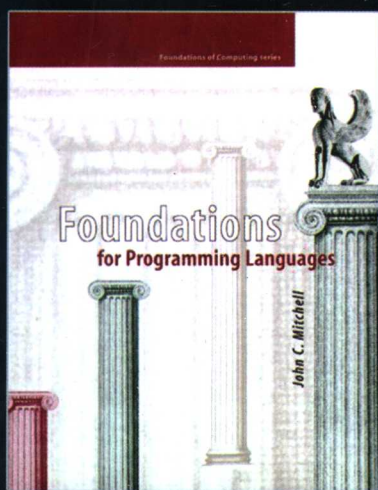




程序设计语言 理论基础

Foundations for
Programming Languages



[美] John C. Mitchell 著

许满武 徐建 袁宜 杨群 译



电子工业出版社

Publishing House of Electronics Industry
<http://www.phei.com.cn>

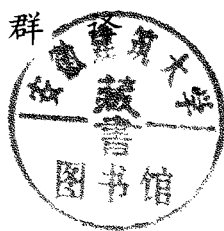
国外计算机科学教材系列

程序设计语言理论基础

Foundations for
Programming Languages

[美] John C. Mitchell 著

许满武 徐建 袁宜 杨群



电子工业出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书提出了一个框架,用于分析程序设计语言的语法、操作和语义性质,该框架基于称为类型化 λ 演算的数学系统。 λ 演算的主要特色是对于函数和其他可计算的值的一种记法,以及一个等式逻辑和用于表达式求值的一组规则。

本书中最简单的系统是称为泛代数的一个等式系统,它可以用来公理化和分析通常用于程序设计的许多数据类型。更先进的技术机制,诸如逻辑关系的方法、范畴论和递归定义类型的语义在中间的几章中论述。本书最后三章研究多态类型,连带讨论了抽象数据类型的说明形式和程序模块、类型适应性和类型推理。

本书可作为理论计算机科学、软件系统和数学专业的大学本科高年级或者研究生初始学习阶段的教材,同时也适合用于高等研究的技术参考书。

Foundations for Programming Languages, Third printing, 2000

© 1996 Massachusetts Institute of Technology

All rights reserved. No part of this publication may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the publisher.

Chinese Simplified language edition published by Publishing House of Electronics Industry, Copyright © 2006

本书中文简体版专有出版权由 MIT Press 授予电子工业出版社,未经许可,不得以任何方式复制或抄袭本书的任何部分。

版权贸易合同登记号 图字:01-2003-6401

图书在版编目 (CIP) 数据

程序设计语言理论基础 / (美)米切尔 (Mitchell, J. C.) 著;许满武等译.

北京:电子工业出版社,2006.11

(国外计算机科学教材系列)

书名原文:Foundations for Programming Languages

ISBN 7-121-03224-4

I. 程... II. ①米... ②许... III. 程序语言-教材 IV. TP312

中国版本图书馆 CIP 数据核字 (2006) 第 119164 号

责任编辑:许菊芳

印 刷:北京季蜂印刷有限公司

装 订:三河市皇庄路通装订厂

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:787 × 1092 1/16 印张:36.5 字数:935 千字

印 次:2006 年 11 月第 1 次印刷

定 价:68.00 元

凡所购买电子工业出版社的图书有缺损问题,请向购买书店调换;若书店售缺,请与本社发行部联系。联系电话:(010) 68279077。邮购电话:(010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010) 88258888。

出版说明

21世纪初的5至10年是我国国民经济和社会发展的关键时期,也是信息产业快速发展的关键时期。在我国加入WTO后的今天,培养一支适应国际化竞争的一流IT人才队伍是我国高等教育的重要任务之一。信息科学和技术方面人才的优劣与多寡,是我国面对国际竞争时成败的关键因素。

当前,正值我国高等教育特别是信息科学领域的教育调整、变革的重大时期,为使我国教育体制与国际化接轨,有条件的高等院校正在为某些信息学科和技术课程使用国外优秀教材和优秀原版教材,以使我国在计算机教学上尽快赶上国际先进水平。

