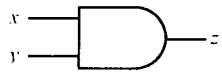
逻辑门回路(logic gate circuit)完成数字计算的任务。逻辑门回路由一组输入和一个或多个输出组成：每个输入取分别用 0 和 1 表示的两个状态之一，同样，每个输出也取分别用 0 和 1 表示的两个状态之一。逻辑门回路由*逻辑门*构成。就像我们将要看到的那样，考虑由三个逻辑门以及值 0 和 1 构成的布尔代数。需要牢记的是，虽然这些逻辑门对应于物理装置，但是我们关心的是它们的抽象、逻辑行为，像电力供应和同步等技术问题不在考虑范畴。

第一个逻辑门是非(NOT)门(有时称为反相器)。这一数字电路接受一个输入(记作 x)，生成一个输出(记作 y)。下面是非门的图示。



其中，若输入是 0，则输出是 1。反之，若输入是 1，则输出是 0。这一逻辑门对应于布尔代数的取补运算符，即 $y = x'$。

与(AND)门是接受两个输入(x 和 y)、生成一个输出(z)的数字电路。下面是与门的图示。



其中，若输入是两个 1，则输出是 1。否则，若至少一个输入是 0，则输出是 0。这一逻辑门对应于布尔代数的乘积运算符，即 $z=x*y$。

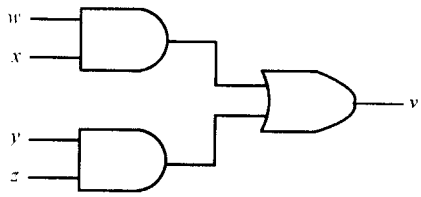
或(OR)门是接受两个输入(x 和 y)、生成一个输出(z)的数字电路。下面是或门的图示。

其中，若至少有一个输入是 1，则输出是 1。否则，若两个输入都是 0，则输出是 0。这一逻辑门对应于布尔代数的和运算符，即 $z=x+y$。

练习 14.37 说明值 0 和 1 以及非门、与门和或门构成布尔代数。 □

当然，由于这些逻辑门构成布尔代数，所以第 5 章推导出来的规则都可运用于逻辑门。

通过组合上述三种逻辑门可以构造复杂的数字电路。例如，两个与门的输出可以进一步成为另一个或门的输入。这可由下图表示。



这一电路有 4 个输入：w、x、y 和 z，以及 1 个输出：v。而且，这一电路对应于布尔等式 $v=wx+yz$。

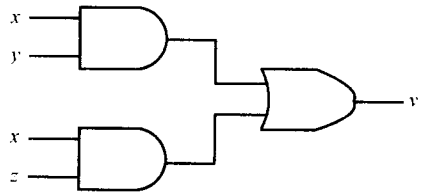
正如计算真值表时要依次考虑每个部分命题一样，计算数字电路的可能输出时，要依次计算每个部分逻辑门。为求上面电路的输出，首先求 w AND x，然后求 y AND z，最后求(w AND x) OR(y AND z)。

练习 14.38 假设有上图所示的数字电路。对于下列输入组合，计算输出 v。

1. $w=0$，$x=1$，$y=1$，$z=0$
2. $w=1$，$x=1$，$y=0$，$z=0$
3. $w=0$，$x=1$，$y=0$，$z=1$ □

练习 14.39 命题 $x \vee y$ 可以用数字电路中的 x OR y 表示。还有，命题 $\neg x$ 和 $x \wedge y$ 可以用数字电路中的 NOT x 和 x AND y 来表示。说明如何用数字电路表示 $x \Rightarrow y$ 和 $x \Leftrightarrow y$。 □

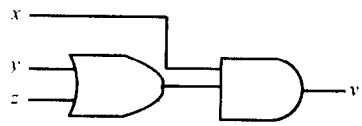
考虑上面表示(x AND y) OR (x AND z)的电路的一个变形，其图示如下所示。



这一电路由三个门组成：两个与门和一个或门。回忆一下，一个分配律规则指出

$$x \text{ AND } (y \text{ OR } z)$$
$$=(x \text{ AND } y) \text{ OR } (x \text{ AND } z)$$

因此，下面的电路与上面所给的电路等价。



由于这一电路只含有两个逻辑门，一个与门和一个或门，所以从现实和物理的角度看，它比原来的电路更好。这是因为只使用两个逻辑门实现它需要更少的物理空间。而且，由于只需两个计算，而不是三个，所以可以更快地得到输出。由此可见，第 5 章所学的布尔代数的规则以及第 3 章所学的等值推理技术都可以应用于数字电路，这非常重要。

练习 14.40 说明如何将下式表示的含

有 7 个逻辑门的电路转换成含有 2 个逻辑门的等价电路。

((x AND 0)OR(x AND 1))OR((NOT x)AND(NOT y)) □

练习 14.41 证明如下电路对的等价性。

1. (x AND y)OR(x AND(NOT y))

   和

   x

2. x OR(((NOT x)AND y)OR((NOT z)AND y))

   和

   x OR y □

14.5 学校数据库

本节考虑如何使用函数来对学校的数据库进行建模。[Hay93]含有运用 Z 的一系列相似的案例研究，感兴趣的读者可以参考；其中，Z 是构建于这个案例研究中所用记法的一种形式描述技术。

首先，介绍与形式描述相关的集合。第一个这样的集合(*Teacher*)是教师的集合。类似地，有集合 *Student*。假设集合 *Subject* 含有历史和数学这样的元素。还假定有一个集合 *Classroom*。下面，我们如下定义集合 *Day*。

Day={星期一，星期二，星期三，星期四，星期五}

假设有类型为 *Day* × *Day* 的如下关系 *before*。

before={星期一↦星期二，星期一↦星期三，星期一↦星期四，星期一↦星期五，星期二↦星期三，星期二↦星期四，星期二↦星期五，星期三↦星期四，星期三↦星期五，星期四↦星期五}

最后，集合 *Period* 表示每天可能有课的课时：假定每天有 8 节课。

Period=1..8

第一个函数将科目映射到教师所教课程的集合。

teacher: *Subject* → ℙ *Teacher*

练习 14.42 你期望函数 *teacher* 是单射吗？ □

练习 14.43 用 *teacher* 定义下列集合。

1. 地理教师的集合。

2. 教一门以上课程的教师的集合。 □

下面，我们考虑函数 *chosen_subjects*，这一函数将学生映射到他们所选课程的集合。

chosen_subjects: *Student* → ℙ *Subject*

练习 14.44 通过公理定义说明如何将下列限制施加于 *chosen_subjects*。

1. 所有学生都至少选择 2 门课程，最多选择 5 门课程。

2. 一名学生选择人文课，当且仅当他既没有选历史课，也没有选地理课。

3. 每个至少选择 4 门课程的学生必须选择数学。

4. 以上限制的合取。 □

练习 14.45 使用已有的函数 *chosen_subjects* 和 *teacher* 定义函数 *taught_by*，该函数将每个学生映射到所有教授该学生所选课程的教师的集合。 □

