λ-演算的语法和语义

——计算机科学基础理论之一

［荷兰］ H. P. 巴伦德莱赫特　著
朱一清　译

南京大学出版社

1992 · 南京

λ-演算的语法和语义

[荷兰]　H.P. 巴伦德莱赫特　著

朱一清　译

南 京 大 学 出 版 社

1992·南京

(苏)新登字 011 号

内 容 简 介

本书介绍了不带类型的 $\lambda-$ 演算中迄今为止的几乎全部内容。分概念、换位、归约、理论和模型五大部份逐一介绍。其中包括 $\lambda-$ 项，$\lambda-$ 可定义，$\lambda-$ 换位，$\beta-$ 归约，$\eta-$ 归约，组合理论，Böhm 树，归约对策，$\lambda-$ 模型等概念，分述于二十一章，并附有大量习题。

本书可作为计算机专业、数学专业大学生和研究生的教材，作为从事计算机、数学、逻辑学和语言学教学及研究工作者的参考书。

λ-演算的语法和语义

[荷兰] H.P. 巴伦德莱赫特 著

朱一清 译

屠念祖 校

责任编辑 顾其兵

*

南京大学出版社出版发行

(南京大学校内，邮政编码 210093)

南京航空航天大学印刷厂印刷

*

开本 787×1092 1/16 印张 15 字数 710 千

1992 年 9 月第 1 版 1992 年 9 月第一次印刷

印数 1—1000

ISBN 7-305-02050-8/〇·107

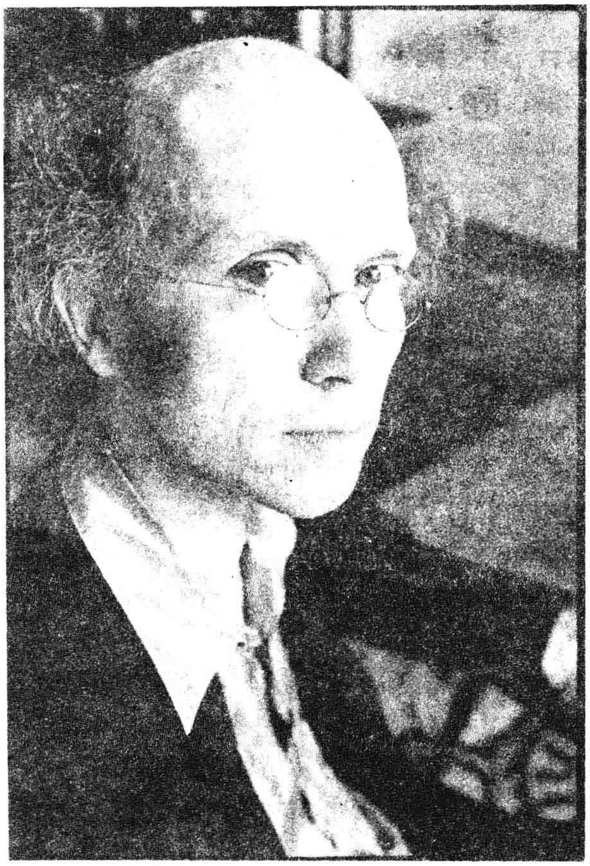
平装定价 24.00 元

精装定价 36.00 元

音乐像神秘的数学，它变幻无穷、意境深远。

Claude Debussy [1903]

本书谨献两位女士:
Dolly Barendregt-Kessing
和
Maria Montessori.
第一位是我的母亲，她使我善于思考；第二位，尽管她不是教师，但她教导我并引导我去体验真理。