电子工业出版社秉承多年来引进国外优秀图书的经验,翻译出版了“国外计算机科学教材系列”丛书,这套教材覆盖学科范围广、领域宽、层次多,既有本科专业课程教材,也有研究生课程教材,以适应不同院系、不同专业、不同层次的师生对教材的需求,广大师生可自由选择 and 自由组合使用。这些教材涉及的学科方向包括网络与通信、操作系统、计算机组织与结构、算法与数据结构、数据库与信息处理、编程语言、图形图像与多媒体、软件工程等。同时,我们也适当引进了一些优秀英文原版教材,本着翻译版本和英文原版并重的原则,对重点图书既提供英文原版又提供相应的翻译版本。

在图书选题上,我们大都选择国外著名出版公司出版的高校教材,如Pearson Education培生教育出版集团、麦格劳-希尔教育出版集团、麻省理工学院出版社、剑桥大学出版社等。撰写教材的许多作者都是蜚声世界的教授、学者,如道格拉斯·科默(Douglas E. Comer)、威廉·斯托林斯(William Stallings)、哈维·戴特尔(Harvey M. Deitel)、尤利斯·布莱克(Uyless Black)等。

为确保教材的选题质量和翻译质量,我们约请了清华大学、北京大学、北京航空航天大学、复旦大学、上海交通大学、南京大学、浙江大学、哈尔滨工业大学、华中科技大学、西安交通大学、国防科学技术大学、解放军理工大学等著名高校的教授和骨干教师参与了本系列教材的选题、翻译和审校工作。他们中既有讲授同类教材的骨干教师、博士,也有积累了几十年教学经验的老教授和博士生导师。

在该系列教材的选题、翻译和编辑加工过程中,为提高教材质量,我们做了大量细致的工作,包括对所选教材进行全面论证;选择编辑时力求达到专业对口;对排版、印制质量进行严格把关。对于英文教材中出现的错误,我们通过与作者联络和网上下载勘误表等方式,逐一进行了修订。

此外,我们还将与国外著名出版公司合作,提供一些教材的教学支持资料,希望能为授课老师提供帮助。今后,我们将继续加强与各高校教师的密切联系,为广大师生引进更多的国外优秀教材和参考书,为我国计算机科学教学体系与国际教学体系的接轨做出努力。

电子工业出版社

教材出版委员会

- | | | |
|----|-----|---|
| 主任 | 杨芙清 | 北京大学教授
中国科学院院士
北京大学信息与工程学部主任
北京大学软件工程研究所所长 |
| 委员 | 王 珊 | 中国人民大学信息学院院长、教授 |
| | 胡道元 | 清华大学计算机科学与技术系教授
国际信息处理联合会通信系统中国代表 |
| | 钟玉琢 | 清华大学计算机科学与技术系教授、博士生导师
清华大学深圳研究生院信息学部主任 |
| | 谢希仁 | 中国人民解放军理工大学教授
全军网络技术研究中心主任、博士生导师 |
| | 尤晋元 | 上海交通大学计算机科学与工程系教授
上海分布计算技术中心主任 |
| | 施伯乐 | 上海国际数据库研究中心主任、复旦大学教授
中国计算机学会常务理事、上海市计算机学会理事长 |
| | 邹 鹏 | 国防科学技术大学计算机学院教授、博士生导师
教育部计算机基础课程教学指导委员会副主任委员 |
| | 张昆藏 | 青岛大学信息工程学院教授 |

译者序

早在电子计算机出现(1946年)之前,数理逻辑学者们就着手研究了多种自动计算的模型,粗略地可以分为三类:计算机模型(如图灵机),函数模型(如递归函数)和演算模型(如谓词演算和 λ 演算),这些理论工作对其后的计算机的研制以及在计算机上使用的程序设计语言的发展起着决定性的影响。

软件工作者采用程序设计语言来描述问题在计算机上的求解,这里所讲的语言不像人们日常生活中使用的自然语言那样允许灵活多样地理解,从而具有丰富的内涵,问题求解的计算要求程序语言必须具备明确而严格的定义,才能在计算机上加以实现。

至今计算机已深入广泛应用在人类社会活动的各种领域。为适合不同应用领域和设计抽象层次的需要,涌现出了诸多表达风格的程序语言和强力的程序运行机制,它们有力地推进了计算机软件开发的成效。值得注意的是,对它们形式化的研究又反过来使计算机理论得以充实与提高。

