

EXPERT SYSTEM

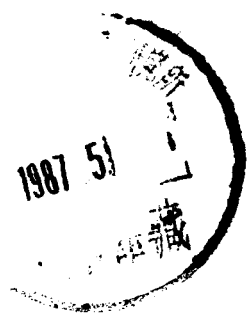
LISP语言

孙宗智 赵瑞清编著 · 气象出版社

73.8/221
487

LISP 语言

孙宗智 赵瑞清 编著



高等教育出版社

8710344

内 容 简 介

LISP语言是人工智能程序设计的常用语言,是实现人工智能程序设计的有力工具。本书简要介绍了GCLISP语言的主要内容,为初学LISP语言的读者提供了掌握这种语言的方法。在叙述上尽量与其它程序语言进行对照,并用实例说明了该语言的各种数据类型以及在程序设计方面的技巧。书中的程序已在IBM-PC XT微型机上调试。

本书是学习通用LISP语言的入门读物,可供计算机软件工作人员和计算机专业的大、专师生及人工智能有关人员参考。

55/30/20

LISP 语言

孙宗智 赵瑞清 编著

责任编辑:黄丽荣

* * *

化学工业出版社

(北京西郊白石桥路46号)

化工出版社印刷厂印刷

新华书店北京发行所发行 全国各地新华书店经售

* * *

开本: 787×1092 1/32 印张: 5.75 字数: 126千字

1986年2月第一版 1986年2月第一次印刷

印数: 1—10,000

统一书号: 13194·0384 定价: 1.35元

引 言

你想更有效地使用计算机吗？你想让计算机做一些具有智能的工作吗？掌握LISP语言是实现上述目的的途径之一。有人讲，LISP语言是人工智能语言，这并不过分，因为LISP语言是实现人工智能程序设计的有力工具。

LISP语言是由麻省理工学院的John McCarthy和他的研究小组在1960年设计实现的。20多年来，它广泛用于研究工作中，尤其是在人工智能系统的研制方面。

和其它计算机语言一样，LISP语言随着计算科学的不断发展而不断更新。前几年的LISP1.5，近几年的COMMON LISP是LISP语言的较完整版本。

本书向你介绍的GOLDEN COMMON LISP（简称GCLISP）语言是美国GOLD HILL COMPUTER公司于1984年底最新开发推出的一个在IBM-PC机（PC/XT，PC/AT及各种兼容机）上实现的人工智能语言，在PC-DOS2.0版本的操作系统支持下运行。

GCLISP语言是根据COMMON LISP语言的精髓部分设计而成。COMMON LISP是近年来美国几所大学（如MIT，STANFORD，CMU，RUTGERS，BERKELEY，UTAH，YALE等）和工业界（如Bell Lab等）的人工智能研究人员协同推出的，概括了以前诸版本的特点。GCLISP作为COMMON LISP在PC机上实现的第一个缩本，它具有COMMON LISP的通用性，它和COMMON LISP核心部分兼容，拥有若干先进的数据类型，效率较高，用户易于

掌握。

本书的宗旨仅为初学LISP语言的读者提供学习LISP语言的方法。对LISP语言中的一些概念和功能的介绍,尽量与其它语言相对照;同时突出LISP语言自身的特点。本书涉及的程序均已在IBM-PC机上实现。应该指出,本书只介绍GCLISP中主要的数据类型和系统函数,想全面了解GCLISP的读者请查阅《GOLDEN COMMON LISP REFERENCE MANUAL》。

书中难免出现一些差错,敬请读者批评指正。

目 录

引言

第一章 LISP语言的程序和数据	1
1.1 基本数据类型S-表达式	1
1.2 LISP的程序结构	3
1.3 LISP语言的功能	6
练习一	8
补充一	8
第二章 符号原子	10
2.1 变量和赋值	10
2.2 符号原子的特性值和特性表	14
2.3 解释程序对LISP式求值	16
练习二	18
补充二	19
第三章 表处理 (一)	22
3.1 CAR和CDR分别取表头、留表尾	22
3.2 表的构造和修改	24
3.3 序列	26
3.4 表的存贮的抽象表示	28
练习三	29
补充三	30
第四章 谓词和数	33
4.1 谓词	33
4.2 数	38