练习 14.46 使用 *chosen_subjects* 定义下列集合。

1. 学习数学的所有学生的集合。

2. 刚好选择 3 门课程的学生的集合。

3. 刚好选择 3 门课程的学生所选的所有课程的集合。 □

我们现在考虑教师和学生间可能成立的两个关系中的第一个关系。每位教师都与一个辅导小组相关联，辅导小组由若干名学生组成。这一信息由函数 *tutees* 表示。

tutees: *Teacher* → ℙ *Student*

练习 14.47 通过公理定义说明如何将

下列限制施加于 *tutees*。

1. 每个辅导小组由 15～30 名学生组成(包括 15 人和 30 人)。

2. 没有出现于多个辅导小组的学生。

3. 以上限制的合取。 □

练习 14.48 使用 *teacher* 和 *tutees* 定义函数 *tutor_teaches*，它将每个学生映射到他们的辅导教师所教的课程。 □

练习 14.49 使用 *teacher*、*chosen_subjects* 和 *tutees* 定义下列集合。

1. 所有在其辅导小组刚好有 30 名学生的辅导教师的集合。

2. 至少辅导一名选择 5 门课程的学生的辅导教师的集合。

3. 所有序偶(s，t)的集合，其中 t 是 s 的辅导教师且 t 教授 s 所选的一门课程。 □

现在，我们考虑与学生的成绩相关的信息，用函数 *marks* 来表示这一信息。*marks* 将类型为 $Student \times Subject$ 的序偶映射到自然数。

marks：$Student \times Subject \rightarrow \mathbb{N}$

练习 14.50 通过公理定义说明如何对 *marks* 施加下列限制。

1. 每一门课程的最高成绩是 100。

2. 没有多于 5 门课程成绩的学生。

3. 以上限制的合取。 □

练习 14.51 给出函数 *marks* 和 *chosen_subjects* 间必须满足的关系的形式定义。 □

练习 14.52 使用 *marks* 定义关系 *best_subject*，它将每名学生映射到他们得到最高分数的课程(可以是复数)。 □

练习 14.53 使用集合描述定义下列集合。

1. 在其所选的课程中至少有一门过 50 分的学生的集合。

2. 历史课和地理课都超过 50 分的学生的集合。

3. 其辅导教师所教的某门课程过 50 分的学生的集合。

4. 每个学生都过 50 分的课程的集合。 □

现在，我们考虑教师和学生间的第二个关系。函数 *class* 将类型为 $Subject \times Teacher \times Day \times Period$ 的多元组映射到学生的集合。因此，例如，*class*(s，t，d，p)表示教师 t 在星期 d 的第 p 节所教课程 s 的学生的集合，这一函数的类型如下所示。

class：$Subject \times Teacher \times Day \times Period \rightarrow \mathbb{F}\, Student$

练习 14.54 通过公理定义说明如何对 *class* 施加下列限制。

1. 没有可以同时在不同地方的学生。

2. 没有可以同时在不同地方的教师。

3. 以上限制的合取。 □

练习 14.55 陈述下面函数对应该满足的关系。

1. *class* 和 *chosen_subjects*

2. *class* 和 *teacher*

3. *class* 和 *marks* □

练习 14.56 定义一个函数 *taught_tutees*，它把给定教师 t 映射到 t 所教授且辅导的学生的集合。 □

练习 14.57 使用集合描述定义下列集合。

1. 星期一的第一节教授的所有课程的集合。

2. 星期一的第一节上历史课的所有学生的集合。

3. 有历史课的序偶(d，p)$\in Day \times Period$ 的集合。 □

下面，我们考虑每个教室的位置，这一信息由下面的函数 *classroom* 表示。

classroom：$Subject \times Teacher \times Period \times Day \rightarrow Classroom$

练习 14.58 通过公理定义说明如何将下列限制施加于 *classroom*。

1. 在任意给定时间，一位教师最多只能在一个教室教课。

2. 在任意给定时间，一个教室最多只能有一位教师教课。

3. 上面限制的合取。 □

练习 14.59 陈述下面函数对应满足的关系。

1. *classroom* 和 *class*

2. *classroom* 和 *teacher* □

练习 14.60 使用 *class* 定义下列函数。

1. 给定教师、课时和日期，返回教室的函数 *location*。

2. 将每一位教师映射到他所教授的课时数目的函数 *teaching_periods*。 □

练习 14.61 使用集合描述定义下列集合。

1. 星期一的第一节被占用的教室的集合。

2. 上历史课的教室的集合。

3. 所有教室都在使用的序偶 $(d, p) \in Day \times Period$ 的集合。 □

最后，我们考虑函数 *timetable*，它将每名学生与一个时间表(即一个多元组序列)相关联。

$timetable: Student \rightarrow \mathrm{seq}(Period \times Day \times Subject \times Teacher \times Classroom)$

练习 14.62 通过公理定义，说明如何对 *timetable* 施加下列限制。

1. 没有学生的关联时间表同时含两个地方。

2. 每名学生的时间表都是按时间排列的，即星期一第一节的多元组在星期一第二节的多元组之前出现，以此类推。

3. 每个学生的时间表中最多有 5 门课程。 □

练习 14.63 陈述下列函数对应该满足的关系。

1. *timetable* 和 *classroom*

2. *timetable* 和 *class*

3. *timetable* 和 *marks*

4. *timetable* 和 *chosen_subjects* □

练习 14.64 使用 *timetable* 定义下列函数。

1. 给定学生、日期和课时，返回教室的函数 *student_location*。

2. 给定学生，返回教该学生的所有教师的集合的函数 *teachers*。

3. 将每名学生映射到他的所有有课课时的数目的函数 *taught_periods*。 □

练习 14.65 使用集合描述定义下列集合。

1. 所有课时都有课的学生的集合(注意，一周有 40 课时)。

2. 星期一第一节有课的学生的集合。

3. 星期一第一节有历史课的学生的集合。 □

14.6 知识库系统

从计算的角度看，专家系统领域致力于开发表示知识和特殊领域的专家的经验的计算机系统的研究。所有专家系统的两个基本组成是**知识库**和**推理规则集合**。前者是一组已知的事实(特别地，一组真谓词)，而后者是一组规则，这些规则使我们可以从已知的事实得到新事实(类似于自然演绎的规则)。这样，用户可以对系统提出查询，而系统按知识库所存储的信息生成回答。