本书作者长期深入地探讨以程序语言概念为特征的语义表述,本书集中了作者的研究成果,并参考与剖析了大量的重要相关文献,书中的内容既为读者研究程序语言语义打下了坚实的理论基础,同时也是读者开展工作的良好示范。

我们受托翻译此书,望能为国内计算机程序工作者对程序语言的认识与发展带来帮助。由于原著涉及面广,部分内容较深,译文错误之处在所难免,敬请批评指正。

译者于南京
2006年9月

丛书前言

理论计算机科学目前已经经历了几十年的发展。“经典的”自动机理论、形式语言和计算机复杂性等专题已被坚实地构建出来,它们对于其他理论工作与实践的重要性受到广泛认可。在技术进步的推动下,理论工作者迅速地扩展研究的领域,并且理论的进展与其对实践的影响之间的时延急剧缩短。现今广泛宣称在密码学和线性规划方面已有突破,并在程序设计语言语义、计算几何和高级数据结构方面有着稳固的进步。更新的富有成果的研究领域包括关系数据库、VLSL理论和并行分布式计算。随着所列的专题的不断扩充,人们越来越难以止于所取得的成就之上,愈显重要的是大量有意义的工作被精练和交流,以便于这些工作的进一步研究和应用。通过出版在计算机科学理论方面的综合读物和专著,这一套关于计算理论基础的丛书提供了一个场所,使得重要的研究专题能以其完整的形式表示出来,并以不同角度呈现在研究人员、学生和实践工作者的面前。

Michael R. Garey

Albert R. Meyer

前 言

本书提出了一个框架,用于分析程序设计语言的语法、操作和语义性质,该框架基于称为类型化 λ 演算的一种数学系统。 λ 演算的主要特色是对于函数和其他可计算的值的一种记法,以及一个等式逻辑和用于表达式求值的一组规则。本书围绕 λ 演算的一个序列进行组织,该序列逐步增添更为复杂的类型系统。它们用于分析和讨论相关的程序设计语言概念,重点研讨顺序语言,当然许多的技术和概念也可以应用到并发程序设计语言。

本书中最简单的系统是有时也被称为泛代数的一个等式系统,这个不带函数变量的逻辑可以被用来公理化和分析通常用于程序设计的许多数据类型。接下来的系统是带有函数类型和任选的笛卡儿积和不相交并的 λ 演算。一旦充实以递归定义,这样的语言提供了研究函数式程序设计操作和语义性质的有用框架。一旦结合代数数据类型,这个系统足以定义许多常用的程序设计语言。特别是有了内存位置和存储时,我们就可以研究命令式程序传统的公理、操作和指称语义。更先进的技术机制,诸如逻辑关系的方法、范畴论和递归定义类型的语义在中间的几章中论述。本书最后三章研究多态类型,连带讨论了抽象数据类型的说明形式和程序模块、类型适应性和类型推理。

必备基础以及与其他专题的联系

本书是为理论计算机科学、软件系统和数学专业的本科高年级或者研究生初始学习阶段而写作的,它同时也适用于高等研究或技术参考。唯一确实不可缺少的公认前提是“合适的数学成熟度”,绝大多数学生会发现对于形式逻辑、可计算性或计算复杂性理论和程序设计语言的已有的经验是很起作用的。一般说来,熟悉这些专题的学生只要达到如[AU92]的通用引论的程度或更高些,就应该有信心并继续攻读。为给可能的读者或教学人员更多的信息,与相关专题的主要联系概述如下。

数理逻辑

本书采用的 λ 演算系统秉承传统数理逻辑的许多特征,均有一个语法、一个证明系统和一个模型论。正因如此,逻辑的一般思想,诸如合适公式的定义、证明系统的可靠性与完备性,采用数学结构的表达式解释都被用上了。这些方面根据需要做了简短的介绍。一阶逻辑本身只用在证明程序性质的章节。这里假定对公式的概念已具有直观的理解。

可计算性和复杂性理论

可计算与不可计算函数之间的基本不同用于研究PCF(第2章),文中定义和使用了部分递归函数类并在两个章节的练习中引用了图灵机。在使用递归函数的Gödel计数法构造语义模型时(第4章),采用了少量取自递归论的补充概念,这一切应该是从包含通用图灵机和可计算函数不可判定性质的任何教程中知晓的。一定程度的基本递归论在课文中使用PCF加以研究,包括证明对于PCF的停机问题是不可用PCF编程来解决的简单练习。

