



陆汝钊 著

计算系统的形式语义 上

FORMAL SEMANTICS OF COMPUTING SYSTEMS



清华大学出版社

陆汝钊 著

计算系统的形式语义 ①

FORMAL SEMANTICS OF COMPUTING SYSTEMS

清华大学出版社

北 京

内 容 简 介

计算系统的形式语义是目前计算机科学理论研究的两大方向之一,其研究成果对程序设计语言、编译技术、应用软件、分布式系统等分支领域有重大的实际意义。本书大体上分为三个部分。第一部分是数学基础,为第一章。第二部分包括第二到第五章,概述了形式语义中的操作语义、指称语义、公理语义和代数语义四大经典流派。第三部分包括第六到第九章,概述了形式语义学的现代应用,分别介绍分布式系统、移动计算和移动通信系统、非规范进程代数和微观生命系统,以及量子程序设计语言的形式语义。

全书内容丰富,结构严谨,集形式语义学理论及其应用的有关分支之大成,系统地反映了这个领域各方面的研究成果,特别是它的近代发展潮流和趋势,并对不同流派的理论和方法给予了分析和评论。

本书可作为计算机专业研究生、本科生有关课程的教材或教学参考书,也可供有关专业或交叉学科的科研人员进行修或作为工具书。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121993

图书在版编目(CIP)数据

计算系统的形式语义/陆汝铃著. —北京:清华大学出版社,2017
ISBN 978-7-302-41494-0

I. ①计… II. ①陆… III. ①形式语义—研究 IV. ①TP301.2

中国版本图书馆 CIP 数据核字(2015)第 212873 号

责任编辑:薛 慧
封面设计:何凤霞
责任校对:王淑云
责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:北京雅昌艺术印刷有限公司

经 销:全国新华书店

开 本:170mm×230mm 印 张:118 插 页:10 字 数:2102 千字

版 次:2017 年 1 月第 1 版 印 次:2017 年 1 月第 1 次印刷

印 数:1~1500

定 价:398.00 元(全二册)

产品编号:012233-01

谨以本书纪念
吴允曾教授

前 言

本书是 1992 年出版的《计算机语言的形式语义》一书(简称 1992 版)的更新版。在 1992 版中我们分别介绍了形式语义的四个主要流派:操作语义、指称语义、公理语义和代数语义,并在此基础上综合阐述了并发和分布式程序的形式语义。从该书出版以来,已经二十余年过去了,在此期间,国内外形式语义学的研究有了许多重大的突破。

首先,应该提到的是移动通信的发展促进了一类新的进程代数—— π 演算的出现和发展,使得用进程代数编写的程序不仅可以在固定网络上运行,而且可以改变通信网络的拓扑结构,并在可变的网络结构上实行通信。围绕 π 演算出现了许多变种,开展了深入的研究。除了可变的网络结构之外, π 演算还创造了其他的手段来改进通信技术。例如,利用 π 演算,有可能建立专用的私密信道。随着 π 演算而兴起的环境演算 Ambient 甚至可以提供建立网上防火墙的手段。服务于保密通信的 Spi 演算把编码和解码引进了语言之中。带有噪音信道的含噪进程代数有希望把香农的信息论引进进程代数中,从而在两类通信理论之间架起一座桥梁。所有这些使得服务于移动通信的进程代数成了一门单独的学问。

其次,分子生物学是近年来发展最快的研究方向之一。人类基因图谱的测序完成大大推动了基因组学、蛋白质组学,以及系统生物学的研究。恩格斯曾经说过,数学在生物学中的应用为零,但是这样的时代早就过去了。以数学为基础的形式化方法已经深入到生物领域,成为描述微观生命现象的有力手段。特别值得称道的是概率论和随机过程作为重要的数学手段在面向微观生命现象的进程代数理论中发挥了重要作用。除了概率进程代数、随机进程代数和性能评估进程代数这些比较通用的工具之外,一大批特意针对微观生命现象的描述工具已经被设计出来并投入应用。诸如 Bioambient、因果 π 演算、化学抽象机、生化抽象机等工具提供了许多颇有启发的思路。描述微观生命现象的数学工具早已经不纯粹是微积分和微分方程等连续数学的天下,而成了连续工具和离散工具八仙过海的多极世界。