下面给出一个简单的知识库系统的例子。首先，考虑下面的事实。

capital(法国，巴黎)
currency(法国，法郎)
capital(意大利，罗马)
currency(意大利，里拉)
capital(西班牙，马德里)
currency(西班牙，比塞塔)
capital(澳大利亚，堪培拉)
currency(澳大利亚，澳元)
capital(美国，华盛顿)
currency(美国，美元)

lives_in(皮埃尔，巴黎)

origin(皮埃尔，法国)

lives_in(伊格尔，威尼斯)

origin(伊格尔，意大利)

lives_in(安娜，马德里)

origin(安娜，加拿大)

lives_in(安德烈，布里斯班)

origin(安德烈，英国)

lives_in(西里尔，巴尔的摩)

origin(西里尔，津巴布韦)

在此，*capital*(x，y)表示 y 是国家 x 的首都；*currency*(x，y)表示 y 是国家 x 的货币；*lives_in*(x，y)表示 x 住在城市 y；而 *origin*(x，y)表示 x 是从国家 y 来的。

查询写成

? — *capital*(a，b)

这一查询询问系统 a 是否是 b 的首都。如果系统可以断定它是真的，那么返回 *true*；否则返回 *false*。例如，查询

? — *capital*(法国，巴黎)

返回结果 true。

练习 14.66 给出下列查询的结果。

1. ? — *capital*(澳大利亚，悉尼)

2. ? — *capital*(美国，华盛顿) □

构造了系统的知识库之后，现在就可以考虑系统的推理规则了。这些推理规则使我们得以从已有的谓词构造新谓词。

例如，可以如下陈述谓词 *in* 和 *capital* 间的特殊关系。

capital(x，y)⇒*in*(y，x)

在此，*in*(y，x)表示城市 y 在国家 x。如果 y 是 x 的首都，那么这显然是成立的。因此，例如，*in*(巴黎，法国)返回真，因为我们知道 *capital*(法国，巴黎)成立，而由上面的蕴含式，*in*(巴黎，法国)为真。另一方面，不能推理 *in*(里尔，法国)，因为没有建立这一谓词真假的信息。由此，系统给出 *in*(里尔，法国)为假的结论。

练习 14.67 根据以上事实和推理规则，给出下列查询的结果。

1. ? — *in*(马德里，西班牙)

2. ? — *in*(巴塞罗那，西班牙)

3. ? — *in*(罗马，西班牙) □

练习 14.68 使用 *capital*、*lives_in*、*currency* 和 *origin*，定义表示下列谓词的推理规则。

1. *lives_in_capital*：*lives_in_capital*(x)表示 x 住在首都的事实。

2. *lives_in_home_country*：*lives_in_home_country*(x)表示 x 住在他的祖国的事实。

3. *lives_in_foreign_country*：*lives_in_foreign_country*(x)表示 x 住在外国的事实。

4. *spends*：*spends*(x，y)表示 x 在他居住的国家使用货币 y。 □

练习 14.69 计算下列查询的结果。

1. ? — *lives_in_capital*(伊格尔)

2. ? — *lives_in_home_country*(伊格尔)

3. ? — *lives_in_foreign_country*(伊格尔)

4. ? — *spends*(伊格尔，里拉) □

练习 14.70 对于练习 14.69 中的各查询，会产生一些错误的结果，如何修改系统来修正这些错误？ □

在某些情况下，问“伊格尔住在首都城市吗？”这样的问题不合适，而应该向系统询问在系统中都有谁住在首都城市。为了做到这一点，作为参数，可以传递变量(例如 x)，而不是值(例如，伊格尔)。在这样的情况下，查询 $p(x)$将返回使谓词 p 为真的 x 的值。

例如，查询

? — *lives_in_capital*(x)

将返回结果

x=安娜

和

x=皮埃尔

也就是说，它返回满足描述 x 的条件的所有元素的名字。

练习 14.71 计算下列各题。

1. $spends$(安娜，x)
2. $spends$(x，比塞塔)
3. $spends(x, y)$
4. $lives\_in\_capital(x) \wedge lives\_in\_home\_country(x)$
5. $lives\_in\_capital(x) \wedge lives\_in\_home\_country(y)$ □

14.7 练习解答

14.1 初始值为

$value = \varnothing$

14.2 $value$ 被更新成为

$value \cup \{x \mapsto 0\}$

14.3

1. $value$ 变成 $value \oplus \{x \mapsto 4\}$
2. $value$ 变成 $value \oplus \{x \mapsto value\ y\}$
3. $value$ 变成 $value \oplus \{x \mapsto value\ y,\ y \mapsto value\ x\}$

14.4

1. $\varnothing$
2. $\{x \mapsto 0\}$
3. $\{x \mapsto 0,\ y \mapsto 0\}$
4. $\{x \mapsto 0,\ y \mapsto 0,\ z \mapsto 0\}$
5. $\{x \mapsto 1,\ y \mapsto 0,\ z \mapsto 0\}$
6. $\{x \mapsto 1,\ y \mapsto 2,\ z \mapsto 0\}$
7. $\{x \mapsto 1,\ y \mapsto 2,\ z \mapsto 3\}$
8. $\{x \mapsto 2,\ y \mapsto 1,\ z \mapsto 3\}$
9. $\{x \mapsto 2,\ y \mapsto 3,\ z \mapsto 1\}$
10. $\{x \mapsto 2,\ y \mapsto 3,\ z \mapsto 1\}$

14.5

1. $value\ x = value\ y$
2. $value\ x = 3 \Rightarrow value\ y = 2$
3. $value\ x = 0 \wedge value\ y = 0 \Rightarrow z \notin \mathrm{dom}\ value$
4. $\mathrm{dom}\ value = \{x,\ y\} \Rightarrow value\ x \geqslant value\ y$
5. $value\ x = value\ y \Leftrightarrow value\ x = 0 \wedge value\ y = 0$

14.6[○]

1. $\forall v: \mathrm{dom}\ value \mid value\ v \neq 0 \bullet value\ v \geqslant 10$
2. $\forall v, w: \mathrm{dom}\ value \mid v \neq w \bullet value\ v \neq value\ w$
3. $\exists v: \mathrm{dom}\ value \bullet value\ v = 0$
4. $\exists_1 v: \mathrm{dom}\ value \bullet value\ v > value\ x$
5. $\exists v: \mathrm{dom}\ value \bullet value\ v \neq 0 \Rightarrow \forall v: \mathrm{dom}\ value \bullet value\ v \neq 0$

14.7

$$
\begin{array}{|l}
ordered: \mathrm{seq}\ VARIABLE \\
\hline
\# ordered = \# value \\
\wedge \\
\mathrm{ran}\ ordered = \mathrm{dom}\ value \\
\wedge \\
\forall m, n: \mathrm{dom}\ ordered \bullet m < n \Leftrightarrow \\
value\ ordered\ m \geqslant value\ ordered\ n
\end{array}
$$