程序设计

尽管不要求有专门的程序设计经验,但对于接触带有高阶函数的程序语言,如Lisp、Scheme或ML的学生而言,会发现更容易将理论与实践联系起来。为了提供一个表现类型化 λ 演算的一般方法,第2章包含了一系列的编程例子和技术。这提供了一些相关编程问题的一个自包含的概观。

范畴论

范畴论只出现在本书的较深的章节中。所有必备的定义均以例子提出和揭示,非数学专业读者先前未接触过范畴论的话,可以参阅其他读物。要是需要进度较慢的综合性介绍,读者可考虑一本为计算机科学工作者特别编写的初等引论,即[BW90, Pie91]。

样本课程的纲要

在表1到表3中给出了三个样本课程的纲要。第一个是入门性课程,它作为斯坦福大学CS 258课程讲授过多次。该课程所列的先行课,覆盖了在第2章到第6章中的核心专题,它们是半学期的自动机课程与可计算理论,和半学期的数理逻辑课程,但不深入讨论可靠性、完备性和模型论构造。CS 258课程已被本科生、系统和理论专业的硕士研究生和初期攻读的博士研究生顺利完成了。虽然斯坦福大学的课程在10周中教完,但也不难把课程扩展成15周的一个学期。扩展的选择有:(1)以较慢的进度来讲授所列专题;(2)加进关于代数重写系统的章节;(3)证明类型化演算的可靠性、完备性和其他性质;(4)从第9章到第11章中选择一些专题概要论述,也可以去掉命令式程序(第6章)以便增添一个或多个以上的选择。关于代数的这一章(第3章)严格说来对后面的专题并不重要,但泛代数却提供了一个机会能在相对简洁的数学背景下引进和回顾逻辑的概念。若学生已经学过较为严谨的数理逻辑本科课程,该章的这种做法就是多余的了。

表2中列出的第二个课程,是关于类型化 λ 演算和语义技术的数学性强一点的课程,更多谈的是技术性细节,很少涉及程序设计方面的动机。表3中列出的第三个课程讲授类型系统。它以类型化 λ 演算开始,接着讨论多态、类型适应性和类型推理。这三个互有交叉的课程涵盖了本书的绝大部分内容。

致谢与声明

许多人读过原稿并提供了有益的评论与建议。我们十分感谢M.Abadi, S.Abramsky, V.Breazu-Tannen, K.Bruce, L.Cardelli, R.Casley, P.L.Curien, P.Gardner, D.Gifford, D.Gries, C.Gunter, R.Harper, S.Hayashi, F.Henglein, B.Howard, P.Kanellakis, A.Kfoury, P.Lescanne, H.Mairson, I.Mason, A.Meyer, E.Moggi, N.Marti-Oliet, A.Pitts, J.Riecke, K.Ross, D.Sanella, P.Scott, D.Tranah, T.Uribe和选择斯坦福大学CS 258与CS 358课程的学生。特别的感谢要给助教My Hoang, Brian Howard和Ramesh Viswanathan,是他们帮助完成了课后练习和编写样本答案,其中有一小部分成为本书的例子。

本书几乎全部都建立在先前发表的研究工作之上,部分是作者自己的。当某结果是取自文献时,尽量引用原始资料以及相关的评述论文与书籍。当然,以这本书的规模的任何项目,总难免有错误与遗漏。此外,在转述和讲授任何原始材料时尽力所做的工作,或在本书重新做的证明之中无疑会留下错误之处。