Henk Barendregt

H.P. Barendregt

Photo by Flip Franssen, Nijmegen

译者序言

本书是著名丛书"STUDIES IN LOGIC AND THE FOUNDATIONS OF MATHEMATICS"中的一册。1981 年它首次出版就受到欧洲、美洲等地学者的广泛赞赏，至今已印刷三次。世界著名的期刊杂志"*Mathematical Reviews*"、"*Journal of Symbolic logic*"和"*Zentralblatt für Mathematic*"均撰文赞扬此书。他们说:"作者以非常生动和完善的证明将众多内容如此成功地进行处理，使人惊叹!""……它包括所有不带类型的 λ-演算的内容。""它包括了当今该课题的研究成果，必将推动今后的研究。"

λ-演算作为数学基础理论的研究，在 20 世纪 30 年代吸引了众多的数学家。当时人们提出怎样判断一类数学问题是机械可解的，亦即怎样判断一些函数是可计算的。随即而来的问题是如何定义"函数是可计算的"这个概念。1936 年，*Church* 采用 λ-可定义作为可计算性的定义,这个定义沿用至今。

λ-演算的语法极其简单，但它的功能却相当强，足以描述所有计算机可计算的函数。最近计算机事业的发展又引起人们对 λ-演算的极大兴趣，人们发现 λ-演算可看成范例型的程序设计语言，它的语义适用于 *ALGOL* 语言、*Lisp* 语言、函数式程序设计语言等。这样，这门理论受到当今计算机科学家、数学家、逻辑学家和语言学家的重视。从某种意义上讲、λ-演算是计算机科学和数学、逻辑学的交接点。

H.P. Barendregt 教授是著名荷兰数学家，他非常关心我国学者在这个领域的研究，他热心地帮助我将他的著作介绍给我国读者。

限于译者水平，译文中一定有不少错误，敬请读者批评指正。

感谢南京大学出版社的荣翠琴先生、中科院地理所的徐爱珍女士和叶斌先生在出版和印刷中所给予的真诚帮助。

朱一清 于南京东南大学计算机系
1991 年 10 月

著者为中文版写的序言

在中文版“λ-演算的语法和语义”一书出版之际，我衷心感谢朱一请先生为出色的翻译而作的努力。

1976年到1980年期间，我正着手写这本书，那时我没有能预见到人们对该理论的兴趣会增长得如此之快。

λ-演算的大部份篇幅与计算机科学相联系，并得到了应用。

我非常高兴中国读者是那么容易地就接受了λ-演算。

H.P. 巴伦德莱赫特

1991年11月12日

原　序

20 世纪 30 年代，人们引入不带类型的 λ-演算作为逻辑学和数学的基础。但由于悖论的出现，这个目标未能实现。然而，将 λ-演算中相容的部分作为计算的理论却是相当成功的。它推动了早期递归论的发展，而近来更促进了计算机科学的研究。尽管有了悖论，λ-演算仍有可能作为一种可供选择的基础。这种观点最近重新引起人们的注意。

由于这些学科的发展，λ-演算对它本身来说也已变得值得加以研究。本书主要就"纯"λ-演算而写。同时，关心应用的读者也会觉得本书相当有用，因为它对应用是很有启发性的。λ-演算在构造上是很具直觉性的；例如，程序设计语言的语义。因此，本书是专为逻辑学家、数学家、计算机科学家和哲学家而写的。

在这本书的诞生过程中，我要感谢 Anne Troelstra，她是"*STUDIES INLOGIC*"的编辑，是她建议我写此书。感谢 NORTH HOLLAND 出版公司的 Einar Fredriksson，在过了交稿期后还耐心地等着。

在 1974 年—1975 年间，我与 Jan Bergstra, Jan Willem Klop 和 Henri Volken 进行了令人鼓舞的学术讨论。Jan Willem Klop 总是乐意跟我讨论 λ-演算，本书未能登出他在该论题上的所有观点，详细情况可读 Klop [1980]。Gordon Plotkin 和 Dana Scott 与我亦有过好几次极其有益的讨论。Mariangiola Dezani, Roger Hindley, Karst Koymans, Jean-Jacques Lévy, Giuseppe Longo, Gerd Mitschke, Albert Visser 和 Chris Wadsworth 读过我部份手稿并提出了批评建议。