14.8 首先，如下定义函数 add。

$add(\langle\rangle) = 0$

$add(\langle x \rangle \frown s) = value\ x + add(s)$

由此，$add(ordered)$给出我们希望的结果。

14.9 首先，如下定义函数 $double$。

$double(\langle\rangle) = \langle\rangle$

$double(\langle x \rangle \frown s) = \langle 2 \times value\ x \rangle \frown double\ s$

由此，$double(ordered)$给出我们希望的结果。

14.10 首先考虑基底阶段。可以如下证明。

$add(double\langle\rangle) = add\langle\rangle$

○ 需要考虑 $valne = \varnothing$ 的特殊情况。后面解答类似。——译者注

$=0$

$=2\times 0$

$=2\times(add\langle\rangle)$

现在，考虑归纳阶段。归纳假设由下式给出。

$add(double\ s)=2\times(add\ s)$

归纳阶段的证明如下所示。

$add(double\langle x\rangle\frown s)$

$=add(\langle 2\times value x\rangle+double\ s)$

$=(2\times value\ x)+add(double\ s)$

$=(2\times value\ x)+(2\times(add\ s))$

$=2\times(value\ x+add\ s)$

$=2\times(add\langle x\rangle\frown s)$

14.11

$\bigcap\{S1, S2, S3\}=\{$www.xyz.com$\}$

14.12

$\bigcup\{S1, S2, S3\}=\{$www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com$\}$

14.13 它们都是单射序列。

14.14

1. ⟨www.xyz.com, www.fgh.com, www.abc.com⟩

2. 1

3. www.lmn.com

4. ⟨www.xyz.com, www.lmn.com, www.xyz.com⟩

5. {www.xyz.com, www.lmn.com}

14.15

1. $\forall u: URL \cdot u\in \mathrm{ran}\ S1\Rightarrow u\in \mathrm{ran}\ S2$

2. $\forall u: URL \cdot u\notin \mathrm{ran}\ S1\cap \mathrm{ran}\ S2$

3. $\forall u, v: URL \mid u, v\in \mathrm{ran}\ S1 \cdot S1^{\sim} u<S1^{\sim} v\wedge u\in \mathrm{ran}\ S2\wedge v\in \mathrm{ran}\ S2\Rightarrow S2^{\sim} u<S2^{\sim} v$

4. $\forall u: URL \cdot S1^{\sim} u=1\Rightarrow S2^{\sim} u=1\wedge S3^{\sim} u=1$

14.16

1. $\{S1, S2, S3\}$

2. {⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩, ⟨www.abc.com, www.fgh.com, www.xyz.com⟩, ⟨www.xyz.com, www.lmn.com⟩}

3. {www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com}

4. {⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩↦S1, ⟨www.abc.com, www.fgh.com, www.xyz.com⟩↦S2, ⟨www.xyz.com, www.lmn.com⟩↦S3}

5. {⟨www.abc.com, www.fgh.com, www.xyz.com⟩}

6. {S1↦⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩}

7. {S3↦⟨www.xyz.com, www.lmn.com⟩}

14.17 它是异类关系。

14.18

1. $\{e: \mathrm{dom}\ result \mid e\neq S1\}$

2. $\{e: \mathrm{dom}\ result \mid e=S1 \cdot e\mapsto result\ e\}$

3. $\{e: \mathrm{dom}\ result \cdot e\mapsto head\ result\ e\}$

4. $\{e: \mathrm{dom}\ result \cdot head\ result\ e\mapsto e\}$

14.19 是的，它是函数。

14.20

1. ⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩

2. {S1↦⟨www.lmn.com, www.abc.com, www.fgh.com⟩, S2↦⟨www.fgh.com, www.xyz.com⟩, S3↦⟨www.lmn.com⟩}

3. {S1↦⟨www.abc.com⟩, S2↦⟨www.xyz.com⟩, S3↦⟨⟩}

14.21 它是单射，但不是满射。

14.22

1. $meta=\{u: URL \mid u\in(\mathrm{ran}\ result\ S1\cap \mathrm{ran}\ result\ S2)\}$

2. $meta=\{u: URL \mid u\in(\mathrm{ran}\ result\ S1\cup \mathrm{ran}\ result\ S2\cup \mathrm{ran}\ result\ S3)\}$

3. $meta=\{e: \mathrm{dom}\ result \cdot head\ result\ e\}$

4. $meta = \{u: URL \mid (\exists e_1, e_2: ENGINE \mid e_1 \neq e_2 \bullet head\ result\ e_1 = u \wedge head\ result\ e_2 = u)\}$

5. $meta = \{u: URL \mid (\exists i: 1..3 \bullet (result\ S1)i = u)\}$

6. $meta = \{u: URL; e: \{S1, S2, S3\}; i: 1..3 \mid (result\ e)i = u \bullet u\}$

14.23

1. $meta = \{$www.xyz.com, www.abc.com, www.fgh.com$\}$

2. $meta = \{$www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com$\}$

3. $meta = \{$www.xyz.com, www.abc.com$\}$

4. $meta = \{$www.xyz.com$\}$

5. $meta = \{$www.xyz.com, www.lmn.com, www.abc.com$\}$

6. $meta = \{$www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com$\}$

14.24

1. 3

2. 1

3. 100

14.25

$$\forall u: URL; e: ENGINE \bullet$$
$$u \in \operatorname{ran} result(e) \Rightarrow position\_scores(u, e) = (\mu n: \mathbb{N} \mid (result\ e)n = u \wedge \forall m: \mathbb{N} \mid (result\ e)m = u \bullet m \geqslant n)$$
$$\wedge$$
$$u \notin \operatorname{ran} result(e) \Rightarrow position\_scores(u, e) = 100$$

14.26

$$scores = \{\text{www.abc.com} \mapsto 204, \text{www.fgh.com} \mapsto 104, \text{www.lmn.com} \mapsto 13, \text{www.rst.com} \mapsto 108, \text{www.xyz.com} \mapsto 107\}$$

14.27

1. ⟨www.xyz.com, www.abc.com, www.fgh.com⟩

2. ⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩或者⟨www.xyz.com, www.abc.com, www.lmn.com, www.fgh.com⟩

3. ⟨www.xyz.com, www.abc.com⟩

4. ⟨www.xyz.com⟩

5. ⟨www.xyz.com, www.lmn.com, www.abc.com⟩或者⟨www.xyz.com, www.abc.com, www.lmn.com⟩

6. ⟨www.xyz.com, www.lmn.com, www.abc.com, www.fgh.com⟩或者⟨www.xyz.com, www.abc.com, www.lmn.com, www.fgh.com⟩

