〔美〕P.H.温斯顿
B.K.P.豪恩 著

LISP
程序设计

黄昌宁 陆玉昌 译
林尧瑞 校

清华大学出版社

内 容 简 介

LISP 语言是人工智能程序设计最常用的语言。本书分两部分。第一部分主要阐述 LISP 语言本身的内容及特点，是 LISP 的程序设计基础。第二部分论述了 LISP 语言在人工智能各个领域（诸如问题求解，搜索技术，模式匹配，定理证明，专家系统，自然语言理解及框架系统）中的应用，每一领域都列举了一些典型程序。全书二十三章，每章都有程序举例，除一、十三、二十一章外都有习题，书后附有习题答案。

本书是从事人工智能人员的主要参考书。也可供大专院校有关专业（计算机软件，模式识别，非数值运算）师生参考。

译者在书末还加上了"PDP-11机上 LISP 语言的使用说明"。

Patrick Henry Winston and Berthold Klaus Paul Horn
LISP
Addison-Wesley, 1981

LISP 程 序 设 计

P.H.温斯顿 B.K.P.霍恩 著
黄昌宁 陆玉昌 译
林尧瑞校

清华大学出版社出版
北京 海淀 清华园
轻工业出版社印刷厂 排版
河北固安印刷厂 印装
新华书店北京发行所发行 各地新华书店经售

开本：787×1092 1/16 印张：23.5 字数：604千字
1982年12月第一版 1983年5月第一次印刷
印数：1—40000
统一书号：15235·49 定价：2.90元

译 者 说 明

LISP 是一种计算机的表处理语言，是迄今在人工智能学科领域中应用最广泛的一种程序设计语言。人们认为这种语言曾武装了一代人工智能科学家。温斯顿本人认为 LISP 语言是人工智能的数学，不仅对人工智能的机器实现有重要意义，而且是人工智能理论研究的重要工具。

LISP 语言自1960年由 J. McCarthy 提出至今已有二十多年历史，不论早期的人工智能程序，还是近期著名的人工智能系统，大部分是用 LISP 语言写的。虽然近年来也出现过一些新的人工智能语言，但就其应用的广泛性和普遍性来看，LISP 仍是最主要的人工智能程序设计语言。

P.温斯顿和 B.霍恩写的这本书对 LISP 语言本身及其在人工智能各领域中的应用作了较为详细和系统的论述。对涉及到的人工智能各种问题都写出了具体程序。书中几乎每章都有大量习题，并附有习题解答，这对学习 LISP 语言和研究人工智能问题都有极大的帮助。读者除应弄懂书中的基本概念之外，应该仔细认真地去看每个程序。在条件允许的情况下上机实践一下书中的程序和习题，会对你有很大的帮助。

为了学习和介绍 LISP 语言，我们翻译了这本书。对书中我们发现的错误有的已改正，有的用 * 或 * * 在相应页上加以说明。对书中部分难看懂的程序我们也用〔注〕在相应页上加了译者注。

近年来很多单位进口了 PDP-11 系列计算机，为了帮助大家在 PDP-11 计算机上使用 LISP 语言，译者在书后加上“PDP-11 机上 LISP 语言使用说明”。

本书由黄昌宁和陆玉昌翻译。教研组的七九届研究生参加了部分翻译工作。最后由林尧瑞副教授校阅定稿。研究生贾培发同志还参加了“PDP-11机上 LISP 语言使用说明”部分的编写工作。在全书翻译和编写“LISP 语言使用说明”的过程中，得到了教研组很多同志的热情帮助，特别是石纯一副教授给了很多具体的指导和热情的鼓励，在此一并致谢。

由于我们对 LISP 语言使用和研究都不够，再加上外语水平的限制。书中一定会有不妥和错误之外，敬请读者指正。

1981年10月

目　录

第 二 部 分

前　言

本书分两部分。第一部分的目的是介绍 LISP 程序设计的基础。这一部分的前九章加上从其余三章中选出的材料作为 LISP 的导论科目是适宜的。读者学完了第一部分之后将能理解符号处理的基本概念，并且懂得在他们自己感兴趣的领域中如何使用 LISP，尽管这些应用比较简单。举例来说，编写搜索用的程序就不难。

第二部分的目的是通过实例来说明在实践中怎样使用 LISP。我们选择的例子都是为了有助于说明在人工智能及其有关领域中人所共知的一些概念。结合这些例子，我们讨论了好的程序设计实践的准则，基本的调试手段，数据驱动的程序设计，以及建立嵌入式的解释程序和编译程序。第二部分各章加上第一部分最后三章选出的材料作为怎样利用 LISP 的中高级科目是适宜的。读者学完了第二部分即可准备从事目前在人工智能中共同拥有的那些大系统，诸如问题求解，模式匹配，专家系统，自然语言理解及框架系统。