再次,实用化量子计算机出现和推广应用的前景正在吸引越来越多的科学家关注量子计算领域。近年来出现了一批面向量子计算的程序设计语言,其中也包括量子进程代数,科学家们已经开始讨论量子软件和量子软件工程。量子程序设计和经典程序设计有许多本质不同。例如,量子不可克隆原理的存在使得量子程序设计语言不可能有简单的赋值语句。同时,由于对一个量子系统的测量会引起该系统所处状态的坍缩,这使得系统状态的可观察性和程序运行流

程的可控制性受到影响。另一方面,量子力学的特殊性质并非只对量子程序设计起限制作用,它的某些性质有可能对量子程序设计是“有用”的。例如,量子离物传态、量子密集编码、量子密码通信等方面的研究成果有可能对量子网络和分布式量子软件工程产生较大影响,宜于我们去关注和开发利用。在一定意义上,量子程序的运行可以用经典程序来模拟。因此,尽管量子计算机还没有完全实现,但是量子程序设计和量子软件的研究并不一定要等到量子计算机实现以后再做,而是可以先行一步。而量子密码通信先于量子计算机的实现又提供了研究量子软件的必要性和现实性。更进一步说,量子软件的研究成果可以为未来量子计算机的设计提供新思路,比起先有经典计算机,后有经典程序设计的历史来说,这样做更合理。同样,在实际设计和使用量子程序设计语言之前先把它们的数学基础,即形式语义研究清楚,然后再去设计和使用,这比起传统的程序设计语言先设计、先使用,后论证其理论基础、后补其漏洞来说,未尝不是一件好事。

近十余年来形式语义学的研究成果极其丰富,本书把以上三个方面作为补写和更新的重点,是充分考虑到实践需求的一种选择。这三个方面都具有强烈的应用背景,同时又有深刻的理论问题,构成了三个新的章。同时本书也对原有的六章内容做了必要的补充。其中主要的四部分是:指称语义一章原书以(反映程序不确定性的)幂域理论结尾,本书添加了概率幂域理论和概率加不确定性幂域理论,同时还介绍了基于单体的计算理论和指称语义的完全抽象性研究。在公理语义一章添加了 Hoare 逻辑的概率推广和 Dijkstra 最弱前置条件语义(简称最弱前置语义)的概率推广。同时还介绍了基于一种实时模态逻辑的时段演算。并发和分布式语义一章原来重点介绍了 CSP 和 CCS 两个进程代数理论,本书补充介绍了另一个重要的进程代数 ACP,从而比较完备地展示了进程代数家族的三剑客架构。另外还分别给出了以扩充 Petri 网形式出现的 CCS 真并发语义和以抽象数据类型形式出现的 CCS 代数语义。这两种方法原则上可以推广到其他进程代数。这一章最后以 Glabbeek 的进程代数并发语义的比较研究结尾。最后,由于本书增添了以上各项内容,对第一章数学基础也要做必要的(最低限度的)增补。其中完整增加的有三节:一些基本的代数、拓扑和泛函知识,概率和随机过程知识,以及线性逻辑和 Gentzen 演算知识。另外对范畴论和格论两节做了必要的增补。

自然界有大量的信息交换并不采取人类语言或计算机语言的形式。一个明显的例子是生命系统。例如,DNA 序列就可以看成是一种文字,它构成了生命的“天书”。基因对蛋白质的表达,细胞间和细胞内部的通信,这里都有信息交换和信息处理在起作用。另一个明显的例子是量子系统。凡涉及量子计算、量子纠缠、量子通信、量子密钥等问题,无不归于信息表示和信息处理的范畴。尽管