特别感谢 Adrian Rézus 为我准备索引和参考资料，感谢 Jan Willem Klop 为我绘图。

本书主要部份在 Rijksuniversiteit Utrecht 写成，感谢 Dirk Van Dalen 给我提供出色的工作条件。还谢谢荷兰的 Z. W. O.机构、E. T. H. Zürich 的 Mathematisches Forschungsinstitut 的支持。最后在 Mathematisches Forschungsinstitut Oberwolfach 我完成了手稿。

手稿的 90%由 Do Breughel-Vollgraff 打印，10%由 Ans, Ellwyn, Joke, Marianne, Renske, Sincke 和 Sophie 打字，一并在此表示由衷的感谢。

Henk Barendregt

1980 年 2 月

目　　录

第二部份　换　　位

第三部份　归　　约

第四部份 理 论

第五部份　模　　型

第0章 读者须知

读者知识准备

本书基本上是自足的，尚须一阶逻辑、拓扑学、集合论、递归论和范畴论的基本知识。读者如果需要，可参考下列书籍:

First order logic: Enderton [1972], Barwise [1977a].

General topology: Kelly [1955].

Set theory: Halmos [1960], van Dalen 等 [1978].

Recursion theory: Rogers [1967].

Category theory: Arbib and Manes [1975], Maclane [1972].

约定记号

下述符号在本书中使用:

$\mathbb{N}$: 自然数的集合。

$P(\mathbb{N})$ 或 $P\omega$: $\mathbb{N}$的子集的集合。

$\lambda x.\cdots$: (元 λ 抽象) 表示对所有 x 而言，集论函数$f(x)=\cdots$。(例如: $(\lambda x.x^2+1)(3)=10$)。

$\mu x\cdots$: 使得…成立的最小的$x\in\mathbb{N}$。

Seq: 编过码的序数的集合，例如 $Seq=\{<n_1,\ldots,n_k>\in\mathbb{N}|k\in\mathbb{N},n_1,\ldots,n_k\in\mathbb{N}\}$。如果 $\alpha=<n_1,\ldots,n_k>\in Sep$，则 $lh(\alpha)=k$。我们约定 $<\ \ >\in Seq$ 且 $lh(<\ \ >)=0$。如果 $\alpha=<n_1,\ldots,n_p>,\beta=<m_1,\ldots,m_q>$，则 $\alpha*\beta=<n_1,\ldots,n_p,m_1,\ldots,m_q>$；$\alpha\leqslant\beta$ 当且仅当 $p\leqslant q$ 且 $n_i=m_i(1\leqslant i\leqslant p)$; $\alpha<\beta$ 当且仅当 $\alpha\leqslant\beta$ 且非 $\beta\leqslant\alpha$。

逻辑连接词: 用$\neg$表示“非”; $\vee$表示“或”; $\wedge$表示“与”; $\Rightarrow$表示“蕴含”; $\Longleftrightarrow$表示“当且仅当”亦即“iff”; $\forall$ 表示全称量词; $\exists$ 表示存在量词; $\exists!x$ 表示存在唯一的 x。

