# 形式语义学基础与形式说明

（第二版）

屈延文　编著

◆中国信息安全测评中心自然科学基金项目内容组成之一
◆站在软件立场上讨论计算机科学的理论及其应用
◆详细给出形式语义学的基础理论框架
◆理论与软件实践相结合

科学出版社
www.sciencep.com

# 形式语义学基础与形式说明

（第二版）

屈延文　编著

科学出版社

北京

## 内 容 简 介

本书第一版是20世纪80年代国家教委计算机软件专业教材编委会推荐教材之一。本书详细地给出了形式语义学的基础理论框架，但它并不是一本纯理论的教材，而是一本理论与软件实践相结合的教材。

全书共分十章。介绍了指称语义学、代数语义学、操作语义学与公理语义学的基本内容及其应用，并介绍了并发程序设计语言各流派的语义模型和新一代计算机计算模型的理论问题。例如Curry的组合逻辑，Martin-Löf的直觉主义数学的讨论都是近代计算机理论较重要的基础内容。

本书内容丰富，重点突出，并配有大量习题，可作为高等院校电子信息、计算机科学专业本科高年级学生、研究生的教材，也可供信息技术人员和计算机软件设计、工程人员参考。

**图书在版编目(CIP)数据**

形式语义学基础与形式说明(第二版)/屈延文编著. —北京：科学出版社，2009

ISBN 978-7-03-026238-7

Ⅰ.形… Ⅱ.屈… Ⅲ.形式语义学 Ⅳ.TP301.2

中国版本图书馆CIP数据核字(2009)第232458号

责任编辑：杨 凯 刘晓融 / 责任制作：董立颖 魏 谨

责任印制：赵德静 / 封面设计：许思麒

北京东方科龙图文有限公司 制作

http://www.okbook.com.cn

科学出版社 出版

北京东黄城根北街16号

邮政编码：100717

http://www.sciencep.com

北京天时彩色印刷有限公司 印刷

科学出版社发行 各地新华书店经销

*

1989年12月第 一 版 开本：B5(720×1000)

2010年3月第 二 版 印张：32

2010年3月第一次印刷 字数：625 000

印数：1—3000

**定 价：59.00元**

(如有印装质量问题，我社负责调换)

# 第二版序

屈延文先生请我为他的著作《形式语义学基础与形式说明》的再版书写一个序，感到很荣幸。这本书是作者在上世纪80年代，为我国高等院校计算机和软件专业研究生写的关于计算机科学方面的教材。该书在培养我国计算机科学高素质的研究人才和加强我国计算机科学理论教育方面发挥了作用。作者在国家的计算机技术研究机构中工作，长期从事计算机软件、计算机科学和技术的研究与实践，并热衷于计算机与信息化的理论研究和创新。作者在网络世界的行为学理论与应用方面的工作，例如他的著作《软件行为学》、《银行行为监管》和《银行行为控制》及其在国内信息安全产业推动代理技术中的应用和发展，给我留下了深刻的印象。他是我国网络世界行为学学科与理论的发起者和主要推动者。读者可以从本书内容中看到，作者之所以能够不断探索与研究新问题，是由于他具有深厚基础的知识结构和理论联系实际的科学作风。

相信本书会引起从事信息科学技术工作和网络空间安全工作人员的极大兴趣，并有所裨益。

何德全

2009年9月

# 第二版前言

本书是计算机专业人员和大学软件、计算机科学和信息化科学专业高年级学生以及硕士研究生或博士研究生了解计算机科学理论在软件工程应用的专门书籍。作者站在软件的立场上来讨论计算机科学(尤其是形式语义学)的理论及其应用。虽然,我们比较详细地介绍了形式语义学基本理论框架,但它不是一本纯理论的书籍,而是一本理论联系实际的书籍;因此,在叙述中不苛求理论上的严密性。