现在的生命过程描述语言和量子程序设计语言都以计算机语言的形式出现,用于生命现象的模拟、预测和量子计算机的编程,我们仍然可以把微观生命系统和量子系统看成自然界的“天造”计算系统。俗话说人算不如天算,它们不仅应该和人造的计算系统平起平坐,而且还为人造计算系统提供了启示和展望。为此也需要形式化的数学工具,以严格和正确地刻画生命信息和量子信息的表示和处理。为此,我们统称本书的内容为计算系统的形式语义。

本书的内容通过每章的概述、除概述以外的各节,以及文献和中、英名词索引四部分组成,它们互不包含。概述是各章涉及内容的一个鸟瞰。章中各节是对其中某些内容有选择地展开讲解。文献是前两部分内容的出处和延伸阅读信息,其中早期文献可以提供有关研究的发展渊源信息,而近期文献则可以提供相关领域的发展前景和专家们的视角。由于形式语义的文献数量极多,肯定有一些重要的有关文献在写作本书时被遗漏掉。即使是已经列入本书介绍范围的一些文献,也肯定有一些因为作者理解不深,甚至理解有误而未能在本书中正确介绍其思想。我想提请读者注意的是,本书不能作为开展研究工作的直接依据。有兴趣在形式语义领域开展研究的读者,可以参照本书提供的线索,进一步阅读原著文献,可以在阅读时对照本书的解释、分析和某些进一步的发展,但是不能忽视阅读原著,以免出现可能的误导。

本书在撰写过程中得到了很多专家的帮助。从开始有写这本书的计划起,作者经常和应明生教授就此交流看法并得到他的很多宝贵建议。应明生、周巢尘、林惠民、夏培肃、陈仪香、袁崇义、林闯、李未、李克正、Bauer、Petri、Broy、Krieg-Brückner、Reisig 等教授都曾向作者提供他们的著作,使作者获得了宝贵的知识来源。为撰写本书,作者参考的文献数量较大,无法一一致谢。其中包括许多不署名的维基百科类资料,非常感谢这些未曾谋面的知识传播者。我们在书中尽可能地对引用的资料给出了出处,包括插图。

为了减少本书可能给读者带来的枯燥感,我们在每一章的前面加配了一首唐诗,每首诗的意境和它所在章的内容(之某些重要部分)存在一些本质的联想。我想读者会同意:在文化艺术和科学技术这两个差别巨大的领域之间不可能有真正的科学对应关系或推理关系,必须强调这仅仅是一种联想关系。它肯定不唯一,也肯定因人而异。学习和研究都是艰苦的劳动,但同时也应该是快乐的。作者赞成博拉·米卢蒂诺维奇的快乐足球理念,从事科学工作也要有快乐感。在此感谢马冬洁应邀为封面作图,并为每章的唐诗配图作画,使本书增色不少。

作者真诚地感谢清华大学出版社的大度宽容和坚定支持,使得这本撰写时间宽度达十年以上的书稿能够最终完成。

陆汝铃

2015 年春节前

《计算机语言的形式语义》前言

语义学(semantics)一词出自希腊文 *sēmantikos*,它由动词 *sēmainō* 转化而来,表示语言的意义(相当于英文的 *mean* 或 *signify*)。一般认为语言学的研究可分为三部分:语法学(syntax),研究语言的形态结构;语义学,研究语言 and 它所指的对象之间的关系;语用学(pragmatics),研究语言 and 它的使用者之间的关系(当然也要涉及语言所指的对象)。三者合在一起,组成了指号学(semiotics)的最重要部分。在计算机语言的范围内,语法学的研究已经相当成熟,语用学是一个基本上还没有被人们所认识的陌生世界,只有语义学是一个正在蓬勃发展中的领域。在这个领域中既有已经取得的丰硕成果,又有尚待解决的重大问题,它是计算机科学中勇敢的探索者的乐园。

