人工智能原理与应用

专家系统·机器学习·面向对象的方法

田盛丰　等编著

北京理工大学出版社

人工智能原理与应用

人工智能原理与应用

——专家系统、机器学习、面向对象的方法

田 盛 丰　等编著

北京理工大学出版社

(京)新登字149号

内 容 简 介

本书系统地介绍了人工智能的基本原理，构造专家系统的实用技术，以及人工智能研究的一些前沿课题，包括近年来国际上开始流行的一些新技术，如面向对象的方法，基于模型的推理，基于事例的推理，遗传算法等内容。

全书共分十章，前五章介绍了人工智能的基本原理与方法，包括程序设计语言，知识表示，搜索策略和演绎推理，后四章详细介绍了建造专家系统方面的一些深入的课题，包括非单调与不精确推理，系统结构，知识获取与机器学习。最后一章详细地剖析了一个新型的面向对象的专家系统构造工具OEC系统。

本书注重实用性与先进性，并附有习题。可作为大学计算机及有关专业高年级和研究生的教材，也可供有关科研人员参考。

人工智能原理与应用
——专家系统、机器学习、面向对象的方法
田盛丰 等编著
*
北京理工大学出版社出版发行
各地新华书店经售
一二〇一工厂印刷
*
787×1092毫米 16开本 22.75印张 562千字
1993年8月第一版 1993年8月第一次印刷
ISBN 7-81013-735-2/TP·84
印数：1—6000册 定价：14.00元

前　言

人工智能是目前获得迅速发展的一门新兴学科。在人工智能领域的研究中，专家系统是其中最活跃和应用最广泛的一个分支。专家系统已广泛应用于工业、农业、气象、地质、医疗卫生、交通运输、数学、物理、化学、军事等各个领域，各个领域纷纷研究建立本专业的专家系统。本书试图从人工智能的基本原理出发，介绍建造专家系统的实用技术，以期推动专家系统在我国的应用和发展。

近年来，人工智能的基本原理虽然没有重大的突破，但是新的思想和新的方法层出不穷，推动了人工智能学科的飞速发展。例如，面向对象的方法，基于模型的推理，基于事例的推理，各种机器学习算法等等，在80年代和90年代开始获得了广泛的应用，本书试图对这些研究成果给予较系统的介绍。

本书是在编者长期从事本科生和研究生人工智能课的教学实践的基础上，以及长期从事专家系统研究工作的基础上产生的。涉及基本原理的各章都附有习题，可作为大学计算机及有关专业高年级和研究生的教材，也可供从事专家系统开发研究的科技人员参考。

本书共分十章。第一章为绪论，第二章介绍常用的人工智能程序设计语言，包括LISP语言、PROLOG语言及面向对象的程序设计的基本概念。第三章至第五章叙述人工智能的基本原理与方法，包括知识表示、搜索方法和推理方法等。第六章介绍实用的推理技术，包括非单调推理和不精确推理。第七章介绍专家系统的结构，包括元知识系统、黑板系统和黑板控制系统等。第八章与第九章介绍知识获取和机器学习的方法。第十章介绍编者研制的一种新型的面向对象的专家系统构造工具，以供读者参考。本书还附有该系统的软件作为引玉之砖，以迎接读者创建的更先进的系统。

本书第二章第2.2节由陈峰执笔，第九章由任建宏执笔，其余各章节由田盛丰执笔，全书由田盛丰统稿。在本书的编写过程中，还得到温熙丹老师的热情帮助，在此表示感谢。对于本书中的错误和不足之处，恳请各位专家和广大读者指教。