John C. Mitchell
Stanford, CA

表 1 入门课程的纲要

样本入门课程

1. 函数式程序设计和类型化 λ 演算 (第 2 章)
 - (a) 布尔值、自然数、序偶和函数表达式; 利用不动点算子定义递归函数 (2.2 节)
 - (b) 公理语义、操作语义和指称语义的比较 (2.3 节)
 - (c) 归约的性质; 确定性符号的解释程序 (2.4 节)
 - (d) 程序设计的技巧、表达能和限度 (2.5 节)
2. 泛代数和代数数据类型 (第 3 章)
 - (a) 代数项、等式和代数 (3.1~3.3 节)
 - (b) 等式证明系统、可靠性和完备性 (3.4 节)
 - (c) 同态和初始性 (3.5 节)
 - (d) 数据类型的代数理论方面 (3.6 节)
3. 类型化 λ 演算的语义和递归 (第 4 章和第 5 章)
 - (a) 利用类型求取规则表示上下文相关文法 (4.3.1 节、4.3.2 节和 4.3.5 节)
 - (b) 通用模型、可靠性和完备性的总结 (4.5.1~4.5.4 节)
 - (c) 带不动点算子的类型化 λ 演算的域论模型 (5.1 节和 5.2 节; 若时间充裕可阅读 5.3 节和 5.4 节)
4. 命令式程序 (第 6 章)
 - (a) while 程序的文法; L 值和 R 值 (6.2 节)
 - (b) 结构化操作语义 (6.3 节)
 - (c) 利用带位置和存储类型、不动点算子的类型化 λ 演算给出的指称语义 (6.4 节)
 - (d) 部分正确性断言; 可靠性、相对完备性和样例证明 (6.5 节)

表 2 关于类型化 λ 演算的数学课程

关于语义和类型化 λ 演算的课程

1. 类型化 λ 演算的文法和证明系统
 - (a) 上下文相关文法和类型求取算法 (4.1~4.3 节)
 - (b) 等式证明系统和归约 (4.4 节)
 - (c) 利用不动点算子的递归 (2.2.2~2.2.4 节的概要, 2.2.5 节)
 - (d) 递归类型和显式提升 (2.6 节)
2. 类型化 λ 演算的模型理论
 - (a) 通用定义、可靠性和完备性 (4.4.1 节, 4.5.1~4.5.6 节)
 - (b) 论域 (5.1 节, 5.2 节)
 - (c) 朴素集 (5.5 节, 5.6 节)
3. 逻辑关系
 - (a) 定义和基本引理 (8.1 节, 8.2 节)
 - (b) 证明论的结果: 完备性、正则化和汇聚 (8.3 节)
 - (c) 用于集合论层次、朴素集和论域的完备性理论 (8.4 节)
4. 范畴理论和递归类型
 - (a) 范畴、函子和自然转换 (7.1 节, 7.2.1 节, 7.2.2 节)
 - (b) 笛卡儿闭范畴和类型化 λ 演算 (7.2.3~7.2.6 节)
 - (c) 不带良不动点的范畴的一个例子: Kripke λ 模型 (7.3 节)
 - (d) 递归类型的域模型 (7.4 节)

表 3 关于类型理论的课程

关于类型理论的课程

1. 简单类型化 λ 演算

- (a) 上下文相关文法和类型求取算法 (4.1~4.3 节)
- (b) 等式证明系统和归约 (4.4.1 节, 4.4.2 节)

2. 多态

- (a) 多态类型的引言 (9.1 节)
- (b) 预知多态 (9.2 节)
- (c) 不可预知多态的性质 (9.3.1~9.3.4 节)
- (d) 数据抽象和存在类型 (9.4 节)
- (e) 通积、通和与程序模块 (9.5 节)

3. 衍类型

- (a) 衍类型的基本的语法论点, 等式推理, 包含和转换解释 (10.1~10.4 节)
- (b) 记录、递归类型和记录作为对象 (10.5 节)
- (c) 带衍类型约束的多态 (10.6 节)

4. 类型推理

- (a) 类型推理和去除函数 (11.1 节)
 - (b) 利用统一为简单类型 λ 演算进行类型推理 (11.2 节)
 - (c) ML 风格的多态声明 (11.3 节)
-