作者在1980年,由于准备Ada语言的编译程序和环境工作以及对软件工程理论的关注,开始对形式语义学理论与实践进行研究。当时出版本书的目标是全面提高软件重用、软件自动生成和软件智能化的理论和技术水平。目前看来,这个目标现在仍然没有过时。在20世纪80年代初,作者用三册油印的讲义,经过多年在北京大学、北京航空学院(北京航空航天大学)、中国科学院研究生院、北京信息工程学院和华北计算技术研究所为研究生讲授此课;北京之外,在国防科技大学、武汉大学和其他大学也讲授过此课程。在多年研究和讲课的基础上,本书的内容更加完善。20世纪80年代后期科学出版社正式出版了《形式语义学基础与形式说明》一书。作为第一版,该书正式出版已经有20多年了,曾经两次印刷,多年来一直有人不断向作者寻求本书。在21世纪,信息化发展带来了科学发展的强大新动力,尤其是信息化科学学科的提出,冲破了经典计算机科学研究的范围,其中信息化和网络世界的安全是最突出的领域。信息安全不仅需要实施更加严格的产品保证计划,同时要求信息安全测评事业具有逐步达到高级别的安全测评能力,尤其是提高结构性和形式化验证能力,依此来推动我国信息系统产品质量的全面提升。所以,必须对信息化与安全产业和测评机构进行结构性和形式化验证方法的培训。再版本书是实施这种高素质培训的教材建设,再版工作被列为中国信息安全测评中心的自然科学基金项目内容之一。

本书再版还有更广泛和深刻的意义。在20世纪90年代,作者开始特别关注信息化体系结构和信息安全的理论问题。显然,信息化与网络世界中的大量问题,例如互操作性和安全性问题,已经不可能从经典的计算机科学理论中得到帮助和指导。信息化的体系结构已经不是计算机的体系结构,不是软件的体系结构,也不是应用系统的体系结构。信息化的体系结构概念要比这些方面的体系结构复杂得多。信息化体系结构的实践告诉我们,要划分运营、系统和技术三个视图来研究体系结构。经过多年研究,作者认为,运营体系结构,从理论的高度看,其核心的概念范畴是主体和行为,包括人类信息化行为和网络世界中的虚拟主体和行为。

网络世界传统的系统体系结构方法给我们提供的知识结构,也仅仅是对过去的总结,不能指导我们面临新的问题,例如我们所面临的互操作性以及如何充分、有效地利用资源。其中,互操作性在信息化的历程中也经历了三个发展阶段。而在20世纪80

年代，采用的方法是面向过程，通过通信和过程调用解决互操作问题。在20世纪90年代，采用的是面向对象、可视化和中间件技术，通过互操作性标准的方法来解决信息交换和共享问题。在21世纪前10年，开始研究面向主体和代理技术，通过多代理组织和系统以及它们的行为，例如代理信息交换平台和服务递交平台，来解决代理组织协同、群体业务服务等问题。并且，通过代理与代理组织协同、递交、移动、生长等互操作性技术实现了网络世界服务的本地化、个性化、规模化、关联化和代理化自治服务新目标。互操作性技术的新发展进程，又一次把我们引向了主体与行为学理论方法。

所谓充分利用资源，传统的软件工程学说告诉我们，就是采用软件重用、资源共享和引用等许多方法。但是，传统的软件工程学说主要在讲一个软件工程师如何在一张白纸上写字、画画，告诉我们如何开发一个新的软件系统和产品。然而，信息化的进程已经有几十年，人类已经开发了太多的具体业务功能的软件产品。好比茫茫的信息大草原已经不再是荒原，不仅有茂密的植物，还充满了各种食草动物。以至于在绝大多数情况下，只要我们合理使用各种软件资源，就可以完成我们需要的业务。这种状态告诉人们，新的信息化的体系，它们处理的对象已经不是信息大草原中的植物（信息），而是那些“吃信息的食草动物”（信息系统）。在许多情况下，不需要再从植物中获取营养，而是直接吃多种动物，吸取它们的主体资源（利用、调度、组合它们的资源）和行为，就足够了。这一点对信息安全产业来说尤其重要，它们的系统与产品是信息化生态体系中的“食肉动物”。在新软件工程学方面，我们将面向主体的程序设计和行为开发方法，而不仅仅是程序代码的开发方法（这是多么巨大的变化！）。新软件工程学的研究，把我们更加深入地引向了主体与行为学理论方法。