田盛丰

1992年12月于北方交通大学计算机系

目　录

第一章　绪　论

1.1　人工智能的发展概况……（1）

1.1.1　什么是人工智能……（1）

1.1.2　人工智能的研究途径……（1）

1.1.3　人工智能学科的发展历史……（1）

1.2　人工智能的应用……（2）

1.2.1　知识工程……（2）

1.2.2　专家系统……（2）

第二章　人工智能程序设计语言

2.1　LISP语言……（5）

2.1.1　LISP语言概述……（5）

2.1.2　LISP的基本功能……（7）

2.1.3　递归与迭代……（15）

2.1.4　输入输出功能……（19）

2.1.5　LISP的其它功能……（21）

2.2　PROLOG语言……（23）

2.2.1　PROLOG语言概述……（23）

2.2.2　PROLOG语言程序设计基础……（24）

2.2.3　重复和递归的方法……（31）

2.2.4　表处理方法……（35）

2.2.5　建立数据库的方法……（37）

2.2.6　字符串处理方法……（38）

2.2.7　输入输出设计……（39）

2.2.8　PROLOG的其它功能……（40）

2.3　面向对象的程序设计……（42）

2.3.1　概述……（42）

2.3.2　面向对象程序设计的基本思想……（43）

2.3.3　基于类的系统……（44）

2.3.4　基于原型的系统……（48）

2.3.5　混合系统……（50）

2.3.6　面向对象的计算模型……（51）

习题……（53）

第三章　知识表示

3.1　概述……（55）

3.1.1　知识与知识表示……（55）

3.1.2　知识的表示方法……（56）

3.2　逻辑表示法……（57）

3.2.1 一阶谓词逻辑……(57)
3.2.2 谓词逻辑用于知识表示……(60)
3.3 规则表示法……(61)
3.3.1 产生式规则与产生式系统……(61)
3.3.2 Markov算法和Rete算法……(64)
3.3.3 控制策略的类型……(65)
3.4 语义网络表示法……(66)
3.4.1 语义网络的基本概念……(66)
3.4.2 语义网络的应用……(69)
3.5 框架表示法……(72)
3.5.1 框架的基本概念……(72)
3.5.2 框架表示的应用……(73)
3.6 概念从属与剧本表示法……(74)
3.6.1 概念从属……(74)
3.6.2 剧本……(77)
习题……(79)
第四章 基本的问题求解方法
4.1 状态空间搜索……(80)
4.1.1 概述……(80)
4.1.2 回溯策略……(84)
4.1.3 图搜索策略……(87)
4.1.4 任一路径的图搜索……(90)
4.1.5 最佳路径的图搜索……(90)
4.1.6 与或图的搜索……(97)
4.2 博弈树搜索……(101)
4.2.1 概述……(101)
4.2.2 极小极大过程……(102)
4.2.3 α-β过程……(103)
4.3 通用问题求解……(106)
4.3.1 手段目的分析……(107)
4.3.2 生成与测试……(109)
4.3.3 约束满足……(110)
习题……(113)
第五章 基本的推理方法
5.1 归结反演系统……(115)
5.1.1 谓词演算基础……(115)
5.1.2 归结反演……(119)
5.1.3 归结反演的控制策略……(121)
5.1.4 从归结反演中提取解答……(123)
5.2 基于规则的演绎系统……(127)
5.2.1 正向演绎系统……(128)
5.2.2 逆向演绎系统……(133)
5.3 规划生成系统……(136)

5.3.1 机器人问题求解……(126)
5.3.2 正向系统……(127)
5.3.3 规划的表示……(133)
5.3.4 逆向系统……(140)
习题……(144)
第六章 实用推理技术
6.1 推理的类型……(146)
6.1.1 从逻辑基础上的分类……(146)
6.1.2 从推理方法上的分类……(148)
6.2 非单调推理……(149)
6.2.1 概述……(149)
6.2.2 非单调逻辑……(150)
6.2.3 非单调系统……(152)
6.3 不精确推理……(155)
6.3.1 概述……(155)
6.3.2 可信度方法……(159)
6.3.3 主观Bayes方法……(163)
6.3.4 证据理论……(169)
6.3.5 可能性理论……(177)
6.4 基于模型的推理……(185)
6.4.1 基本原理……(185)
6.4.2 基于规则与模型的系统……(188)
6.5 基于事例的推理……(189)
6.5.1 基本概念……(189)
6.5.2 基本方法……(190)
6.5.3 与基于规则的系统的比较……(191)
习题……(191)
第七章 专家系统的结构
7.1 基本结构……(192)
7.2 元知识系统结构……(194)
7.2.1 什么是元知识……(194)
7.2.2 元知识的作用……(194)
7.2.3 元知识在专家系统中的应用……(196)
7.3 黑板系统结构……(199)
7.3.1 黑板模型……(200)
7.3.2 黑板结构……(201)
7.3.3 知识源……(202)
7.3.4 控制策略……(204)
7.3.5 黑板模型的优越性……(205)
7.4 黑板控制结构……(205)
7.4.1 基本概念……(205)
7.4.2 知识源的表示……(205)
7.4.3 控制黑板的组织……(206)