14.28

1. ⟨www.lmn.com, www.fgh.com, www.xyz.com⟩

2. ⟨www.lmn.com, www.fgh.com, www.xyz.com, www.rst.com, www.abc.com⟩

3. ⟨www.fgh.com, www.xyz.com, www.abc.com⟩

4. ⟨www.fgh.com⟩

5. ⟨www.lmn.com, www.fgh.com, www.abc.com⟩

6. ⟨www.lmn.com, www.fgh.com, www.xyz.com, www.rst.com, www.abc.com⟩

14.29

1. 12
   14
2. 8
   10
   12
   14

3. 10

4. false

14.30

1. 12，14

2. 10，12，14，8

3. 10

4. false

14.31 首先考虑基底阶段，其证明如下所示。

$$pop(push(\langle\rangle, x)) = pop(\langle x\rangle) = \langle\rangle$$

现在考虑归纳阶段，归纳假设由下式给出。

$$pop(push(s, x)) = s$$

可以使用归纳假设来证明归纳阶段。然而，如下所示，也可以不用归纳假设证明归纳阶段。

$$pop(push(\langle y\rangle \frown s, x))$$
$$= pop(\langle x\rangle \frown \langle y\rangle \frown s)$$
$$= \langle y\rangle \frown s$$

14.32 由于这是一个关于序列的定理，而序列是特殊的关系，所以需要证明下式成立。

$$(a, b) \in pop(push(s, x)) \Leftrightarrow (a, b) \in s$$

可以如下完成它的证明。

$$(a, b) \in pop\ push(s, x)$$
$$\Leftrightarrow (a, b) \in tail\ push(s, x) \quad [pop\ 的定义]$$
$$\Leftrightarrow (a, b) \in tail\langle x\rangle \frown s \quad [push\ 的定义]$$
$$\Leftrightarrow (a, b) \in s \quad [tail\ 的定义]$$

14.33

$$is\_empty(push(pop(s), top(s)))$$
$$\Leftrightarrow push(pop(s), top(s)) = \langle\rangle \quad [is\_empty\ 的定义]$$
$$\Leftrightarrow push(tail\ s, top(s)) = \langle\rangle \quad [pop\ 的定义]$$
$$\Leftrightarrow push(tail\ s, head\ s) = \langle\rangle \quad [top\ 的定义]$$
$$\Leftrightarrow \langle head\ s\rangle \frown tail\ s = \langle\rangle \quad [push\ 的定义]$$
$$\Leftrightarrow s = \langle\rangle \quad [规则\ 10.5]$$
$$\Leftrightarrow is\_empty(s) \quad [is\_empty\ 的定义]$$

14.34 不是定理。作为反例，考虑 $q = \langle 1, 2\rangle$ 及 $x = 3$。在此，有

$$dequeue(enqueue(q, x)) = \langle 2, 3\rangle$$

它显然与 q 不同。

14.35

$$(a, b) \in dequeue(enqueue(\langle\rangle, x))$$
$$\Leftrightarrow (a, b) \in dequeue\langle x\rangle \quad [enqueue\ 的定义]$$
$$\Leftrightarrow (a, b) \in \langle\rangle \quad [dequeue\ 的定义]$$

14.36

$$is\_empty(q) \Rightarrow \#enqueue(enqueue(q, x), y) = 2$$
$$\Leftrightarrow \neg is\_empty(q) \vee \#enqueue(enqueue(q, x), y) = 2 \quad [规则\ 3.18]$$
$$\Leftrightarrow \neg is\_empty(q) \vee \#enqueue(q \frown \langle x\rangle, y) = 2 \quad [enqueue\ 的定义]$$
$$\Leftrightarrow \neg is\_empty(q) \vee \#q \frown \langle x\rangle \frown \langle y\rangle = 2 \quad [enqueue\ 的定义]$$
$$\Leftrightarrow \neg is\_empty(q) \vee \#q = 0 \quad [\#\ 的定义]$$
$$\Leftrightarrow \neg is\_empty(q) \vee q = \langle\rangle \quad [\#\ 的定义]$$
$$\Leftrightarrow \neg is\_empty(q) \vee is\_empty(q) \quad [is\_empty\ 的定义]$$
$$\Leftrightarrow true \quad [规则\ 3.15]$$

14.37 由于下面各式为真，所以值 0 和 1 与非门、与门和或门一起构成布尔代数。

x OR $y = y$ OR x

x AND $y = y$ AND x

x OR(y AND z) = (x OR y) AND (x OR z)

x AND(y OR z) = (x AND y) OR (x AND z)

x OR $0 = x$

x AND $1 = x$

x OR(NOT x)=1

x AND(NOT x)=0

14.38

1. $v=0$
2. $v=1$
3. $v=0$

14.39 命题 $x \Rightarrow y$ 可以表示成如下形式。

(NOT x)OR y

而命题 $x \Leftrightarrow y$ 则可以表示成如下形式。

((NOT x)OR y)AND((NOT y)OR x)

14.40 首先，

((x AND 0)OR(x AND 1))OR((NOT x)AND(NOT y))

等价于

(0 OR(x AND 1))OR((NOT x)AND(NOT y))

而它又等价于

(0 OR x)OR((NOT x)AND(NOT y))

这一表达式又等价于

x OR((NOT x)AND(NOT y))

通过分配律，它等价于

(x OR(NOT x))AND(x OR(NOT y))

由于 x OR NOT x 等价于 1，这一电路等价于

x OR(NOT y)

14.41

1. (x AND y)OR(x AND(NOT y))
   =x AND(y OR(NOT y))
   =x AND 1
   =x
2. x OR(((NOT x) AND y) OR ((NOT z)AND y))
   =x OR(y AND((NOT x) OR (NOT z)))
   =(x OR y)AND(x OR((NOT x) OR(NOT z)))
   =(x OR y)AND((x OR(NOT x)) OR(NOT z))
   =(x OR y)AND(1 OR(NOT z))
   =(x OR y)AND 1
   =(x OR y)

14.42 不一定。考虑历史课和地理课由 *Teacher* 的同一子集教授的情况。在这种情况下，*teacher* 不是单射。

14.43

1. $\{t: Teacher \mid t \in teacher\ 地理\}$
2. $\{t: Teacher \mid (\exists s_1, s_2: Subject \bullet s_1 \neq s_2 \wedge t \in teacher\ s_1 \wedge t \in teacher\ s_2)\}$

14.44

1.

$$
\begin{array}{l}
chosen\_subjects: Student \rightarrow \mathbb{F} Subject \\
\hline
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad \#(chosen\_subjects\ s) \geqslant 2 \wedge \#(chosen\_subjects\ s) \leqslant 5
\end{array}
$$

2.

$$
\begin{array}{l}
chosen\_subjects: Student \rightarrow \mathbb{F} Subject \\
\hline
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad humanities \in chosen\_subjects\ s \Leftrightarrow \\
\quad \{历史，地理\} \cap chosen\_subjects\ s = \varnothing
\end{array}
$$