网络世界的安全问题同样把我们引入了主体与行为学理论方法。安全问题从传统的软件工程理论来讲是产品的正确性、质量、漏洞、缺失等问题。对于这类原因引发的安全我们称之为“脆弱性安全”。解决的办法当然还是产品的保证计划。但是在信息化新条件下，产生了非脆弱性引起的安全问题，即互操作性和关联性引发的安全问题。我们把这样的安全称作“结构性安全”或“体系结构安全”。面对结构性安全问题，传统的软件工程方法不能解决，也不是产品保证计划能够解决的。既然安全问题由互操作性和关联性而发，解决的办法依然是通过互操作性和关联性。结构性安全的理论基础同样是主体和行为学。

读者也许已经看到，传统计算机科学的本质是语言学的理论方法，在计算机科学中称作形式语言的理论方法。可是，在信息化和网络世界中遇到的各种问题的核心概念是主体和行为，属于行为学理论方法。在网络世界中，我们要研究人类信息化行为和各种虚拟主体和行为，可以称作“形式行为学”问题。所谓形式行为学就是要区别某些生物和人类的一些行为特点和模式，如同形式语言是为了区别人类的自然语言一样。人们会问，如何研究行为？一种非常重要的方法是“行为的语言表达”。有了行为的语言表达后，就可以研究“语言表达中的行为”。显然，其中最重要的是解决行为的数学语言表达和数学语言表达中的行为两个大问题。但是，人类发明的科学范畴和整个数学体系中，居然没有面向主体的数学。无论是集合、函数、代数、几何、逻辑等各种

数学体系中，都没有面向主体的，都是面向客体的。于是我们需要全面改进数学体系，使之面向主体和行为学的研究。当然，在解决了“行为的数学语言表达和数学语言表达中的行为”后，下一步需要解决的是信息化的形式化方法以及行为的形式语言表达和形式说明。所不同的是，语义范畴概念发生了很大变化。例如，需要研究各种行为模式的预期语义、态势语义和预期与态势之间的评估语义等。关于这些研究的新进展，在作者关于信息化行为学理论基础和形式化方法的新著作中有特别详细的介绍。本书介绍的理论和方法，在行为学理论的研究中依然发挥着基础作用。

总之，通过上面的论述，我们又回到了形式语言和形式语义学。可以有一个结论：在21世纪，形式语义学和形式化方法不仅是计算机科学的基础课程，同时还是信息化科学的基础课程。该书对未来计算机科学和信息化科学的研究与发展依然具有指导和帮助作用。

本书再版时，作者特别怀念吴允曾先生、唐稚松院士、陈火旺院士的友谊和帮助。作者感激杨芙清院士、周巢尘院士和杨天行先生曾经给予的支持和帮助。作者作为中国信息安全产业商会常务副理事长和中国信息安全测评中心的顾问，特别感谢中国信息安全测评中心主任和商会理事长吴世忠同志一贯的支持与帮助。感谢中国信息安全测评中心常务副主任和商会秘书长王贵骄同志、高新宇副主任的支持和帮助。

由于多年的友谊，作者特别请何德全院士为本书写第二版序，在此表示由衷感谢。作者真诚感谢刘晓融女士为本书第二版工作付出的努力。感谢学生张艳在第二版校对工作中给予的帮助。感谢作者工作室的李明生先生、刘玉林、林洁女士和商会副秘书长梁进女士多年的帮助、支持和本书第二版时所付出的努力。

屈延文

2009年8月

# 第一版前言

本书是计算机软件专业人员，大学软件专业本科高年级学生及研究生了解计算机科学理论在软件工程中应用的专门书籍。作者是站在软件的立场上来讨论计算机科学(尤其是形式语义学)的理论及其应用的。虽然，我们比较详细地介绍了形式语义学的基本理论框架，但它不是一本纯理论的书籍，而是一本理论联系实际的书籍。因此，在叙述中不苛求理论上的严密性。

