LISP

符号计算简明教程

D.S.图尔兹基

姜 新 译

黄玉霞 校

北京科海总公司培训中心

一九八八年三月

译 者 的 话

本书原著名为“ LISP — A Gentle Introduction to Symbolic Computation”是1984年出版的。作者 DAVID S. TOURETZKY 是美国卡内基－梅龙大学计算机系的教授，长期从事 Lisp 语言的教学工作。

在学习 Lisp语言的过程中，本人仔细地阅读了原著，觉得该书深入浅出，通俗易读，引人入胜，便动笔把它译出来，供对人工智能感兴趣的读者作为学习 Lisp语言的入门教材，或者作为教授 Lisp语言程序设计课的教课书。作者写该书的目的及本书的阅读对象请见前言部分。

本书原著出版后，在美国引起了强烈的反响，有关情况请见“对本书的评价”。

本书在翻译过程中，得到了龚理嘉和张秀英的热情帮助，并由科学院计算中心付研究员黄玉霞认真校阅，在此表示衷心的感谢。

由于译者水平所限，时间仓促，缺点错误难免，殷切希望读者提出批评指正。

1987．8

前 言

这是一本学习用Lisp语言编写程序的书。Lisp是最老的计算机语言之一，大约有25年的历史了，然而它从没有象今天这样重要和受人们欢迎。Lisp非常重要的一个原因是由于它是人工智能研究或“AI”的公共语言。AI正在从实验室走向日常生活领域。这将使我们的社会发生巨大的变化。Lisp今后还会更受欢迎，因为它提供了一种友好的交互式程序设计环境，对初学编程序的人给以极大地帮助。

当我写这本书的时候，心目中有三种类型的读者，我愿依次予以说明：

• 对于第一次上程序设计课的文科大学生，我想强调一下书名中的“简明”一词。我设想你们除了算术以外没有其它数学的基础。即使你不喜欢数学，这本书仍可以教你喜欢计算机程序设计。我避免使用专门的术语。书中有很多例子，还有大量的练习，其答案可以在附录D中找到。

• 对于心理学家、语言学家以及其他对AI和认知科学感兴趣的人们，当你开始进入人工智能的研究时，你会发现在这个领域中几乎所有的研究都是用Lisp实现的。这本书既可以成为深入理解技术文献的入门教材，又可以成为人工智能核心工具的简捷介绍。

• 对于计算机业余爱好者。当今一些个人计算机，例如Apple和许多Z—80的机器上都配有不同的高质量的小型Lisp系统。这些系统对于学习Lisp和编写简单的程序是足够的，但是目前由于内存不足，它们的功能受到了限制。如果使用功能更强的个人计算机，例如这些机器具有24位或32位的地址空间和虚拟存贮能力，业余爱好者就能够运行在今天AI实验室里使用的Lisp软件。把具有研究特性的Lisp系统置于计算机爱好者手中的远景确实是令人鼓午的。本书将教你使用现今提供的Lisp系统，但是更重要的是它为你接受即将到来的新技术做好了准备。

从未有过标准的Lisp。在整个Lisp历史进程中该语言不断地发展，并逐步地成熟起来；这个进程还未完结，因此存在着大量的Lisp语支。Lisp 1.5是最接近所谓标准的Lisp，但是它是1962年的文本，缺少现代Lisp的许多功能。当今主要的语支是Mac Lisp，Inter Lisp，UCI Lisp和Lisp Machine Lisp；Franz Lisp和Common Lisp都是Mac Lisp的分支，它们正迅速地发展并变得愈来愈重要。本书主要基于Mac Lisp和Common Lisp，少量借用其它语支。附录C说明了如何使任何一种Lisp系统与本书使用的Lisp实际上兼容。

对本书的评价

这是一本极好的介绍性教材……，它是我见过的向非科学工作者介绍 Lisp 书中最好的一本。

—Alan J. Perlis，耶鲁大学

这是有史以来介绍 Lisp语言最好的一本书。

— Daniel L. Weinreb，符号公司

一本十分生动的，通俗易读的介绍 Lisp的著作。— K. N. King乔治亚理工学院

这是唯一的一本为非程序员所写的 Lisp教材。它是一部非常重要的介绍性著作，使用最流行和最先进的 Lisp语支。

Lisp是人工智能研究和认知科学备受推崇的程序设计语言，它比PASCAL和 FORTRAN更易学习，比 BASIC功能更强。Lisp的交互式操作和直观的特性特别受初学计算机程序设计者的欢迎。