本书使用的 LISP 文本是 MACLISP。接触过 INTERLISP 而没有使用过 MACLISP 的人应该注意本书的一个附录，这个附录说明了在计算机辅助下做练习的不同方面。其余的附录提供了 LISP 函数的一个索引，给出了在终端上使用 LISP 人-机对话的一个实例，并对一些更加深奥的概念做了补充说明。

本书中的一些材料是从作者稍早的一本书——*Artificial Intelligence* 中摘录的。由于想不出更好的方法来说明某件事情，偶尔会有一段或一个例子被原封不动地转抄过来。但就大部分内容来说，借用过来的材料已经被广泛地修订、增补，并且一般地来说得到了提炼。

Artificial Intelligence 一书的第十一章——基本的 LISP 程序设计，被分散在本书的前八章中。

第十二章——积木世界，现在是第十三章和第十六章的一部分。第十四章和第十五章以新的方式建立在积木世界上。

第十三章——博弈世界，它介绍了 α-β 程序，现在被有关搜索的第十一章所代替。

第十四章——符号模式匹配，成为本书的第十七章，补充了一个用于命题演算的简单的归结定理证明程序。有关求解代数文字题的材料已经删去，DOCTOR 程序仅用作一个练习。

第十五章——嵌入式语言的实现，在这里几乎无从辨认了。ATN 编译程序现在是第二十章，在这一章之前的第十九章是 ATN 的解释程序，在这一章之后的第二十一章是一个数据库的接口。原先作为低级认识模型的产生式系统被第十八章的一个更一般的高级的似 MYCIN 演绎系统所代替。LISP 解释程序经过改进编在第二十三章中，这个改进涉及到变量辖域和闭包等论题。

第十六章——数据库与幽灵，已被第二十二章的框架系统所代替。

本书确实更新并取代了 *Artificial Intelligence* 一书的第二部分。以后我们也将要修订、补充、改进和分出该书的第一部分。可惜，按着本书的写法看来还需一段时间。

P. H. 温斯顿
B. K. P. 霍恩

第一部分

第一章　理解符号处理

本书分两部分，每一部分有其各自的目的。

▷第一部分的目的是介绍符号处理概念，并讲授 LISP 程序设计的基础。

▷第二部分的目的是阐明 LISP 具有的能力，并用 LISP 语言所做的事情来鼓励人们。

这一短章定义了符号处理的概念，说明为什么 LISP 是一种需要学习的适于符号处理的语言，并总览将要涉及的各种概念。

1.1　符号处理类似于处理词和句子

究竟什么是符号处理呢？每个人都知道，计算机做算术运算十分令人满意，但计算机怎样才能干更多的事情呢？如一台计算机如何去干一些有智能或好象是有智能的事呢？回答这个问题至少取决于这样一个事实：能把贮存在计算机里的数看成是其他一些事情的代码。确实，从某种对应关系来看，计算机里所有内容都是由若干个 0 和 1 组成的二进制数字串．每一位二进制数称作比特。一般，这些二进制数字都被翻译成十进制数字。但从另一种对应关来看，同样的二进制数字串可以译成字符，正像学过莫尔斯电码的人那样把点横序列译成字符。

一旦了解二进制数字串可以表示字符，那么这些字符的组合显然就可以表示单词，单词的组合就能表示句子，句子的组合就能代表一篇短文。通过越来越大的组合，最后能形成相当庞大的结构。这些二进制数字的组合最后就构成了节、章、书和百科全书。

在 LISP 中虽然每个单元的名字不同，但解释和分组想法是类似的。

▷在 LISP 中由二进制数形成的最基本内容是类似单词的对象，称作原子。

▷原子的组合形成表，表还可以组合在一起形成高级表。实际上这种按层次组合的能力是最重要的。

▷原子和表合在一起统称符号表达式。处理符号表达式正是 LISP 语言进行符号处理所涉及的内容。当然有时符号处理又称为表处理。

符号处理程序使用符号表达式来记忆和处理数据及过程。就像人们使用铅笔、纸和人类的语言来记忆和处理数据及过程一样。一个符号处理程序实际上有如下几段：识别特定符号表达式，分解旧的表达式，编排新的表达式。