早在1980年，作者在做Ada语言的准备工作时，就已开始对形式语义学的理论与实践进行研究。在研制Ada语言的编译程序时，我想到的第一个问题是，一本很厚的Ada语言文本，放在每一个程序设计人员面前，你怎么知道他们/她们看懂了Ada文本呢？怎么能相信每一个人能正确地归纳Ada文本的语义呢？如何知道编译程序的设计基础是可靠的呢？从这一点来说，没有Ada语言的形式定义是不行的，这一点可以从已有的文献中得到支持。

在研制Ada编译程序时，我们制定了计算机辅助管理、计算机辅助设计、严格质量管理及严格验收的设计原则。在当时，我们还认为编译程序的语义分析采用属性文法说明及属性文法的语义计算器是比较理想的方案。为了提高程序质量，提出了比较高的程序抽象要求，要求基于抽象数据类型的class(package)程序设计方法，要求程序结构有最好的层次分解性与模块分解性。所有这些，都促使我去写一本形式语义学的讲义，以提高Ada编译程序设计人员的素质。在当时，我们提出，指称语义学、属性文法及抽象数据类型应作为研制Ada编译程序的最基本理论准备。

后来，由于开展了对新一代计算机的研究，我们将形式语义学的研究与新一代计算机计算模型的研究结合起来了。在这方面最为重要的理论研究是对Curry的组合逻辑与Martin-Löf的直觉主义数学的讨论。

作者在不断完善本书(原是一本油印讲义)的过程中，曾受到过多方面的支持与帮助。原北京大学二分校(现在改名为信息工程学院)校长杨天行同志，最早支持我在该校讲授此课。后来，在北京大学计算机科学系主任杨芙清教授的支持下，在北京大学两次为研究生班与学位研究生讲授此课。同时，还在北京航空学院、华北计算技术研究所讲授此课。为使本书达到比较完善的程度，在编写过程中，曾得到中国科学院软件研究所唐稚松教授和周巢尘教授多方面的帮助，他们曾向我提供了许多珍贵的材料。还应当提到的是已故的北京大学吴允曾教授，他也曾给过我多方面的帮助。在本书出版之际，特向他们表示衷心的感谢。

屈延文

1987年7月

# 目　录

# 第1章 引 论

在这一章，我们将介绍形式语义学的基本概念和形式语义学的流派及分类。

## 1.1 形式语义学

一位中国人学习英语，他对英语语义的理解是通过英语的相应汉语的解释来进行的。就是对汉语，他对未知语言成分意义的理解也是通过已知语义的语言成分的注释而达到的。显然，在对自然语言理解的过程中，在说明对象语言的语义时，需要一个已知语义的语言，并建立这两种语言的对应关系，从而得到对于对象语言语义的理解。换句话说，已知语义的语言称之为说明语言，当然，说明语言有自己的语法及语义。令说明语言的语句是 $s$，其语义记为$[\![s]\!]$，而对象语言的一个语句 $r$，语句 $r$ 的语义记为$[\![r]\!]$。$[\![r]\!]$与$[\![s]\!]$的关系就是对象语言的注释。自然语言之间的映射关系是通过语法书及对照字典来实现的，虽然其中存在着很大的不确定性。

什么叫做形式语义学呢？

形式语义学是对形式语言及其程序采用形式系统方法进行语义定义的学问。在定义形式语言时，流行采用 BNF 定义语言的文法。如何解释用 BNF 定义的语言的语义？这个问题就是形式语言研究的对象之一。一个逻辑系统，例如一阶谓词演算，就可以被看成一个形式语言系统，那么这个逻辑语言的语义研究，也就是形式语义研究的对象。形式语言不仅是串文法，还有图文法(graph grammar)及树文法等，这些语言的语义研究，也是形式语义研究的对象。一个程序也有语法与语义两个方面，语法在程序中主要表现为程序结构，而语义在于对语法特性适用的域的解释。由于研究形式语言的语义方法不同，形式语义学大致可以分成如下几个分支：