20 世纪 50 年代是计算机语言兴起的年代。在这一阶段的早期,计算机语言的设计往往主要强调其“方便”的一面,而比较忽略其“严格”的一面,因而对语言的语义,甚至是语法,未下严格的定义,从而语言设计者、语言实现者和语言使用者对同一语言的语义缺乏共同的理解,造成了一定程度的混乱。Chomsky 关于语言分层的理论,以及 Backus 和 Naur 关于上下文无关文法表示形式的研究成果推动了语法形式化的研究。其结果是,在 ALGOL 60 的文本设计中第一次使用了 Backus-Naur 标准型表示语法,并且第一次在语言文本中明确地把语法和语义区分开来。后来,在 20 世纪 50 年代和 60 年代间,面向语法的编译自动化理论研究得到了很大发展,使语法形式化研究的成果达到实用化的地步。

语法形式化问题基本解决以后,人们逐渐把注意力集中到语义形式化方面。20 世纪 60 年代可以说是计算机语言形式语义学正式诞生的 10 年,形式语义学的四大流派皆渊源于这一时期。其中,1964 年被认为是操作语义学和指称语义学的诞生年代,Landin 关于操作语义的奠基性文章《表达式的机械化处理》和 Strachey 关于指称语义的奠基性文章《关于形式语义学》都问世于这一年。1967 年则被认为是公理语义学诞生的年代,Hoare 关于公理语义的奠基性文章《计算机程序设计的一个公理学基础》就发表于这一年。最年轻的是代数语义学,它是在抽象数据类型理论的基础上发展起来的。虽然后者的思想源自 1967 年问世的 SIMULA 67,但正式把它提到抽象数据类型高度的是 Liskov 和 Zilles 在 1974 年的工作,而把它进一步上升为代数语义学则是 20 世纪 70 年代中期以后的事了,当时的主要代表人物有 Goguen 等人的 ADJ 小组及 Guttag 等人。

操作语义的基本思想是用抽象的方法描述语言中每一成分的执行效果,以免所描述的语义依赖于该语言实现时所用的具体计算机。通常的做法是设计一

个抽象机,定义一组抽象状态,把语言的语法表示成抽象的形式,然后指明抽象机每加工一个语言成分时将对状态作何种改变。这种语义方法与语言实现的关系比较紧密,但是难以用数学方法处理,而且对语义描述者个人使用的实现方法依赖很大。

指称语义的基本思想是使语言的每一成分对应于一个数学对象,该对象称为该语言成分的指象,它不像操作语义那样涉及语言成分的执行过程,而是只考虑各成分执行的最终效果,并认为此最终效果应不依赖于其执行过程。由于语言成分可以是很复杂的,因此它的指象也可以很复杂,甚至达到令人怀疑这样的指象在数学上是否存在的程度。Scott 创建的论域理论解决了这个问题,从而为指称语义学奠定了坚实的理论基础。后来 Smyth 和 Plotkin 等人又建立了幂域理论,为不确定、并发和分布式程序的指称语义奠定了基础。

公理语义是在程序正确性验证的基础上发展起来的,它不像其他语义学方法那样,对程序语义作宏观的全局性描述,而只是给出一种方法,使人们能在给定的前提下,验证某种特定的性质是否成立,例如,

$$\{P[e/x]\}x := e\{P\}$$

表示:若在执行赋值语句 $x := e$ 前条件 $P[e/x]$ 成立,且执行能够终止,则在执行后条件 P 成立。因此,公理方法的基础是一个逻辑系统,包括一组公理及其推理规则,它区别于经典逻辑的主要之点是把程序执行的效果也考虑进逻辑系统中。公理语义学的核心课题是研究这类逻辑系统的健康性和完备性。近年来,一些所谓非 Hoare 型的公理语义方法陆续出现,如模态逻辑、时序逻辑、动态逻辑、无穷逻辑、构造性逻辑等,大大丰富了这一领域的研究。

代数语义学的基本思想是把描述语义的逻辑体系和满足这个逻辑体系的模型区分开来。任何程序的操作语义或指称语义描述只给出该程序的一个语义模型,而公理语义描述则只给出逻辑系统,不深入探讨其可能的模型(近年来这种情况已有所改变,特别是对非 Hoare 型逻辑,参见本书 4.14 节)。代数语义学的特点就是用代数方法来处理满足一个逻辑系统的各种模型,把模型的集合看成是一个代数结构。因此,在代数语义中,除了要研究类似于健康性和完备性等公理体系的概念外,还要研究模型之间的关系。简而言之,它要处理的是一整个模型族。