目 录

第 1 章 引言	1
1.1 模型程序设计语言	1
1.2 λ 记法	2
1.3 等式, 归约和语义	4
1.3.1 公理语义	4
1.3.2 操作语义	5
1.3.3 指称语义	5
1.4 类型和类型系统	6
1.5 记法和数学约定	8
1.6 集合论基础知识	9
1.6.1 基础	9
1.6.2 关系和函数	12
1.7 语法和语义	14
1.7.1 目标语言和元语言	14
1.7.2 文法	14
1.7.3 词法分析和语法分析	15
1.7.4 数学解释示例	17
1.8 归纳法	18
1.8.1 自然数归纳法	18
1.8.2 表达式和证明上的归纳法	21
1.8.3 良基归纳法	26
第 2 章 PCF 语言	30
2.1 引言	30
2.2 PCF 语法	31
2.2.1 概述	31
2.2.2 布尔值和自然数	31
2.2.3 配对及其函数	34
2.2.4 声明和语法惯用形	37
2.2.5 递归和不动点算子	40
2.2.6 PCF 语法总结和精选实例	43
2.3 PCF 程序及其语义	45
2.3.1 程序和结果	45

2.3.2	公理语义	46
2.3.3	指称语义	48
2.3.4	操作语义	49
2.3.5	由各种形式语义定义的等价关系	51
2.4	PCF 归约和符号解释程序	52
2.4.1	不确定性规约	52
2.4.2	归约策略	56
2.4.3	最左优先和懒归约策略	57
2.4.4	并行归约	60
2.4.5	积极 PCF	61
2.5	PCF 编程样例, 表达能和限度	64
2.5.1	记录和 n 元组	64
2.5.2	搜索自然数	65
2.5.3	迭代和尾递归	67
2.5.4	完全递归函数	70
2.5.5	部分递归函数	72
2.5.6	并行操作的不可定义性	75
2.6	PCF 的变体和扩展	81
2.6.1	扩展的概述	81
2.6.2	<i>unit</i> 与求和类型	81
2.6.3	递归类型	84
2.6.4	提升类型	88
第 3 章	泛代数及代数数据类型	97
3.1	引言	97
3.2	代数规范概述	98
3.3	代数, 基调和项	99
3.3.1	代数	99
3.3.2	代数项语法	99
3.3.3	代数和项的解释	101
3.3.4	代入引理	104
3.4	等式, 可靠性和完备性	105
3.4.1	等式	105
3.4.2	项代数和代入	106
3.4.3	语义蕴涵和等式证明系统	107
3.4.4	完备性的形式	115
3.4.5	同余, 商和演绎完备性	115
3.4.6	非空类子和最小模型性质	118
3.5	同态和始代数	119

3.5.1	同态和同构	119
3.5.2	始代数	121
3.6	代数数据类型	125
3.6.1	规范和数据抽象	125
3.6.2	始代数语义和数据类型归纳	127
3.6.3	例子和错误值	131
3.6.4	错误值的其他解决方法	135
3.7	重写系统	135
3.7.1	基本定义	135
3.7.2	汇聚性和可证相等性	138
3.7.3	终止性	140
3.7.4	临界对	143
3.7.5	左线性非重叠重写系统	148
3.7.6	局部汇聚, 终止和完备性	151
3.7.7	代数数据类型的应用	153
第 4 章	简单类型化 λ 演算	156
4.1	引言	156
4.2	类型	157
4.2.1	语法	157
4.2.2	类型的解释	157
4.3	项	159
4.3.1	上下文相关语法	159
4.3.2	$\lambda \rightarrow$ 项的语法	160
4.3.3	带有积、和的项及其相关类型	165
4.3.4	公式与类型的对应	166
4.3.5	类型求取算法	169
4.4	证明系统	171
4.4.1	等式和理论	171
4.4.2	归约规则	178
4.4.3	具有附加规则的归约	180
4.4.4	一致性和保持性的证明论方法	182
4.5	Henkin 模型, 可靠性和完备性	186
4.5.1	通用模型和项的含义	186
4.5.2	作用结构, 外延性和框架	187
4.5.3	环境模型条件	188
4.5.4	类型和等式可靠性	192
4.5.5	不带空类型的 Henkin 模型的完备性	195
4.5.6	带有空类型的完备性	197