- 指称语义学(denotational semantics)。
- 代数语义学(algebraic semantics)。
- 操作语义学(operational semantics)。
- 公理语义方法(axiomatic semantic approach)。

我们将在下面分别介绍这些语义学的基本点。

一个程序设计语言有两个最重要的概念：论域及辖域。在本书的讨论中将特别强调这两点。

当代的程序设计语言，除纯表达式语言之外，一般都包括三个语法范畴：

- 声明范畴。
- 命令范畴。
- 表达式范畴。

声明范畴的成分在于建立及改变环境；命令范畴的成分（即语句世界的成分）在于执行并改变状态；而表达式范畴的成分在于计算产生值。在语言的声明范畴中，又主要包括作用域及可见性两个概念。在命令范畴中的原子成分是赋值语句，GOTO 语句的存在主要用于改变程序的执行顺序，也就是说，具有 GOTO 语句的语言程序，其书写顺序与执行顺序是不同的。在表达式范畴中，函数定义（或称函数抽象）与函数施用是两个基本概念。如果一个表达式不仅计算产生值，而且还改变状态，我们称这个表达式有副作用（side-effect）。

在 Von Neumann 计算机体系中，程序设计语言还引入了地址概念，致使表达式计算产生的值与地址联系起来。如果使用这个值，则必须引用置有该值的内存的地址，我们称这样的程序是依赖于环境的，因为计算的结果与所在存储器的位置有关。

如果一个语言没有命令范畴，它只有表达式范畴及表示类型与对象的声明范畴，而且这种语言中的每一个函数都不依赖于环境而存在，那么它所计算的结果与所在存储器的位置无关，只要满足了函数定义域则可以处处施用，绝无副作用，那么称这样的语言为施用型语言（applicative language）。

为什么要研究形式语义学呢？

理论研究的最强大的动力是发展生产，促进社会的进步。软件生产长期停留在手工劳动的状态，生产力很低。像所有其他工业发展那样，软件产业也要追求自动化生产，大规模生成，高效率及高质量生产的目标。

为了发展软件生产力及保证软件生产的质量，软件工程方法被广泛地研究并得到普遍的应用。软件工程方法一般化分为如下几个阶段：

- 软件要求。
- 软件说明。
- 程序设计。
- 测试。
- 维护。

保证软件正确性，能正确地写出软件说明（包括功能说明、系统说明、结构说明、测试说明等）是十分关键的。正确的软件说明是程序设计的正确性的重要保证。怎样才能正确地写出说明呢？形式语义学方法是其中的最重要的方法之一。

软件生产的发展并不会满足于写出正确的程序说明。为了提高软件生产力，最有效率的措施是

- 提高软件重复使用能力。
- 发展软件自动生成技术。

发展软件自动生成技术，必须向计算机说明“做什么”（What to do）的语义及“如

何做”(How to do)的语义。为了使计算机能够理解程序员说明的内容,形式语义说明方法是一个必由之路。提高软件重复使用能力,采用建立在类别代数理论基础之上的抽象数据类型形式方法(代数方法)是一个基本途径。形式语义学的重要性是不言而喻的。

正是由于对软件工程的研究,才使人们认识到程序设计必须从 Von Neumann 计算机中解放出来,从而提出研制新一代机的计划。也正是由于研制软件说明语言及软件自动生成,发现一阶谓词演算完全构造程序的困难性,使人们又重新认识数学并促进了构造数学的发展。而非 Von Neumann 计算机的研制及构造数学的发展,既大大丰富了计算机科学的研究内容,又促进了计算机与软件的发展。