这里给出两个符号表达式的例子。括号对标记出表的起点和终点。第一个例子描述用儿童积木搭成的某种结构。第二个例子描述某个大学。

```
(ARCH (PARTS  LINTEL  POST1  POST2)
      (LINTEL MUST-BE-SUPPORTED-BY  POST1)
      (LINTEL MUST-BE-SUPPORTED-BY  POST2)
      (LINTEL A-KIND-OF  WEDGE)
      (POST1  A-KIND-OF  BRICK)
      (POST2  A-KIND-OF  BRICK)
```

```
        (POST1  MUST-NOT-TOUCH  POST2)
        (POST2  MUST-NOT-TOUCH  POST1))
(MIT    (A-KIND-OF  UNIVERSITY)
        (LOCATION   (CAMBRIDGE  MASSACHUSETTS))
        (PHONE  253-1000)
        (SCHOOLS (ARCHITECTURE
                 BUSINESS
                 ENGINEERING
                 HUMANITIES
                 SCIENCE))
        (FOUNDER (WILLIAM  BARTON  ROGERS)))
```

当然这两个例子并不会使人感到吃惊。这仅仅是根据编排符号的惯例描述了某件事情。下面是另一个例子，这一次是描述一个规则，判断某个动物是不是食肉类动物。

```
(RULE  IDENTIFY6
   (IF      (ANIMAL  HAS  POINTED  TEETH)
            (ANIMAL  HAS  CLAWS)
            (ANIMAL  HAS  FORWARD  EYES))
   (THEN  (ANIMAL  IS  CARNIVORE)))
```

这里我们所看到的正是另一种表达概念的方法，带有锐利牙齿、爪和前视眼睛的动物大概就是食肉动物。使用这个规则相当于先把它分解，找到 IF 后面列举的条件，看看这些条件是否在信赖断言表上，如果这些条件在信赖断言表上，则把 THEN 后面的结论加到信赖断言表上。这种规则的使用就是符号处理的一个实例。

1.2 使计算机有智能，符号处理必不可少

近来不断出现有很多程序的组合成套设备，大多数人认为这种设备具有智能。这些有智能或好象有智能的程序几乎全部可用 LISP 语言来写。许多程序可能具有很大的实际价值。此处举一些例子：

▷专家问题求解程序。最早的 LISP 程序之一是解算大学一年级水平的微积分问题。另一个早期的程序是求解智力测验中用到的那些几何类比问题。在此之后，较新的程序已经能诊断血液的各种传染病，理解电子线路，为矿物勘探估计地质数据，编排有趣的数学问题等。所有这些程序都是用 LISP 写的。

▷常识性推理。人类的思维过程可能会遇到使用大量知识的推理。表示知识意味着选择一套符号词汇和确定编排它们的一些约定。一些好的表示刚好能使事情表达清楚。大多数知识表示的研究都是用 LISP 语言进行的。

▷学习。用计算机进行概念学习所做的工作并不多，但可以肯定已经做的大部分工作也依赖于表示方面的进展。在这方面 LISP 也处于主导地位。

▷自然语言接口。现在日益需要这样一种程序，它们能用英语或别的自然语言和人相互交换信息。计算机能完全理解自然语言大概是很遥远的事，但是已经建立了一些实际的有限范围的问答系统。这个范围包括月亮的岩石、海上的轮船，地毯公司的库存清单等等。

▷教育和智能的支持系统。为了和计算机的交互作用更加自如，人们必须制造这样一种计算机，人知道的事情它都知道，并且知道怎样告诉人们更多的知识。没有一个人在知道了很多知识之后还绕着大弯来解释问题，也没有一个人在一开始就想象电报那样来解释问题。基于LISP的程序开始是通过分析用户所做的事情来建立用户的模型。这些程序使用建立起来模型来选择或精心合成一些解释。

▷语音和视觉。已经证明，要了解人是如何实现听觉和视觉是非常困难的。我们似乎还不充分知道物质世界是如何迫使一种东西传到我们的耳鼓和视网膜的。虽然如此，目前正取得一些进展，而这些进展都是用LISP语言写的程序，当然也用了大量直接面向算术运算的程序设计。诚然，在面向算术运算的程序设计方面LISP没有特别的优点，但也没有什么不好的缺点。在算术运算方面LISP曾一度象糖蜜流那样慢，这一点出乎人们的意料。

由此可见，一个人要想了解计算机在某一方面的智能，他必须懂得LISP，这样他的了解才是完善的。此外，LISP还有其他方面的应用，有些应用也是很惊人的。至少以下几点值得提一提：

▷词处理。EMACS是功能很强的文本编辑程序。在MULTICS操作系统中的EMACS的一个版本就是用LISP写的。在现代专用计算机LISP机上用的ZWEI编辑程序也是用LISP写的。同样，许多文本的整版程序，即编排文本页面的程序也是用LISP写的。

▷符号数学。MACSYMA系统是为应用数学而搞的巨型系统。这套程序能处理代数问题，而完全代替铅笔和纸的作用。

▷系统程序设计。LISP机是完全用LISP编写程序的现代专用计算机。LISP机的操作系统、用户实用程序、编译程序和解释程序都是用LISP语言写的。成本可节省一两个数量级。