形式语义学文献中使用的技术和方法并不一定都能归入上面所说的四大流派,但这四大流派确实代表了形式语义学中最主要的四个研究方向,它们在本书中各自所占的篇幅大体上反映了该领域当前研究的活跃程度。相对地说,在公理语义和代数语义中,尚待研究和解决的问题比起操作语义和指称语义来要多一些。

当前研究兴趣的集中点之一是各种非 Hoare 型逻辑体系,在公理语义和代数语义中都有这种情况,至于哪一种逻辑最有前途,目前尚无定论。有人看好时序逻辑,有人欣赏无穷逻辑,近年来类型理论备受青睐,似乎又预示着研究构造性逻辑的热潮即将兴起。总之,这里还是各种逻辑群雄逐鹿(也许更合适的词是百花齐放)的局面。

有分即有合。也有人研究如何构造一个超级逻辑体系,把各种具体的逻辑系统都容纳进去,例如 Edinburgh 学派的机关(institution)理论。目前,这类研究主要是纯理论性的。

当前研究兴趣的另一个集中点是不确定、并发和分布式程序的形式语义,我们为后两者单设了第 6 章,一般认为,顺序式语言和顺序式程序的形式语义研究比较成熟,而上面提到的三个方面的形式语义研究则还有很长的一段路要走。例如,操作语义的偏序推导和变迁系统,指称语义的幂域理论和不动点理论,都是研究分布式语义的有力手段,但都还不完善。分布式程序的真并发语义,以及它的一些重要性质,如公平性和发散性,是近年来有关文献中的热门话题。如何用代数语义方法和更多的代数手段来处理分布式语义,也是许多研究者特别关心的课题。

操作语义和指称语义并没有失去其生命力。有两个情况使操作语义的研究保持着一定的势头。其一是 Plotkin 在 1981 年发明的结构化操作语义,给人以操作语义重新焕发青春之感;以结构化操作语义为基础的变迁系统已普遍用于各类语言,包括分布式语言的语义描述中。其二是在论述其他语义的正确性时,操作语义至今仍常被作为“参照系”而使用。至于说到指称语义,则在不少计算机科学家的心目中,它仍然是一种“标准”的语义描述方法。

四种语义方法融合的倾向值得注意。它的一种表现是各方法之间的边界逐渐模糊,互相采用对方的长处,有些语义方法已很难绝对地划入某一阵营。另一种表现是有人试图建立一种超级方法体系,把四种现有方法都纳入其中作为子方法,在程序的不同开发阶段根据各方法的特点量才录用。

由于本书涉及的内容比较多,为了方便读者,首先在第 1 章给出了必要的数学基础,在第 2—6 章的每一章前给出了一节概述。概述的目的是,没有时间细读整章内容的读者可以只读概述部分;细读整章内容的读者可以用它把各节内容串成一个整体。概述中还包括了一些不能独立成节的内容。

作者衷心感谢唐稚松教授、林惠民副教授和冯玉琳教授在本书审稿方面给予的宝贵支持。林惠民审阅了第 1 章和第 5 章,冯玉琳审阅了第 2、3、4、6 各章。他们在百忙中花费了大量时间阅读原稿,指出了原稿中存在的许多不当之处,使本书内容得以进一步完善。作者还十分感谢本书的第一位读者韦梓楚同志,他

细读了本书初稿,发现了不少问题,提出了许多宝贵的建议。

作者十分怀念原北京大学教授吴允曾先生。本书是在他的建议和鼓励下开始写作的。吴先生离开我们已有多年,但他的音容笑貌和谆谆教诲却仍浮现在眼前。作者也十分感谢原《计算机科学丛书》编委会的全体成员,特别是王湘浩教授和徐家福教授,感谢他们对本书出版所给予的热情支持。我还特别感谢科学出版社第六编辑室编辑同志们的耐心和宽宏。本书书稿的完成时间大大超过了原合同的规定,于今仍能顺利出版,这样的支持是难能可贵的。当然,时间上的延长也使我有机会接触更多和更新的研究成果,然而,其结果之一是本书的篇幅也大大膨胀了。