研究形式语义学,将会大大提高我们对程序设计语言本质的理解。从某种意义上来说,它会使我们对软件的认识有一个本质性的飞跃。

为什么会出现这样四个语义学的研究领域?这主要是有如下的分成四派的程序计算模型,它们是

- 图灵机模型。
- 谓词演算模型。
- 递归函数论模型。
- 代数模型。

这些模型与计算机模型、程序设计语言有表 1.1 所示的关系。

**表 1.1**

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 计算模型 | 图灵机模型 | 谓词演算模型 | 递归函数论模型 | 代数模型 |
| 硬件系统 | Von Neumann 机器 | | 组合逻辑计算机 | |
| 程序设计语言 | 状态语言 PASCAL, Ada 等 | 逻辑语言 (如 PROLOG) | 函数施用型语言 (如 LISP) | 类型程序设计语言 |

## 1.2 指称语义学

要想了解指称语义学,必须了解什么是指称。例如一个语法符号串 $fxy$,其中 $f$, $x$, $y$ 都是语法符号,是我们需要了解语义的语言的句子单位。显然,仅从这个句子中我们是无法知道它的语义的。如果我们有如下的域(数学对象的域),一个域是函数域,我们记为[Nat. Nat→Nat];在这个函数域中,也有它的值或对象,例如 plus, mult, minus 等。另一个域是一个自然数的值域,即 Nat=$\{0,1,2,\cdots\}$。如果符号 $f$ 的语义被记为$[\![f]\!]$,如令$[\![f]\!]$=plus: Nat, Nat→Nat,我们就说 plus 是 $f$ 的指称。同样的道理,我们也可以注释语法符号 $x$, $y$ 的自然数指称。

因此,指称语义是采用形式系统方法,用相应的数学对象(例如 set, function 等)

对一个即定形式语言的语义进行注释的学问。指称语义还可以被解释为:存在着两个域,一个是语法域,在语法域中定义了一个形式语言系统;另外一个是数学的域(或称之为已知语义的形式系统)。用一个语义解释函数,以语义域中的对象(值)来注释语法域中定义的语言对象的语义,即为指称语义。由于指称语义的理论支持是论域方程,在函数空间解这些论域方程需要不动点理论,于是也有人说:"指称语义就是不动点语义"。

指称语义的研究首先是由 D. Scott 及 C. Strachey 等开始的。

指称语义学的基础包括论域理论和 λ-演算,这些我们将在第 2 章中介绍。论域理论是在偏序集合上展开的,我们还将讨论完全偏序、函数单调性、函数连续性、泛函不动点等理论。将偏序关系限制在 Egli-milner 序及 smyth 序上进行讨论,从而引出了研究不确定性的幂域(powerdomains)理论。在本书的第 3 章,我们讨论程序设计语言的指称语义。在第 4 章,我们给出几个较大的程序设计语言的指称语义的例子。我们希望读者学完这些内容之后,能够学会读写指称语义及指称语义的证明技术。

## 1.3 代数语义学

代数语义学是另一种数学语义的注释方法。用代数方法对形式语言系统进行语义注释的语义学,便称之为代数语义学。例如,我们可以定义一个逻辑系统,当然这个逻辑系统也是一个语言系统,我们就可以用范畴论的抽象代数概念对这个语言进行语义注释。又例如,递归程序设计语言,我们可以用所谓的树文法描述递归程序设计语言程序的语法结构。研究这种程序结构在语义域 I 的注释下,其代数特性(如分配律、交换律、结合律、等价性、条件等价性等)的变化情况,也是一种代数语义学方法。再例如,程序结构框图就可以用所谓的图的重写(rewriting)系统(图范畴)进行注释其结构方面的语义。