范畴论的图也常用来表示命题。例，下图中R表示集合X上的二元关系，

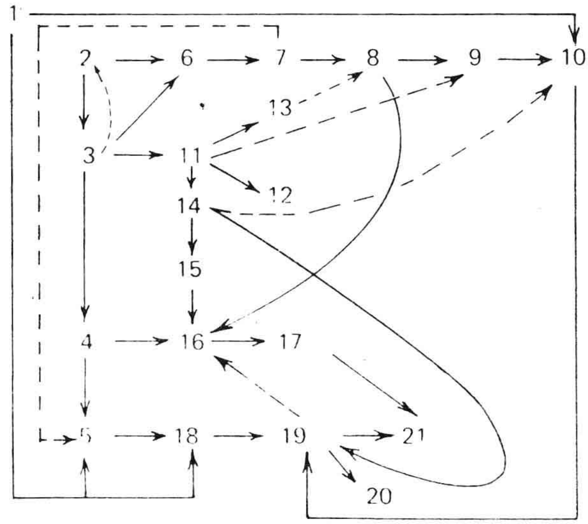
$$\begin{tikzcd} & x \arrow[dl, dash, "R"'] \arrow[dr, dash, "R"] & \\ x_1 \arrow[dr, dashed, dash, "R"'] & & x_2 \arrow[dl, dashed, dash, "R"] \\ & x_3 & \end{tikzcd}$$

则图表示$\forall x,x_1,x_2\in X[xRx_1\wedge xRx_2\Rightarrow\exists x_3\in X[x_1Rx_3\wedge x_2Rx_3]]$。

材料安排

纯 λ–演算的主要论题是: 换位，归约，理论和模型。这些论题将在本书的主要部分中系统讨论，并且在第 i 章 ($2\leqslant i\leqslant 5$) 的最后一节给出第 i 部分的介绍。第一章是一般性的介绍，亦给出了我们必须先具备的而又较少熟悉的完备偏序理论。

下面的表大致地指出各章的关系。虚线箭号表示一些较后章节的定理在较前章节中不加证明而引用情形。



(我希望读者按此图能顺利地阅读本书。)

每章均附以习题，较难者用（*）标记。

本书着重理论体系而不是理论的表述，因此让我来帮你指出如何组织（或读）这些材料。组织一个简短的课程，可选§2.1，3.1，3.2，第六章，§1.2，5.1，5.2，5.3 和 5.4。我们可以从中领略语言的递归能力，用模型论和语法的方法都证明了系统的相容性。在这个基础上，我们可以先将§7.1，7.2，8.1，8.2，8.3，10.1，11.1，11.2，13.1，13.2，18.1 和 18.2 加进去。它们讨论：1.组合理论；2.可解性、Böhm 树和特殊的归约；3.模型$P\omega$, D_∞。

然后，我们可以加进§10.2，10.3，14.1，14.3，16.1，16.2，19.1 和 19.2，在模型内部给出了对项值的分析。

对另外一些课题的兴趣就取决于你——读者的爱好了。你可以发现第 2，3，4，5 章的最后一节会给你描绘出本书其余部分的概貌，而我最喜欢的是下述章节：§13.4，15.3，17.1，20.4，还有§9.4，10.4，17.2 和第 21 章，尽管后者更具技术性。

第一部分　基本概念

第 1 章　介　　绍

1.1　λ-演算的概况

λ-演算是将函数看作规则而不看作图像的一种不带类型的理论。"函数视作规则"是一种老式的函数概念，它将函数看作是按某种定义从自变量到某值的编码过程。Dirichlet 认为亦可将函数看作自由变量和函数值组成的序偶的集合，即看作图。这在数学上相当重要。λ-演算重新提起老概念旨在强调它的计算性质。

将函数视作规则是从一般意义上考虑的。例如，我们可以将函数用普通英语来定义，它对自变量的作用亦用英语表达。更明确地说，将函数考虑为：对某机器而言，函数是由程序所给出的，由机器再作用于另外一些这样的程序。上述两种情况，我们均有一个不带类型的结构，结构中同时研究的对象是函数和自变量。这就是不带类型的 λ-演算的出发点。在特别情况下，一个函数可以作用于自身。这对数学的一般概念而言（例 Zermelo-Fraenkel 的集合论所述），是不可能的（由于基本公理）。(参考练习 18.5.30)