Touretzky的 Lisp“简明教程”一写的清晰，无难懂的技术术语，它使没有数学基础的读者也可以容易地学会程序设计。每一课都针对一个主要概念，结尾时都有一个要点小结并列出所讲的函数。这本书的显著特点是：

- 以交互式的方法演示程序的作用
- 有许多例子，短的笔头练习以及在计算机终端上进行学习的键盘练习
- 递归讲解得特别清楚
- 附有 Lisp语支和 Lisp的扩充部分

这本书对初学程序设计课程的学生是一本理想的教材。它必将引起对人工智能感兴趣的计算机用户和从事于认知科学工作的心理学家及心理系学生的极大兴趣。

目　录

引言：开始了解计算机

1. 美与计算机

现今市场上有数百种关于如何为计算机编制程序的书。本书与其它书的区别除了它使用 Lisp而外，还在于它把计算机程序设计作为一种美的活动来对待。计算机虽不是一种乐器，但是从稍微抽象的角度来看，实际上可以称它是二十世纪最伟大的键盘乐器。正象钢琴使我们在声音结构模型中创造美那样，计算机是在信息结构中寻找美的工具，计算机科学的许多美是数学的风味，但是不要因此而厌烦它，程序设计与普通的笔和纸的数学大不相同。

我相信程序员做的事情和音乐家做的事情之间存在着一些共同的东西。因此在写这本书的过程中，我从一本钢琴书的格式上借用了一些思想。

钢琴书有两部分：理论和键盘练习。乐理是对一些音乐元素的数学描述，它包括和声、和弦结构、配合旋律和节奏的格调形式上的描述。计算机科学理论涉及另一组形式的元素：信息结构、控制结构、算法和描述语言。

练习是单纯的理论理解到可表现出的技巧之间的转换。从理论上理解钢琴（或计算机）的性质到做一些实际有趣的事情是要经历一段很长的路程的。钢琴家和程序员们在他们各自的键盘上进行大量的实践，以便达到熟练的程度。通过本书中的各种练习可以邦助你增进自己的程序设计技能，直到成为一名计算机艺术的能手。

2.学习计算机的具体理由

计算机课程是现代学院里全部课程中迅速增长的领域之一。由于计算机在技术社会中起着重要的作用，因此每个人都需要一些计算机的基础知识。计算机程序设计也是培养形式推理技能极好的方法。人们学习形式推理的传统方法是通过学习数学（主要是几何学和逻辑学），但是由于这些学科很多都太难，太烦而使人不易感兴趣。然而，计算机程序设计是有趣的——甚至能使人上瘾，学生可以向计算机谈一些事情（用计算机语言），而计算机反过来给他们回答。

能对话的计算机可以帮助学生探索一些新想法，学生根据答话可以立刻发现这些想法是否行得通。他们通过亲自与计算机对话进行实验和发现一些新东西，我们也鼓励学生这样学习。计算机可以使一门学科变得生动活泼，而纯粹的笔与纸却使人感到抽象而单调。

3.计算机渗透到各个领域

计算机以多种方式存在，从比指甲还小的10美元的微处理机到可以放满一间房子的千万美元的发电机。计算机在所有机器中用途最为广泛，在我们的社会中使用计算机的方式有许多种，在此列出几种：

·**数据处理**。这是众所周知的计算机应用，数据处理是商人购买计算机的主要目的。他们用计算机做这样一些事情，如记录用户定货单，送出账单，管理报表，打印工资单及保

存人事记录使得日常的文书工作自动化。

· 嵌入应用。一些机器内部常装有小计算机，例如显示游戏机，袖珍计算器，电子存款机，以及程序控制的微波炉。甚至一些汽车现在也装有计算机：它控制油耗，保障平稳刹车，并使有较多信息的仪表盘能正常工作。

· 通信。我们的电话系统没有计算机控制开关设备是不可能工作的。计算机还能操纵通讯卫星和电子通信网络，象 ARPANET 能跨越半个地球连接一百多所大学、研究中心和政府计算站。一个人利用这个网可以在几秒钟内将电子邮件送给网上的另外一个人，而无需象常规邮寄那样花费几天的时间。飞机订票系统是计算机使通讯更加迅速的另一个例子。

· 文本处理。计算机已彻底改革了排版和编辑正文的过程。远比普通的打字机万能得多，一个计算机化的字处理程序让作者仅需按电钮就可以修改拼音错误，变换措词以及删除和插入一些句子以至完整的一段。还有你现在正读的这一页书就是用协同操作的计算机网络组织编辑，排版和印刷的。