本书的文献目录是冯方方、曹存根和孙荣平协助编辑和录入的,张松懋帮助整理了全部书稿。我在此向他们一并致以衷心的感谢。

形式语义学是一门发展中的学科,涉及面广,有较强的理论性。尽管作者已几易其稿,但书中的错误和不妥之处仍在所难免,欢迎读者批评指正。

陆汝铃

1992 年春节前



射人先射马，擒贼先擒王

——杜甫

目 录

第 1 章 数学基础	1
1.1 λ 演算	1
1.2 格论	14
1.3 范畴论	32
1.4 不动点理论	56
1.5 Petri 网论	64
1.6 Hilbert 空间和相关拓扑、代数结构	82
1.7 概率和随机过程	95
1.8 矢列演算、线性逻辑、线性类型系统和线性带类型 λ 演算	107
1.8.1 从矢列演算讲起	107
1.8.2 线性逻辑	111
1.8.3 线性类型系统	118
第 2 章 操作语义	125
2.1 概述	125
2.2 SECD 抽象机	133
2.3 维也纳定义语言	142
2.4 赫斯利方法和 PL/ I 标准	161
2.5 W 文法及其抽象机	174
2.6 变换语义学	184
2.7 结构化的操作语义	199
第 3 章 指称语义	207
3.1 概述	207
3.2 指称语义的描述方法	225
3.3 函数式语言的指称语义	229
3.4 命令式语言：直接语义和继续语义	234
3.5 变量、说明和作用域	243
3.6 过程和函数	252
3.7 元语言 META IV	264
3.8 域的递归理论	278

3.9	递归域的两个模型	287
3.10	幂域理论	303
3.11	不确定程序的指称语义	316
3.12	概率幂域和概率指称语义	322
3.13	基于概率不确定幂域的指称语义	332
3.14	计算理论的范畴论语义	344
第 4 章	公理语义	351
4.1	概述	351
4.2	Hoare 公理系统	361
4.3	分程序的公理语义	373
4.4	过程的公理语义	382
4.5	联立子程序的公理语义	395
4.6	类程的公理语义	412
4.7	Pascal 的公理语义	420
4.8	完备性和可表达性	431
4.9	过程公理的健康性和完备性	440
4.10	完全正确性	451
4.11	最弱前置条件和不确定性公理语义	461
4.12	最弱概率前置语义	472
4.12.1	概率程序的最弱前置语义	472
4.12.2	概率不确定程序的最弱前置语义	479
4.13	类型理论和程序逻辑	483
4.14	模态逻辑和时序逻辑	498
4.15	分支时序逻辑和线性时序逻辑	505
4.16	区间逻辑和时段演算	518
4.16.1	区间逻辑 IL	518
4.16.2	时段演算 DC	522
4.16.3	一个实例	527
4.17	动态逻辑	531
第 5 章	代数语义	539
5.1	概述	539
5.2	Σ 代数和初始语义	545

5.3	扩充的公理形式	556
5.4	健康性、完备性和可判定性	565
5.5	充分完备性和层次一致性	577
5.6	理论描述语言 Clear	583
5.7	代数语义的范畴论基础	591
5.8	终结语义	605
5.9	格语义	613
5.10	可观察性和观察等价性	620
5.11	偏 Σ 代数	635
5.12	模型描述语言 ASL	648
5.13	程序设计语言的代数语义	657
5.14	带动态结构的程序的语义	667
 第 6 章 并发和分布式程序的形式语义		679
6.1	概述	679
6.2	分布式程序设计语言 CSP	707
6.3	CSP 的结构化操作语义	716
6.4	CSP 的流语义	727
6.5	TCSP 和失败语义	734
6.6	并行程序的公理语义	743
6.7	CSP 的公理语义	751
6.8	通信系统演算(CCS)	768
6.9	CCS 的操作语义	772
6.10	同步树和通信树	779
6.11	双模拟和行为等价性	786
6.12	SCCS 和集合推导语义	797
6.13	CCS 的偏序推导语义	802
6.14	CCS 的 Petri 网语义	814
6.15	分布式变迁系统和 CCS	820
6.16	CCS 的真并发语义	830
6.17	Hennessy-Milner 逻辑	848
6.17.1	基本 HM 逻辑	848
6.17.2	带递归 HM 逻辑	853
6.18	通信进程代数 ACP 家族及其静态语义	859