3.

$$
\begin{array}{|l}
chosen\_subjects: Student \rightarrow \mathbb{P} Subject \\
\hline
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad \#(chosen\_subjects\ s) \geqslant 4 \Rightarrow \text{数学} \in chosen\_subjects\ s
\end{array}
$$

4.

$$
\begin{array}{|l}
chosen\_subjects: Student \rightarrow \mathbb{P} Subject \\
\hline
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad \#(chosen\_subjects\ s) \geqslant 2 \wedge \#(chosen\_subjects\ s) \leqslant 5 \\
\wedge \\
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad \text{人文学} \in chosen\_subjects\ s \Leftrightarrow \\
\quad \{\text{历史，地理}\} \cap chosen\_subjects\ s = \varnothing \\
\wedge \\
\forall s: \mathrm{dom}\ chosen\_subjects \bullet \\
\quad \#(chosen\_subjects\ s) \geqslant 4 \Rightarrow \text{数学} \in chosen\_subjects\ s
\end{array}
$$

14.45

$taught\_by = \{s: Student \bullet s \mapsto \bigcup \mathrm{ran}\ (\{chosen\_subjects\ s\} \lhd teacher)\}$

14.46

1. $\{s: Student \mid \text{数学} \in chosen\_subjects\ s\}$
2. $\{s: Student \mid \#(chosen\_subjects\ s) = 3\}$
3. $\{s: Subject \mid (\exists x: Student \bullet \#(chosen\_subjects\ x) = 3 \wedge s \in chosen\_subjects\ x)\}$

14.47

1.

$$
\begin{array}{|l}
tutees: Teacher \rightarrow \mathbb{F} Student \\
\hline
\forall s: \mathrm{ran}\ tutees \bullet \#s \geqslant 15 \wedge \#s \leqslant 30
\end{array}
$$

2.

$$
\begin{array}{|l}
tutees: Teacher \rightarrow \mathbb{F} Student \\
\hline
\forall s, t: \mathrm{dom}\ tutees \mid s \neq t \bullet tutees\ s \cap tutees\ t = \varnothing
\end{array}
$$

3.

$$
\begin{array}{|l}
tutees: Teacher \rightarrow \mathbb{F} Student \\
\hline
\forall s: \mathrm{ran}\ tutees \bullet \#s \geqslant 15 \wedge \#s \leqslant 30 \\
\wedge \\
\forall s, t: \mathrm{dom}\ tutees \mid s \neq t \bullet tutees\ s \\
\qquad \cap tutees\ t = \varnothing
\end{array}
$$

14.48

$tutor\_teaches = \{s: Student; x: Subject; t: Teacher \mid s \in tuteest \wedge t \in teacher\ x \bullet (s, x)\}$

14.49

1. $\{t: Teacher \mid \#(tutees\ t) = 30\}$
2. $\{t: Teacher \mid (\exists s: Student \bullet s \in tuteest \wedge \#(chosen\_subjects\ s) = 5)\}$
3. $\{s: Student; t: Teacher \mid (\exists x: Subject \bullet x \in chosen\_subjects\ s \wedge t \in teacher\ x \wedge s \in tutees\ t)\}$

14.50

1.

$$marks: Student \times Subject \rightarrow \mathbb{N}$$
$$\forall n: \operatorname{ran} marks \bullet n \leqslant 100$$

2.

$$marks: Student \times Subject \rightarrow \mathbb{N}$$
$$\forall s: Student \bullet \# \{x: Subject \mid (s, x) \in \operatorname{dom} marks\} \leqslant 5$$

3.

$$marks: Student \times Subject \rightarrow \mathbb{N}$$
$$\forall n: \operatorname{ran} marks \bullet n \leqslant 100$$
$$\wedge$$
$$\forall s: Student \bullet \# \{x: Subject \mid (s, x) \in \operatorname{dom} marks\} \leqslant 5$$

14.51

$$\forall s: Student; t: Subject \bullet t \in chosen\_subjects\ s \Leftrightarrow (s, t) \in \operatorname{dom} marks$$

14.52

$$best\_subject = \{s: Student; x: Subject \mid (s, x) \in \operatorname{dom} marks \wedge (\forall y: Subject \mid (s, y) \in \operatorname{dom} marks \bullet marks(s, x) \geqslant marks(s, y))\}$$

14.53

1. $\{s: Student \mid (\exists x: Subject \bullet marks(s, x) \geqslant 50)\}$

2. $\{s: Student \mid marks(s, 历史) \geqslant 50 \wedge marks(s, 地理) \geqslant 50\}$

3. $\{s: Student \mid (\exists x: Subject; t: Teacher \bullet s \in tuteest \wedge t \in teacher\ x \wedge marks(s, x) \geqslant 50)\}$

4. $\{s: Subject \mid (\forall x: Student \bullet (x, s) \in \operatorname{dom} marks \Rightarrow marks(x, s) \geqslant 50)\}$

14.54

1.

$$class: Subject \times Teacher \times Day \times Period \rightarrow \mathbb{F} Student$$

$$\forall s: Student; d: Day; p: Period \bullet$$
$$\exists_1 x: Subject; t: Teacher \bullet s \in class(x, t, d, p)$$
$$\vee$$
$$\neg \exists x: Subject; t: Teacher \bullet s \in class(x, t, d, p)$$

2.

$$class: Subject \times Teacher \times Day \times Period \rightarrow \mathbb{F} Student$$

$$\forall t: Teacher; d: Day; p: Period \bullet$$
$$\exists_1 s: Subject \bullet (s, t, d, p) \in \operatorname{dom} class$$
$$\vee$$
$$\neg \exists s: Subject \bullet (s, t, d, p) \in \operatorname{dom} class$$

3.

$class: Subject \times Teacher \times Day \times Period \leftrightarrow Student$

$(\forall s: Student; d: Day; p: Period \bullet$
$\quad \exists_1 x: Subject; t: Teacher \bullet s \in class(x, t, d, p)$
$\quad \vee$
$\quad \neg \exists x: Subject; t: Teacher \bullet s \in class(x, t, d, p))$
$\wedge$
$(\forall t: Teacher; d: Day; p: Period \bullet$
$\quad \exists_1 s: Subject \bullet (s, t, d, p) \in \operatorname{dom} class$
$\quad \vee$
$\quad \neg \exists s: Subject \bullet (s, t, d, p) \in \operatorname{dom} class)$

14.55

1. $\forall s: Student; x: Subject \bullet x \in chosen_subjects\ s \Leftrightarrow \exists d: Day; t: Teacher; p: Period \bullet s \in class(x, t, d, p)$
2. $\forall t: Teacher; x: Subject \bullet t \in teacher\ x \Leftrightarrow \exists d: Day; p: Period \bullet (x, t, d, p) \in \operatorname{dom} class$
3. $\forall s: Student; x: Subject \bullet (s, x) \in \operatorname{dom} marks \Leftrightarrow \exists t: Teacher; d: Day; p: Period \bullet s \in class(x, t, d, p)$

14.56

$taught_tutees = \{t: Teacher \bullet t \mapsto (tutees\ t \cap \{s: Student \mid (\exists d: Day; p: Period; x: Subject \bullet s \in class(x, d, t, p))\})\}$

14.57

1. $\{s: Subject \mid (\exists t: Teacher \bullet (s, t, \text{星期一}, 1) \in \operatorname{dom} class)\}$
2. $\{s: Student \mid (\exists t: Teacher \bullet s \in class(\text{历史}, t, \text{星期一}, 1))\}$
3. $\{d: Day; p: Period \mid (\exists t: Teacher \bullet (\text{历史}, t, d, p) \in \operatorname{dom} class)\}$

14.58

1.