练习四.....	43
补充四.....	43
第五章 控制结构	45
5.1 基本控制式COND及其它分支控制.....	45
5.2 PROG及RETURN, GO	47
5.3 DO循环	49
练习五.....	51
补充五.....	52
第六章 自定义函数.....	58
6.1 DEFUN定义一个有名函数.....	58
6.2 简单的LISP程序举例	60
6.3 LAMBDA-表达式.....	64
6.4 泛函数FUNCALL, APPLY和MAPCAR	66
练习六.....	69
补充六.....	72
第七章 基本的输入输出函数.....	77
7.1 基本输入函数READ和READ-LINE	77
7.2 基本输出函数PRINT, PRIN1和PRINC	78
7.3 FORMAT格式输出	80
7.4 例	82
练习七.....	84
补充七.....	85
第八章 变量的类型、闭包和宏	88
8.1 约束变量和自由变量	88
8.2 函数闭包	91
8.3 宏	94
8.4 SETF	96

练习八	99
补充八	100
第九章 表处理 (二)	102
9.1 PUSH和POP是堆栈操作	102
9.2 改变自变量的表结构的函数	105
9.3 联结表	107
练习九	108
补充九	110
第十章 数组、串和结构	114
10.1 数组	114
10.2 字符串	117
10.3 结构	119
10.4 例	123
练习十	126
补充十	126
第十一章 流和文件	131
11.1 一个生成顺序文件的LISP程序	131
11.2 流和路径	132
11.3 输入输出语句的完整形式	134
11.4 流可以作为内部文件	135
练习十一	137
补充十一	138
第十二章 LISP程序例	142
12.1 梵塔问题	142
12.2 把集合的元素分成等价类	144
12.3 编译简单的算术表达式	146
12.4 实现搜索	148

12.5	分类问题	152
12.6	皇后问题	154
12.7	拓扑排序	157
12.8	自修改函数例	163
附录 I	源程序的编辑	166
附录 II	GCLISP解释程序工作方式	168
附录 III	本书介绍的GCLISP中的系统函数、特殊符 号、宏及系统变量索引	170
参考文献		174

第一章 LISP语言的程序和数据

一种程序语言的核心是如何设计各种程序实现对数据的处理,以解各类问题。这章主要通过实例介绍LISP语言的程序的结构和基本数据类型。它是后继各章的纲要,这章涉及的概念将在以后详细讨论。

1.1 基本数据类型S-表达式

LISP语言最初设计的目的主要用于象定理证明那样的人工智能领域。以后不断发展、完善,到现在已经形成了通用的语言,即COMMON LISP。

在通用LISP语言中可以说包括了象PASCAL等语言的各种数据类型,但LISP语言中的S-表达式仍然还是它的基本数据类型。

S-表达式也叫符号表达式,是原子(ATOM)和表(LIST)的总称。

原子是一个一般符号的序列,比如:ATOM, B27, DICK, 12FE, 123, 3.14都是原子。但特殊的一些符号,比如空格,括号“(”和“)”,反叙杆“\”等在LISP语言中有特殊含义,因此它们不能直接地用于构造原子。下列都不是原子:

$A \sqcup B C, \backslash X Y, (M).$

空格在LISP中作为分隔符来用;“/”用于文件的路径名中;而括号是表的专用符。其它有特殊意思的符号暂不介绍。

在原子中,象123, 3.14是数(关于LISP中的数将在后边

详细讨论);不是数的原子称为符号原子。于是ATOM, 12FE都是符号原子;早期的LISP限制符号原子必须用字母开头,但有些LISP语言已经不受这个限制了。符号原子也简称为符号。

表是由括号和原子构成的,可以这样定义:

- 1. () 是表;
- 2. (<表或原子><表或原子>...<表或原子>)是表。

注意,两个原子相邻必须用空格分开。

于是 (), (A B), (A(X)((Y)))都是表。

() 称为空表,用NIL表示它。而NIL本身是符号原子。在LISP语言中只有NIL既是原子又是表,并且NIL恒等于 ()。

值得注意的,一个表中的左、右括号要符合括号匹配规律,否则出错。

表是可以任意嵌套的结构,表中可以有表。从外到里可

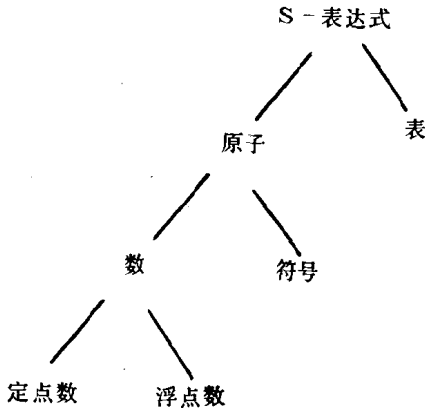


图 1.1 LISP 的基本数据

以分为0层, 1层, ...; 最外层括号括起的原子或子表称作表的元素, 例如:

(A (2) (D (3 5)))

的元素是A, (2), (D(3 5))。

我们将会看到S-表达式可以用以描述任何事物及其属性; 可以表示复杂的知识。LISP作为这种程序语言的名称就来自于表处理 (LIST PROCESSING)。LISP的基本数据可由图1.1表示。

1.2 · LISP的程序结构

LISP作为人工智能程序设计语言, 它的程序往往比较复杂, 但就其程序结构讲却十分简单。从形式上看LISP程序是S-表达式的有限序列, 让我们看一个简单例子:

```
(DEFUN POWER (M N)
  (COND((ZEROP N) 1)
        (T(* M(POWER M(1- N))))))
(SETQ X (READ))
(SETQ L (READ))
(POWER X L)
```

这个简单的LISP程序; 由4个S-表达式组成。

从程序构成的成份看一个LISP程序包含三个实体:

1. 数据 (DATA), 它作为被操作的对象;
2. 表达式 (EXPRESSIONS), 在程序运行中要对它求值;
3. 函数调用。

上述LISP程序中第一个式子给出幂函数 m^n 的定义, 它类似于其它语言中的函数子程序; 第二、三个LISP式子, 分

别给变量（即符号原子 X, L）赋值，其值（数据）由键盘敲入，它们类似于其它语言的赋值语句；第四个式子则是对已定义的 POWER 函数的调用。

我们可以粗略地认为 LISP 式子与其它某些语言中的语句一样。

上述程序中出现的 DEFUN, SETQ, READ 是 LISP 系统提供的函数，以后称 LISP 解释系统提供的函数为系统函数（或基本函数）。GCLISP 中大约提供 400 多个系统函数，本书后几章将介绍其中常用的，有些在每章的补充中介绍。由 DEFUN 定义的函数是用户自定义函数。一般，一个 LISP 程序往往要定义若干函数，然后对它们调用，从而完成一定计算任务。

在 LISP 程序中没有“语句”术语。相当于其它语言中的“语句”，一律采用 S-表达式形式，即调用函数来实现。另外 LISP 程序中也没有过程（PROCEDURE），相当于它的功能通过定义函数实现。

因此 LISP 程序可以粗糙认为是一系列函数的调用。下边我们给出 LISP 程序的严格定义，定义中涉及的概念将在后面精确定义。

一个 LISP 程序是按顺序排列的有限个 LISP 式子（FORMS）。

LISP 式子（FORM）是可以被合法求值的对象，有以下几种：

- 常数（其值为自身），数、字符串，T，NIL；
- 变量，已赋值的符号原子；
- 特殊式子（见本章的补充），形为非空表；
- 宏调用，形为非空表；

函数调用，形为非空表。

由上讨论可见，LISP的程序与LISP的基本数据一样，都采用S-表达式这一形式，至于在一个程序中哪些S-表达式是作为数据自然有办法认定，将在下边讨论。另外，LISP程序中的数据一般采用自说明的形式，即LISP程序中一般不对数据说明，LISP解释程序根据数据自身的特征识别其类型。

但在有编译功能的COMMON LISP中，为给编译程序提供必要的信息，要有必要的说明函数。目前GCLISP只有解释功能。

为增加LISP程序的易读性，程序中允许有注释。在一行中出现分号“;”，则该分号以后直到该行结束为止部分为注释部分。LISP解释程序不对一行中的分号“;”及注释部分予以解释，注释部分仅在列程序清单时有用。下边是有注释部分的LISP程序：

```
; A EXAMPLE OF LISP PROGRAM
(SETQ X (READ)) ; INPUT AND ASSIIGN
(PRINT X)       ; PRINT
```