· 信息检索。计算机非常善于快速地对大量信息进行查询。较大的图书馆都连接着计算机，这样就可以对医学或法律文摘数据库进行检索，几万个条目能在几分钟内检查完，节省了研究人员许多小时的手工工作。大的商店或公用事业公司也使用类似的检索技术，当他或她用电话询问有关服务事宜时，它们能立即检索顾客的记录。电话公司使用计算机协助号码操作员快速查找号码。

· 个人计算。由于计算机的价格继续不断下降，而它们的性能不断提高，计算机越来越多地进入到家庭中。人们使用计算机玩游戏，从传统的国际象棋，十五子游戏及blackjack到太空探险及pac— Man 。计算机还能做更多的事情，例如，预算计划和保存圣诞节卡片表。学会为他们自己的计算机编制程序的人们具有无限的机会来创造性地使用计算机。

4. 计算机作为研究的工具

计算机作为研究工具有几种不同的方式。它们可以分为处理数据，查询数据和产生数据。

处理数据的一个例子是统计分析，例如，线性回归或方差分析。这样一些事用手来做是困难的并且使人厌烦，但是用标准的统计计算机程序如SPSS或BMD可以做得相当快相当精确。

当计算机用于查询数据的时候，它进行的是有组织的查询，按字母顺序保存的一些表，或产生索引，或根据要求交叉引用各种表。一个复杂的系统就接受这样的查询“列出在三车间工作每年收入在 15000 美元以上的所有顾员的姓名和住址”；然后计算机从一个大的信息文件中只选择满足上述要求的那些记录。计算机帮助人们查询数据的另一种方法是自动生成曲线和图形。

计算机可以根据研究人员正在研究的一些理论或模型产生数据。例如，物理学家可以用描述原子核反应堆状态的详细方程式为计算机编制程序。然后，在确定初值条件之后，计算机就可以解这个方程并说明反应堆会产生什么情况。这比实际建立一个反应堆既快又便宜。如果物理学家想研究堆芯的熔化或大规模冷却系统的失效，那么计算机模型可能是安全地引导这项研究的唯一方法。

实际上，研究中的计算工作通常包括上述三项活动的组合。

5. 计算机与形式推理

形式推理是我们与周围世界打交道的一种方式。其它推理的风格，例如，翻译一首诗，发明一种烹饪方法或头脑中想象三维物体这类事情，它们虽不是形式的，但也一样的重要，形式推理之所以重要是因为它是现代科学和数学的基础，每一个受教育的人都应该熟悉这个领域。

我们应该怎样学习形式推理呢？通常是通过学习逻辑学或几何学以及学习证明定理。可以要求一个初学逻辑学的大学生证明如下的定理：

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

或者，在几何学课上，可能要“证明一条直线与两条平行线相交形成的内错角相等”。证明过程的形式部分是学生证明这个定理为真所用的方法。你不能说，“噢，从图上看这些角大约是相等，所以这个定理必须是真。”而且也不能从美学的观点来证明，比如说内错角应该是相等的，因为那样才能产生合意的对称。要形式地说明一种命题是真的唯一方法是提供一个逐步地合乎逻辑的证明来说明它是真。

许多人发现他们证明定理有困难，因而也就认为他们不能做形式推理。这很遗憾，因为他们切断了自己与许多吸引人的智力的学科的联系。定理证明不是学习形式推理的唯一方式，而且它甚至也不是应用形式推理的最有用的方式。其实，离开课堂以后，毕竟很少有机会需要你再去证明什么定理。我们以后将考虑形式推理的另一种途径：计算机程序设计。

在计算机程序设计中，目标不是证明定理而是写程序使计算机做某些事情。程序与定理之间的区别是程序更鲜明，更有生气。计算机要执行程序中的指令，而程序员可以监视计算机的动作看它是否在正确的进行。

计算机程序设计的形式部分是在程序的说明中，程序是用程序设计语言的简明逻辑符号写出来的。因为计算机在实际翻译程序并执行它的指令，所以发现程序中的错误比发现证明中的错误更容易。而且，计算机程序设计更直观，更易于学习。

6. 程序就是描述

一个计算机程序是对某种事物的描述。例如，一个程序可以是一个过程的描述。假设我们需要一个制作烤面包的过程，如果我们的指令是给人来读的，它们可以是很不规则的。我们可以描述制作烤面包的过程为：