根据代数语义学的发展状况,我们将在本书的第 5 章介绍代数语义学的基础,它们包括范畴论、图范畴、类别代数理论、抽象数据类型的说明与实现等内容。

对于函数型程序设计语言,也许读者已经知道 McCarthy 的 LISP 语言,这是一个纯表达式语言。而表达式则可以用一个执行树来描述,可以借助于递归函数论及代数语义学的知识来理解 LISP 语言。Buckus 在 1978 年的 Turing 奖演说中提出了 FP (functional programming)程序设计思想,即将一般的函数施用形式 $f(x)$(LISP 语言就是采用这种施用形式的),定义成一种新的施用形式,即 $F:\langle x\rangle$。例如一个表达式,采用 LISP 风格,则为

$$f(x,g(h(y),z))$$

其树结构如图 1.1 所示。

如果把图 1.1 树结构的叶子结点去掉,剩下的树不再有变量,如果用 $F$ 来表示,将它施用三元组$\langle x,y,z\rangle$之后与 $f(x,g(h(y),z))$有相同的计算结果。可见 $F:\langle x,y,$

z)本质意义在于将函数进行无变量(无形式参数)的抽象,也巧,Curry 的组合逻辑(combinatory logic)可以解决这个问题,这就是组合逻辑被关注的原因。组合逻辑有可能成为下一代计算机的模型,我们将在第 7 章比较详细地介绍组合逻辑。

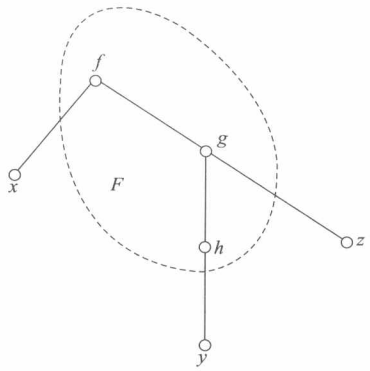


图 1.1

## 1.4 操作语义学

操作语义是用机器模型语言来解释语言语义的,一般说来它与编译程序有直接关系。在定义操作语义时,被采用的机器模型是 SECD 机器。Landin(1964 年)第一个在 SECD 机器上定义了 λ-表达式的操作语义,虽然现在看来这种定义方法并不那么美,也不如后来由 G. Plotkin 采用的归约系统方法清楚,但 Landin 的工作是有重大意义的。

操作语义学是所有语义学派别中最早出现的,在 20 世纪 80 年代后期又得到了新的发展。

操作语义学除定义"做什么"之外,主要是定义"怎么做",从这个意义上讲,属性文法属于操作语义学范畴。我们在本书中将介绍属性文法的基本概念,属性文法的分类,如何用属性文法进行编译程序设计及定义语言。

## 1.5 公理语义方法

公理语义方法是把程序设计语言视为一个数学对象,建立它的公理系统,从而使程序设计语言有坚实的逻辑基础。在这方面从事有意义工作的人不少,例如 Floyd, Z. Manna, Hoare, Dijkstra 等。

由于 Martin-Löf 的研究成果,使我们比较清楚地认识到,一阶谓词演算因其不完全显式性及不完全构造性,并不十分适合于计算机与程序设计语言。为根本性地解决软件问题,除 Buckus 所说从 Von Neumann 计算机中解放出来之外,Martin-Löf 又向我们说,应当创立更适合于计算机与程序设计语言的新数学。而 Martin-Löf 的类型论是构造数学的最新流派。不管别人怎么看待构造数学,而我们却认为从计算机科学

的角度来说，构造数学派是应当受到欢迎的。计算机科学家对构造数学感兴趣的道理是完全可以理解的。

除此以外，模态逻辑、多值逻辑、非单调逻辑系统、线性逻辑的研究在20世纪80年代也十分活跃。

## 1.6 形式说明语言

最早用于程序设计的形式说明语言是VDL(vienna definition language)，这个语言曾被用于定义PL/1的操作语义。随着软件工程的发展，形式说明语言也多了起来。例如，属性文法定义语言ALADIN，为Ada语言的编译程序定义过语义的VDM的元语言，还有ALPHARD抽象数据类型的定义语言，以及将逻辑与函数组合在一起的LCF语言、AFFIRM语言、SOL语言、FUN语言、OBJ2语言、PL/CV3语言等。在20世纪90年代产生的Z语言，也是一个被关注的形式说明语言。

由于篇幅及应用方面的原因，在本书中只向读者较详细地介绍VDM的元语言。