该程序被解释执行的部分是：

```
(SETQ X (READ))
(PRINT X )
```

在程序中的一个LISP式子可以分在几行中写；一行中也可以写几个LISP式子。LISP程序中的一行由“回车”结束。一个或连续几个空格以及括号都起分隔符作用。

值得注意的是：LISP中的函数调用、宏调用以及特殊式子均采用前缀形式，即函数、宏、特殊式子名在前，其实际参数在后。采用表的形式，函数（宏、特殊式子）名及各

参数间用空格分开，例如：

```
*(+ 2 3)
```

5

这里LISP式子前出现的* 是LISP解释状态的提示符，自动在屏幕上显示。* 后边是用户敲入的LISP式，如上(+ 2 3)，LISP解释程序对它求值（即对它解释）然后在下一行显示调用该函数所得的函数值，如对(+ 2 3)求值结果为5 在下一行显示。如把(+ 2 3)写成(2 + 3)则出错。再如：

```
*( * 2 3 )
```

6

```
*(SETQ X 5)
```

5

关于SETQ在下章讨论，它可以完成变量X的赋值，使其值为5，而它本身也是一个函数，其值为5。

1.3 LISP语言的功能

前二节我们注意到LISP语言中的基本数据类型和它的程序都采用S-表达式形式。我们将会看到LISP解释程序有办法清楚地认定S-表达式什么情况是作为数据，什么情况是作为程序。一般两者之间是可以转化的：当对数据执行求值时，数据就变成程序，同时把要修改的程序看成数据。

当前COMMON LISP语言，可以说已具备了象PASCAL那样语言中的先进的数据类型，可以完成象一般数据结构书中介绍的对各类数据的操作，其程序设计是十分方便的。但LISP语言从它最初形成起就把表处理(LIST PROCESSING)定为自己的基本功能。学习LISP除了用它实现其它语言能胜任的任务之外，重要的是学习它的符号处理技术，让我们先

举两个例子。

一个有向图如图1.2所示。

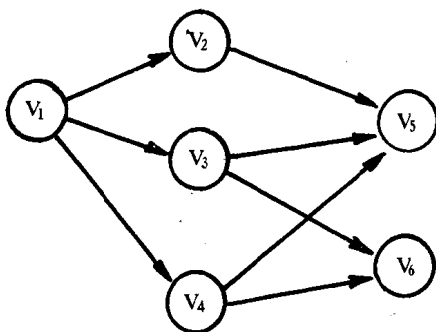


图 1.2 一个有向图的例子

我们用S-表达式很容易表示它：

```
(V1(V2 V3 V4)V2(V5)V3(V5 V6)
V4(V5 V6))
```

我们将会看到，LISP所提供的表处理功能，很容易对有向图进行各种处理。

在人工智能领域往往要涉及推理，下边是用S-表达式给出的一个规则：

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

在设计一个象专家系统或者问题解答系统时，往往要进行知识存贮和建立推理规则，LISP语言将提供十分清楚、严密的模型。基于其上的表处理技术将容易实现相应的程序

设计。

总之,LISP语言不仅能作为十分有力的人工智能程序设计语言;而且它提供的数值处理与非数值处理的功能较其它较好的程序设计语言(比如PASCAL, FORTRAN77, 扩展BASIC)毫不逊色。

练 习 一

1. 下面的各式, 哪些是原子、表, 哪些是错误的表达式:

ATOM
(THIS IS A S-EXPRESSION)
((A B) (C D))

3.14

((()))
(A, B)
((())
()) (
(A B

2. 把下面用表表示的二叉树画出来, 规则是:

叶是树, 用原子表示;

一个节点是树用表表示: (<表示节点的原子> <左子树>
<右子树>)

(* (+ A B) (- C (/ D E)))

补 充 一

1. GCLISP中的数(NUMBER)、一维数组(ARRAY)、结构(STRUCTURES)、字符串(STRING)、流(STREA-