6.18.1	基本进程代数 BPA	859
6.18.2	进程代数 PA	860
6.18.3	通信进程代数 ACP	862
6.18.4	ACP 的扩充	864
6.18.5	ACP 的最大扩充 ACP_e	870
6.19	动态 ACP 及其操作语义	873
6.20	ACP 的指称语义和双模拟语义	883
6.21	抽象数据类型作为进程代数的代数语义	892
6.22	进程代数并发语义的比较研究	913
第 7 章	移动通信和移动计算系统的形式语义	927
7.1	概述	927
7.2	π 演算及其操作语义	952
7.3	π 演算的双模拟语义	975
7.4	进程代数的符号变迁语义	986
7.4.1	CCS 型的进程代数的符号语义	986
7.4.2	π 演算的(强)符号语义	995
7.4.3	π 演算的(弱)符号语义	999
7.5	多维 π 演算和异步 π 演算	1001
7.5.1	多维 π 演算	1001
7.5.2	异步 π 演算	1010
7.6	安全 π 演算 SPI	1020
7.7	SPI 演算的环境敏感双模拟语义	1039
7.8	Applied π 演算	1059
7.9	Applied π 演算的符号语义	1072
7.9.1	Delaune, Kremer 和 Ryan 的 DApplied π 演算及其符号语义	1072
7.9.2	Dolev-Yao 模型、可达性模型和约束系统	1082
7.9.3	刘佳和林惠民的符号 LApplied π 演算语义	1085
7.10	对称 π 演算: χ 演算和 Fusion 演算	1092
7.10.1	χ 演算	1092
7.10.2	Fusion 演算	1098
7.11	移动 Ambient 演算	1106
7.11.1	基本 Ambient 演算——移动 Ambient 演算	1107

7.11.2	完整 Ambient 演算——通信 Ambient 演算	1116
7.12	Ambeint 演算的类型系统	1119
7.13	分布式 Ambient 演算的双模拟语义	1133
7.14	安全 Ambient 演算及其双模拟语义	1142
7.14.1	安全 Ambient 演算 SA	1142
7.14.2	带口令的安全 Ambient 演算 SAP	1146
7.15	封装 Ambient 演算	1155
7.15.1	封装 Ambient 演算 BA	1155
7.15.2	新封装 Ambient 演算 NBA	1161
7.15.3	密封 Ambient 演算 SBA	1165
第 8 章	非规范进程代数和微观生命系统的形式语义	1169
8.1	概述	1169
8.2	从强化操作语义到因果 π 演算	1199
8.3	概率进程代数	1210
8.3.1	部分概率进程代数 PCCS	1211
8.3.2	全概率进程代数 APPA	1226
8.4	性能评估进程代数 PEPA	1233
8.5	随机 π 演算	1252
8.6	含噪 π 演算	1261
8.7	进程演算的拓扑理论	1275
8.8	进程序列演算 CPS	1296
8.8.1	CPS 的语法和操作语义	1296
8.8.2	CPS 的序列双模拟语义	1299
8.8.3	CPS 的特征序列双模拟语义	1309
8.9	CPS 的序列极限双模拟	1315
8.9.1	动程的贴近双模拟语义	1315
8.9.2	CPS 的极限序列双模拟语义	1318
8.10	Gillespie 算法	1331
8.11	π 通路演算——分子水平的生物进程代数	1336
8.11.1	关于通路	1336
8.11.2	π 通路演算编程信号传导通路	1338
8.11.3	随机 π 通路演算编程基因调控通路	1345
8.12	κ 演算——基于规则的蛋白质相互作用语言	1352