$classroom: Subject \times Teacher \times Period \times Day \rightarrowtail Classroom$

$\forall x, y: \operatorname{dom} classroom \bullet$
$\quad x.2 = y.2 \wedge x.3 = y.3 \wedge x.4 = y.4 \Rightarrow$
$\quad x = y \wedge classroom\ x = classroom\ y$

2.

$classroom: Subject \times Teacher \times Period \times Day \rightarrowtail Classroom$

$\forall x, y: \operatorname{dom} classroom \bullet$
$\quad x.3 = y.3 \wedge x.4 = y.4 \wedge classroom\ x = classroom\ y \Rightarrow$
$\quad x.2 = y.2$

3.

$$
\begin{array}{l}
classroom: Subject \times Teacher \times Period \times Day \rightarrow Classroom \\
\hline
\forall x, y: \mathrm{dom}\ classroom \bullet \\
\qquad x.2 = y.2 \wedge x.3 = y.3 \wedge x.4 = y.4 \Rightarrow \\
\qquad x = y \wedge classroom\ x = classroom\ y \\
\wedge \\
\forall x, y: \mathrm{dom}\ classroom \bullet \\
\qquad x.3 = y.3 \wedge x.4 = y.4 \wedge classroom\ x = classroom\ y \Rightarrow \\
\qquad x.2 = y.2
\end{array}
$$

14.59

1. $\mathrm{dom}\ classroom = \mathrm{dom}\ class$

2. $\forall x: Subject \times Teacher \times Period \times Day \bullet$

$x \in \mathrm{dom}\ classroom \Leftrightarrow x.2 \in teacher\ x.1$

14.60

1. $location = \{x: \mathrm{dom}\ classroom \bullet (x.2, x.3, x.4) \mapsto classroom\ x\}$

2. $teaching\_periods = \{t: Teacher \bullet t \mapsto \#\{x: \mathrm{dom}\ classroom \mid x.2 = t\}\}$

14.61

1. $\{x: \mathrm{dom}\ classroom \mid x.3 = 1 \wedge x.4 = 星期一 \bullet classroom\ x\}$

2. $\{x: \mathrm{dom}\ classroom \mid x.1 = 历史 \bullet classroom\ x\}$

3. $\{d: Day; p: Period \mid (\forall c: Classroom \bullet \exists s: Subject; t: Teacher \bullet classroom(s, t, p, d) = c)\}$

14.62

1.

$$
\begin{array}{l}
timetable: Student \rightarrow \mathrm{seq}\ (Period \times Day \times Subject \times Teacher \times \\
\qquad Classroom) \\
\hline
\forall s: Student \bullet \forall x, y: \mathrm{ran}\ timetable\ s \bullet x.1 = y.1 \wedge x.2 = y.2 \Rightarrow x = y
\end{array}
$$

2.

$$
\begin{array}{l}
timetable: Student \rightarrow \mathrm{seq}\ (Period \times Day \times Subject \times Teacher \times \\
\qquad Classroom) \\
\hline
\forall s: Student \bullet \forall x, y: \mathrm{dom}\ timetable\ s \bullet \\
\quad x < y \Leftrightarrow \\
\qquad (((timetable\ s\ x).2, (timetable\ s\ y).2) \in before) \\
\qquad \vee \\
\qquad ((timetable\ s\ x).2 = (timetable\ s\ y).2 \wedge \\
\qquad\quad (timetable\ s\ x).1 < (timetable\ s\ y).1)
\end{array}
$$

3.

$$timetable: Student \rightarrow \mathrm{seq}(Period \times Day \times Subject \times Teacher \times Classroom)$$

$$\forall s: Student \bullet \# \{x: \mathrm{ran}\ timetable\ s \bullet x.3\} \leqslant 5$$

14.63

1. $\forall s: Student;\ x: Period \times Day \times Subject \times Teacher \times Classroom \bullet x \in \mathrm{ran}\ timetable\ s \Leftrightarrow classroom(x.3,\ x.4,\ x.1,\ x.2) = x.5$

2. $\forall s: Student;\ x: Period \times Day \times Subject \times Teacher \times Classroom \bullet x \in \mathrm{ran}\ timetable\ s \Leftrightarrow s \in class(x.3,\ x.4,\ x.2,\ x.1)$

3. $\forall s: Student;\ x: Period \times Day \times Subject \times Teacher \times Classroom \bullet x \in \mathrm{ran}\ timetable\ s \Leftrightarrow (s,\ x.3) \in \mathrm{dom}\ marks$

4. $\forall s: Student;\ x: Period \times Day \times Subject \times Teacher \times Classroom \bullet x \in \mathrm{ran}\ timetable\ s \Leftrightarrow x.3 \in chosen\_subjects\ s$

14.64

1. $student\_location = \{s: Student;\ d: Day;\ p: Period;\ c: Classroom \mid \exists y: \mathrm{ran}\ timetable\ s \bullet y.1 = p \wedge y.2 = d \wedge y.5 = c \bullet (s,\ d,\ p) \mapsto c\}$

2. $teachers = \{s: Student \bullet s \mapsto \{x: \mathrm{ran}\ timetable\ s \bullet x.4\}\}$

3. $taught\_periods = \{s: Student \bullet s \mapsto \#(\mathrm{ran}\ timetable\ s)\}$

14.65

1. $\{s: Student \mid \# (timetable\ s) = 40\}$

2. $\{s: Student \mid (\exists x: \mathrm{ran}\ timetable\ s \bullet x.1 = 1 \wedge x.2 = 星期一)\}$

3. $\{s: Student \mid (\exists x: \mathrm{ran}\ timetable\ s \bullet x.1 = 1 \wedge x.2 = 星期一 \wedge x.3 = 历史)\}$