通过这些例子几乎每一个人都会很自然地接受下述观点：

▷符号处理是一种基础性的工具。计算机科学家和工程师必须懂得符号处理技术。

1.3 LISP正是要学的符号处理语言

现在程序设计语言非常之多，而仅有少数几种语言用于符号处理，其中LISP是最常用的。懂了LISP语言之后，其余大部分符号处理语言学起来也就很容易了。

为什么LISP成为符号处理最常用的语言呢？以后会不会更加如此。各种说法不一。下述的各种说法都有支持者：

▷人-机联系的观点。LISP要面向快速响应终端的程序设计。所有程序和所有数据都能随意的显示或修改。

▷环境的观点。需要得到最好编辑和调试手段的一些单位已经使用了LISP。二十多年来世界上最大的人工智能中心的研究人员围绕着LISP已经创立了非常完善的计算环境。制造出无与比伦的系统，可以用来编写大型的智能程序。

▷性能的观点。起初LISP是为符号处理而设计的。实际上，LISP是表处理语言(list processing language)的缩写。LISP正好有令人满意的性能。

▷一致性的观点。LISP函数和LISP数据有相同的形式。一个LISP函数可以分析另一个LISP函数，一个LISP函数甚至可以和另一个LISP函数组成一个整体，并加以使用。

可喜的是LISP是一种很容易学的语言。经过几个小时的学习就足以理解一些令人感到

吃惊的程序。先去学一点别的更普通的语言是完全不必要的。事实上，在某种意义上讲，了解某种其他语言会成为一种障碍。因为这可能使你建立一系列不良的习惯，其他语言处理问题的方式和LISP不同，把其他语言的函数逐个地翻译为LISP可能产生难以处理的结构。

LISP好学的另一个原因是它的句法极其简单。奇怪的是这是无意中产生的。LISP的发明者John McCarthy最初使用的一套LISP语言读起来就象读古英语那样困难。然而有一点值得提一下，他希望在函数和数据二者具有相同句法形式的上下文关系中使用LISP。LISP形成的最后格式很受欢迎，这种格式McCarthy当初仅想用来研究某些数学问题。很快LISP的形式就成为一种标准。

从历史上来看IPL先于LISP。IPL是五十年代在卡内基-梅隆大学由Newll, Show和Simon创立的。IPL介绍了许多基本概念，但IPL有它自己的用途，现在IPL基本上不用了。

1.4 本书第一部分介绍LISP

▷第二章，基本LISP函数。大约二十个基本函数构成了符号处理的基本词汇。本章介绍其中的一半以及一些用作算术运算的函数。

▷第三章，定义、谓词、条件式和变量的辖域。LISP能产生新的函数，这是很有趣的。这一章说明如何产生新的函数，同时也说明了被称为谓词的测试函数，以及有条件地做某件事情的一种方法。

▷第四章，递归和迭代。有时函数求解一个问题的最好方法是把问题分解成许多简单问题，然后再用这个函数本身来处理每个简单问题，在使用该函数时这种分解和处理问题的过程又可能反复地进行。这种方法就叫递归。另外，有时用一系列显式的测试-运算-测试的方法更好些。某件事情被重复进行，直到条件满足为止。这种方法叫迭代。

▷第五章，特性，A-表和数组。把通常用符号表达式表示的信息附加到原子上去的方法有多种。这一章介绍三种最普遍的方法：原子-特性-值三元组法，联结表法和数组法。

▷第六章，使用λ定义。定义仅仅出现一次而又几乎不值得给个名字的函数常常是很有用的。这一章说明怎样定义这样的函数。

▷第七章，打印，读和原子处理。LISP有READ和PRINT这两个简单的函数，以便向函数提供数据和给出输出结果。这一章就介绍它们。同时还对输入输出感兴趣的人们讲讲其他一些函数。

▷第八章，FEXPR和MACRO类函数的定义。在LISP中有几个函数有特殊用法，它们对给定的要处理的符号表达式进行非标准的处理。这一章将说明用户如何定义这样的函数。

▷第九章，表的存贮、回收和手术。知道表实际怎样去表示有时也是有用的。这一章说明了方盒-箭头表示法，并描述了一些危险的更改结构的函数。

▷第十章，有关二值图象的例子。LISP的应用不限于符号处理。这一章揭示了LISP语言的普遍规律，说明采用别的程序设计语言那样的方式来使用LISP语言是可能的，如果人们要这样做的话。这一章还介绍了数组的应用，并说明如何识别一个简单的物体。

▷第十一章，搜索的例子。许多问题求解工作要求对由节点和连接弧组成的网络进行搜索。这一章说明了如何实现最一般的搜索技术。同时也涉及一点分类问题。