1. 走到面包箱。
2. 你需要几块烤面包你就拿出多少片面包。
3. 把面包放到烤面包的电炉中并按下把手。
4. 等候烤面包跳出。

任何一个有理解能力的人都可以遵循这个过程做，但是计算机不能，因为过程太含糊。例如，如果你以前从未见过面包箱或烤面包的电炉，而这个过程没有告诉如何去识别它们。

如果电炉没接通电源，它也没有说该怎么办。如果你需要4片烤面包而烤面包炉只能容纳两片时，又该怎么办？它也没有说。如果你需要焦黄的烤面包，但是电炉烤出的是淡黄的面包，它也没有说怎么办？实际上，它甚至连烤面包是什么都没有说。

我们用Lisp写的过程将涉及称为数和符号的抽象事物，而不是烤面包的电炉和面包片，但是观点是相同的：如果要计算机完完全全地遵循我们的描述去做，则描述必须是精确完整的。

“程序描述什么？”对于这个问题一种可能的回答是程序描述机器。假如我们编制一个计算所得税的计算机程序，我们可以把该程序看作是对一个假想的计算所得税机器的描述。那么一台计算机正是一台通用的机器，它可以象任何一台专用机器一样起作用，并且已给予这台专用机适当的描述。

不论是把程序设计看作一个过程描写，还是看作机器制造，如果想要计算机成功地翻译我们的描述就必须仔细考虑。程序设计语言不同于普通人的语言，它们是为了写精确的（也是简明的）描述而专门设计出的。

7. 人工智能：机器真能思维吗？

在五六十年代把计算机称作为“电脑”是很流行的。就好象称汽车是“机械脚”一样。今天的计算机，即使是最大型的，应与人脑比较时它们也是幼稚简单的。它们和汽车一样不能思维，不能感觉或体验世界——它们仅仅象加法器那样计算一些东西，并显示其结果。

这并不意味着计算机从根本上就不能思维，它们究竟是能还是不能是一个争论得十分激烈的课题。然而目前存在的计算机都还没有大到或复杂到需要认真的考虑它们感觉能力的地步。

人工智能是计算机科学的一个分支，它牵涉到让计算机做有智能的事情，例如下国际象棋，理解口语，猜谜语或诊断疾病。今日的计算机对这些事情全能做（成功的程度不同），但是具有更深刻意义的人类意识方面的“思维”它们是做不到的。

练习

1. 列出在你的日常生活中你使用或接触计算机的五种方式。

2. 走进你最近的计算机商店，并问售货员为什么要买家庭计算机。他或她的论述能使你信服吗？你是否买了一台？

3. 一位经济学家用计算机模拟她的货币供应理论。她写一个计算机程序，给它输送一些当前的数据，并算出从现在开始六个月内她的国家经济状况的预测。说预测的结果是完全错误的。列出你认为导致这种状况的一些因素。

4. 一些人谬误地主张计算机永不可能思维，因为“计算机毕竟只能做人让它做的事情”。写一组告诉计算机如何思维的指令可能吗？你选择一种看法并简短地为之辩解。

第一章　函数和数据

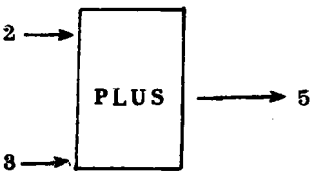
1.1 引言

本章开始介绍函数和数据的表示。数据这个术语意味着信息，例如，数、词或表。我们可以把函数看作为数据通过的一个盒子，它以某种方式对数据进行操作，其结果是从盒子里流出来的信息。

在讲究由 Lisp提供的内部函数之后，我们进一步学习如何将几个函数放在一起组成新的函数，这就是计算机程序设计的实质。本章的结尾介绍描述和组成新函数的几个有用的方法。

1.2　算术函数

人们熟悉的函数是加法、减法、乘法与除法这些简单的函数。在这里，我们是这样描述两个数的加法：

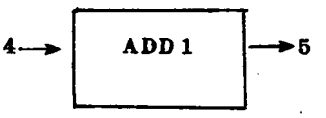


这个函数名是 PLUS，它取两个数 2 与 3 作为输入，而产生 5 作为它的结果。

下面列出一些常用的 Lisp算术函数：

PLUS	两个数加法
DIFFERENCE	从第一个减去第二个数
TIMES	两个数乘法
QUOTIENT	第一个数除以第二个数，返回商的整数部分
ADD1	一个数加上 1
SUB1	从一个数中减去 1

下面再看一个数通过函数的另一个例子：



数 4 进入 ADD1，加上 1 产生结果为 5。

1.3　整数除法

在这本书中，我们只使用整数。当一个整数被另一个整数除时，总是得到一个整数作为结果。如果除法有余数，则丢掉。然而，REMAINDER 函数可以用来计算余数，也就是它取余数作为结果，而整数商丢掉。例如：