4.5.7	组合子和组合模型条件	198
4.5.8	组合代数和 λ 代数	200
4.5.9	其他类型的 Henkin 模型	201
第 5 章	类型化λ演算模型	204
5.1	引言	204
5.2	域论模型和不动点	204
5.2.1	递归定义和不动点算子	204
5.2.2	完全偏序, 提升和笛卡儿积	206
5.2.3	连续函数	209
5.2.4	不动点和完全连续层次	212
5.2.5	PCF 的 CPO 模型	218
5.3	不动点归纳	223
5.4	计算适当性和完全抽象	227
5.4.1	近似原理和计算适当性	227
5.4.2	带并行操作的 PCF 的完全抽象	231
5.5	递归理论模型	237
5.5.1	引言	237
5.5.2	朴素集	239
5.5.3	完全递归层次	241
5.6	部分等价关系和递归	244
5.6.1	类型的部分等价关系解释	244
5.6.2	部分组合代数的一般化	246
5.6.3	提升, 部分函数和递归	249
5.6.4	递归和固有序	251
5.6.5	有效 CPO 的提升, 积和函数空间	256
第 6 章	命令式程序	260
6.1	引言	260
6.2	while 程序	261
6.2.1	L 值和 R 值	261
6.2.2	while 程序的语法	262
6.3	操作语义	263
6.3.1	表达式中的基本符号	263
6.3.2	位置和存储	263
6.3.3	表达式求值	264
6.3.4	命令的执行	265
6.4	指称语义	269
6.4.1	带有存储的类型化 λ 演算	269
6.4.2	语义函数	271

6.4.3	操作语义和指称语义的等价性	275
6.5	关于 while 程序的前-后断言	277
6.5.1	一阶和部分正确性断言	277
6.5.2	证明规则	278
6.5.3	可靠性	283
6.5.4	相对完备性	284
6.6	附加程序构造的语义	288
6.6.1	概述	288
6.6.2	带有局部变量的模块	288
6.6.3	过程	294
6.6.4	组合程序块和过程声明	295
第 7 章	范畴和递归类型	299
7.1	引言	299
7.2	笛卡儿闭范畴	299
7.2.1	范畴论与类型化语言	299
7.2.2	范畴, 函子和自然变换	300
7.2.3	笛卡儿闭范畴的定义	307
7.2.4	可靠性和项的解释	314
7.2.5	Henkin 模型作为 CCC	325
7.2.6	含义函数的范畴刻划	327
7.3	Kripke λ 模型和函子范畴	329
7.3.1	概述	329
7.3.2	可能世界	329
7.3.3	作用结构	329
7.3.4	外延性, 组合子和函子范畴	331
7.3.5	环境和项的含义	333
7.3.6	可靠性和完备性	335
7.3.7	Kripke λ 模型作为笛卡儿闭范畴	336
7.4	递归类型的域模型	338
7.4.1	一个启示性的例子	338
7.4.2	图, 锥和极限	341
7.4.3	F 代数	343
7.4.4	ω 链和初始 F 代数	345
7.4.5	O 范畴和嵌入	348
7.4.6	余极限和 O 余极限	350
7.4.7	局部连续函子	353
7.4.8	该通用方法的例子	355

第 8 章 逻辑关系	359
8.1 逻辑关系概述	359
8.2 作用性结构上的逻辑关系	359
8.2.1 逻辑关系的定义	359
8.2.2 基本引理	361
8.2.3 部分函数和模型的理论	365
8.2.4 逻辑部分等价关系	366
8.2.5 商和外延性	366
8.3 证明论的结果	369
8.3.1 Henkin 模型的完备性	369
8.3.2 正则化	371
8.3.3 汇聚和其他归约性质	373
8.3.4 有 λx 和附加操作的归约	377
8.4 部分满射和特殊模型	385
8.4.1 部分满射和完整的古典层次	385
8.4.2 完整的递归层次	386
8.4.3 完整的连续层次	388
8.5 表示独立性	389
8.5.1 动机	389
8.5.2 实例语言	390
8.5.3 普通的表示独立性	392
8.6 逻辑关系的推广	393
8.6.1 引言	393
8.6.2 启示性例子：全偏序和 Kripke 模型	394
8.6.3 寻求原像体和关系	399
8.6.4 与逻辑关系的比较	402
8.6.5 一般情形和应用到特殊范畴	404
第 9 章 多态与模块性	407
9.1 引言	407
9.1.1 概述	407
9.1.2 类型作为函数实参	407
9.1.3 通积与通和	411
9.1.4 类型作为规范	412
9.2 预知多态演算	414
9.2.1 类型和项的语法	414
9.2.2 与其他多态形式比较	418
9.2.3 等式证明系统和归纳	421