λ-演算的三个方面如下：

Ⅰ、数学的基础。

Ⅱ、计算。

Ⅲ、纯 λ-演算。

λ-演算的另一些基础概念发表在 Scott [1975b], [1980] 和 [1980a] 的文章中。

Ⅰ、数学的基础和 λ-演算

λ-演算和与此相关的组合逻辑的奠基者们具有两个目的：

(1) 发展函数的一般理论（例如，处理公式）。

(2) 用逻辑概念扩展上述理论并为逻辑学和数学的一部分提供基础。

组合逻辑的奠基人 Schönfinkel 和 Curry 明确重视第一个目的，这一工作同时也包含于 Church 在 λ-演算的奠基工作中。

不幸的是，为数学奠基的所有努力都失败了。Kleene 和 Rosser [1935] 指出 Church

的最初系统 [1932／1933] 是不相容的。Aczel [1980] 亦指出 Frege 的著名不相容性理论 [1893，1903] 基本上包含了 λ–演算，这事实再一次表明工作的失败。

Curry [1930] 提出了一个相容理论，在纯组合逻辑方面达到了上述第一个目的，然而他的理论在逻辑方面太弱以致不能作为基础。

自从 Kleene–Rosser 悖论出现之后，Church 的基本计划受挫。Church [1941] 给出了一个他的原始系统的相容子理论（表达成 Church–Rosser 定理）。它仅涉及函数部分，这个理论就是 λI–演算。

另一方面，Curry 也不想回避悖论。他打算用演译概念来扩充纯组合逻辑，去实现第二个目的，同时也给出一个对悖论的分析。（参考 Curry 等 [1958]，[1972]）。尽管 Curry 的计划没有完全实现，但也已取得一些成果。（我们可以参考 Fitch [1974]，Scott [1975a]，Seldin [1976／77] 和 Aczel [1980]）。

λ–演算还提供其他一些基本的方向。Feferman [1975],[1980]将偏函数作为构造数学的基础发展了一个系统。偏函数的基础集合与 Wangner [1969] 和 Strong [1968] 的一致自反结构相关。（参考 Barendregt [1975]）。这些 URS（指一致自反结构）与定义了作用运算的 λ–演算的关系是一个未解决的问题。

除了这些，λ–演算还与证明论，范畴论有关。在这些领域中，λ–演算是带类型的。

Ⅱ、计算和 λ–演算

递归论

λ–演算中涉及整数的那部分是完全成功的。利用这个理论 Church 用"λ–可定义"的概念作为"能行性计算"的形式定义。Kleene [1936] 证明 λ–可定义等价于 Gödel–Herbran 的递归性。同时，Church 将他的理论公式化（曰递归性是能行性计算的固有形式体系）。Turing [1936]，[1937]给出了机器可计算性的分析并证明了所得的概念（Turing 可计算）等价于 λ–可定义性。

发现悖论以后，Kleene 在 λ–可定义概念上转化了一些结论并得到几个基本的递归理论定理（见 Crossley [1975]，p.4–7）。部分递归函数的可枚举定理，S–m–n 定理和递归定理的发现皆因为 λ–演算理论的启发。

计算机科学

最近几十年，计算机事业的兴盛导致程序设计语言的大大发展。λ–演算具程序设计语言及其执行过程的好些特征。例如，λ–演算中的有界变量对应于过程的形式参数；不带类型相应于机器把程序和数据视为同样一串数位序列。

根据 Turing 的分析研究，尽管 λ–演算的语法是如此简单，λ–演算的力量却是强大到足以描述所有计算机可计算的函数。从而 λ–演算可以看成是范例型的程序设计语言。这当然不是说我们必须用它来写实际的程序，而是说 λ–演算处理程序设计若干问题，特别象过程名这类问题纯粹是从形式上处理的。这些研究使我们在程序设计语言的设计和分析上取得成果。

例如，好几种程序设计语言从 λ–演算获得启发（可能是无意识的）。在 ALGOL'60，'68 或 Pascal 中，过程可以是过程的变量。在 Lisp 中情况也是如此，而且一个过程也