14.66 第一个查询返回结果 false，而第二个查询返回结果 true。

14.67 第一个查询返回结果 true。第二个查询返回结果 false，因为系统中没有表明巴塞罗那在西班牙的信息。同理，第三个查询返回结果 false。

14.68

1. $lives\_in(x,\ y) \wedge capital(z,\ y) \Rightarrow lives\_in\_capital(x)$

2. $lives\_in(x,\ y) \wedge origin(x,\ z) \wedge in(y,\ z) \Rightarrow lives\_in\_home\_country(x)$

3. $lives\_in(x,\ y) \wedge origin(x,\ z) \wedge \neg in(y,\ z) \Rightarrow lives\_in\_foreign\_country(x)$

4. $lives\_in(x,\ y) \wedge in(y,\ z) \wedge currency(z,\ w) \Rightarrow spends(x,\ w)$

14.69 正如我们希望的那样，第一个查询的回答是 false。第二个查询返回结果 false，因为根据 *in* 的定义，系统无法推出威尼斯在意大利。根据 *in* 的定义，第三个查询返回结果 true(系统不能确定威尼斯在意大利，所以得出威尼斯不在意大利的结论)。最后，同样根据 *in* 的定义，第四个查询返回结果 false。

14.70 需要添加下面的事实。

in(威尼斯，意大利)

当然，把

in(布里斯班，澳大利亚)

和

in(巴尔的摩，美国)

添加到知识库中也是合理的。

14.71

1. $x =$ 比塞塔

2. x＝安娜

3. x＝皮埃尔，y＝法郎
   x＝伊格尔，y＝里拉
   x＝安娜，y＝比塞塔
   x＝安德烈，y＝澳元
   x＝西里尔，y＝美元

4. x＝皮埃尔

5. x＝皮埃尔，y＝皮埃尔
   x＝皮埃尔，y＝伊格尔
   x＝安娜，y＝皮埃尔
   x＝安娜，y＝伊格尔

参考文献

[Bal97] V. K. Balakrishnan. *Schaum's Outline of Graph Theory, Including Hundreds of Solved Problems.* McGraw-Hill, 1997.

[BMN97] M. Bergmann, J. Moor, and J. Nelson. *The Logic Book with Student Solutions Manual.* McGraw-Hill, 1997.

[Bol86] B. Bollobas. *Combinatorics.* Cambridge University Press, 1986.

[Bru99] R. A. Brualdi. *Introductory Combinatorics.* Prentice-Hall, third edition, 1999.

[CLP00] R. Cori, D. Lascar, and D. Pelletier. *Mathematical Logic: A Course with Exercises Part 1: Propositional Calculus, Boolean Algebras, Predicate Calculus, Completeness Theorems.* Oxford University Press, 2000.

[CLP01] R. Cori, D. Lascar, and D. Pelletier. *Mathematical Logic: A Course with Exercises: Part 2: Recursion Theory, Gödel's Theorem, Set Theory and Model Theory.* Oxford University Press, 2001.

[Dau90] J. W. Dauben. *Georg Cantor.* Princeton University Press, 1990.

[DG00] I. Duntsch and G. Gediga. *Sets, Relations, Functions.* Methodos Publishers, 2000.

[Dun99] W. Dunham, editor. *Euler: The Master of Us All.* The Mathematical Association of America, 1999.

[End77] H. B. Enderton. *Elements of Set Theory.* Academic Press, 1977.

[Eri96] M. J. Erickson. *Introduction to Combinatorics.* John Wiley & Sons, 1996.

[Gar92] A. R. Garciadiego. *Bertrand Russell and the Origins of the Set-theoretic 'Paradoxes'.* Birkhauser Verlag AG, 1992.

[GGB97] I. Grattan-Guinness and G. Bornet. *George Boole: Selected Manuscripts on Logic and its Philosophy.* Birkhauser Verlag AG, 1997.

[Gre98] J. Gregg. *Ones and Zeros: Understanding Boolean Algebra, Digital Circuits, and the Logic of Sets.* Institute of Electrical & Electronic Engineers, 1998.

[Gri99] R. P. Grimaldi. *Discrete and Combinatorial Mathematics: An Applied Introduction.* Addison-Wesley, fourth edition, 1999.

[Hay93] I. J. Hayes. *Specification Case Studies.* Prentice-Hall, second edition, 1993.

[Hen86] J. M. Henle. *An Outline of Set Theory.* Springer-Verlag, 1986.

[Hoa85] C. A. R. Hoare. *Communicating Sequential Processes.* Prentice-Hall, 1985.

[Hun84] G. M. K. Hunt. *Tactics and Strategies in Natural Deduction.* Department of Philosophy, University of Warwick, 1984.

[Hun96] E. Hunt. *Venn Diagrams.* Hunt Publications, 1996.

[Jen00] B. Jensen. *Directed Graphs with Applications.* John Wiley & Sons, 2000.

[Ken80] H. C. Kennedy. *Peano: Life and Works of Giuseppe Peano.* Kluwer Academic Publishers, 1980.

[LL92] S. Lipschutz and M. L. Larson. *2000 Solved Problems in Discrete Mathematics.* McGraw-Hill, 1992.

[Men70] E. Mendelson. *Schaum's Outline of Theory and Problems of Boolean Algebra and Switching Circuits.* Schaum, 1970.

[Men87] E. Mendelson. *Introduction to Mathematical Logic.* Wadsworth and Brookes/Cole, third edition, 1987.

[Mer90] D. D. Merrill. *Augustus De Morgan and the Logic of Relations.* Kluwer Academic Publishers, 1990.

[Mer00] J. Merris. *Graph Theory.* John Wiley & Sons, 2000.

[Mon89] J. D. Monk. *Handbook of Boolean Algebras: Volume 1.* North-Holland, 1989.

[Mon97] R. Monk. *Bertrand Russell: The Spirit of Solitude 1872 - 1921.* Free Press, 1997.

[Ros97] A. W. Roscoe. *The Theory and Practice of Concurrency.* Prentice-Hall, 1997.

[Sol90] D. Solow. *How to Read and Do Proofs.* Wiley, second edition, 1990.

[Spi92] J. M. Spivey. *The Z Notation: A Reference Manual.* Prentice-Hall, second edition, 1992.

[Sup60] P. C. Suppes. *Axiomatic Set Theory.* Van Nostrand, 1960.

[Wan80] M. Wand. *Induction, Recursion, and Programming.* Elsevier North Holland, 1980.

[WD96] J. C. P. Woodcock and J. W. Davies. *Using Z: Specification, refinement, and proof.* Prentice-Hall, 1996.

[Wes95] D. B. West. *Introduction to Graph Theory.* Prentice-Hall, 1995.

[Wil96] R. J. Wilson. *Introduction to Graph Theory.* Longman Higher Education, 1996.

[You64] B. K. Youse. *Mathematical Induction.* Prentice-Hall, 1964.