7.4.4 调度机制 ……(209)
7.4.5 黑板控制结构的优点与不足 ……(211)
7.5 实例研究 ……(211)
7.5.1 MYCIN系统 ……(211)
7.5.2 AM系统 ……(218)
习题 ……(222)
第八章 知识获取与机器学习
8.1 概述 ……(223)
8.1.1 知识获取的基本过程 ……(223)
8.1.2 知识获取的主要手段 ……(225)
8.1.3 机器学习 ……(226)
8.1.4 知识获取工具 ……(228)
8.2 通过例子学习 ……(235)
8.2.1 概述 ……(235)
8.2.2 学习单个概念 ……(230)
8.2.3 学习多个概念 ……(247)
8.2.4 学习执行多步任务 ……(257)
8.3 通过类比学习 ……(262)
8.3.1 概述 ……(262)
8.3.2 类比学习与推理系统 ……(264)
8.3.3 转换类比与派生类比系统 ……(266)
8.4 基于解释的学习 ……(273)
8.4.1 概述 ……(273)
8.4.2 基于解释的抽象 ……(274)
8.5 通过观察学习 ……(276)
8.5.1 合取概念聚类系统 ……(276)
8.5.2 结构对象的概念聚类 ……(281)
习题 ……(285)
第九章 遗传算法
9.1 遗传算法分析 ……(286)
9.1.1 概述 ……(286)
9.1.2 遗传算法的理论基础 ……(290)
9.1.3 算法实现中的一些基本问题 ……(294)
9.1.4 高级算子技术 ……(297)
9.2 基于遗传的机器学习系统 ……(303)
9.2.1 基于遗传学的机器学习 ……(304)
9.2.2 一个分类器系统的实现 ……(308)
第十章 面向对象的专家系统构造工具OEC
10.1 面向对象的知识表示 ……(314)
10.1.1 概述 ……(314)
10.1.2 对象的表示 ……(314)
10.1.3 规则的表示 ……(318)
10.1.4 方法的表示 ……(321)

10.2 问题求解机制 ……(326)
10.2.1 概述 ……(326)
10.2.2 模糊规则推理 ……(326)
10.2.3 模糊决策树推理 ……(332)
10.2.4 神经网络的模拟 ……(336)
10.3 系统的开发 ……(338)
10.3.1 用户界面 ……(338)
10.3.2 知识库开发工具 ……(342)
附录 OEC系统函数 ……(344)
参考文献 ……(349)

第一章 绪 论

1.1 人工智能的发展概况

人工智能(Artificial Intelligence)简称为AI，它是计算机科学的一个重要的研究领域。近二十几年以来获得了迅速的发展，在很多领域都获得了广泛的应用。

1.1.1 什么是人工智能

斯坦福大学人工智能研究中心的Nilsson教授认为："人工智能是关于知识的科学——怎样表示知识以及怎样获得知识并使用知识的科学。"MIT的Winston教授指出："人工智能就是研究如何使计算机去做过去只有人才能做的智能的工作。"这些定义反映了人工智能学科的基本思想和基本范围。事实上，从广义上来讲，一般认为用计算机模拟人的智能行为就属于人工智能的范畴。从狭义上讲，人工智能方法是指人工智能研究的一些核心内容，包括搜索技术、推理技术、知识表示、机器学习与人工智能语言等方面。这些就是本书所要讲述的主要内容。

1.1.2 人工智能的研究途径

对人工智能研究的不同途径来源于对人类智能的本质的不同认识，并由此产生出两大学派：符号主义(Symbolicism)与连接主义(Connectionism)。符号主义认为人类认识的基本元素是符号，认识过程就是一种符号处理过程。因此符号主义的研究者注重于用符号来描述人类的思维过程。事实上，人类的语言本身就是用符号表示的，人类的很多思维活动如决策、计划、设计、诊断等活动都可以用语言来描述，因此也就可以用符号来表示。符号主义在模拟人类这方面的思维活动中获得了很大的进展。但是，对有些思维活动，如图像识别，就难以用语言来描述。对这些问题的解决，符号主义就遇到了很大的困难。

连接主义根据对人脑的研究，认为认识的基本元素就是神经元本身，人类的认识过程就是大量的神经元的整体活动。从本质上讲，是并行的分布式的处理模式，在上述的图像识别等方面，连接主义的模型显示了更大的优越性。

实际上，人类的思维过程是非常复杂的，上面的两种观念哪一种也不能作出完全的解释。有人提出，人类的思维是分层次的。高层次的思维是抽象思维，适用于规划、决策、设计等方面；低层次的思维是形象思维，适用于识别、视觉等方面。符号主义和连接主义两种研究的途径反映了人类思维的两个层次，彼此不能互相代替，而应当互相结合。本书主要介绍符号主义的成果，以及两者互相结合的系统。

1.1.3 人工智能学科的发展历史

"人工智能"这一术语，是在1956年由McCarthy和Minsky等人发起的关于用机器模拟智能的学术讨论会上提出的，这就标志着人工智能这一学科的诞生。1957年，Rosenblatt研制了

感知机，这是一种将神经元用于识别的系统，它的学习功能引起了广泛的兴趣，推动了连接主义的研究。但是很快发现了感知机的局限性，不能解决复杂的识别问题，从而连接主义又转入低潮。从60年代到70年代，符号主义的研究获取了很大发展。特别是DENDRAL、MYCIN和PROSPECTOR等有名的专家系统的研制成功，使人工智能开始走向实际的应用。在80年代，连接主义也有了突破性的进展，特别是Hopfield提出的网络模型和Rumelhart等人提出的多层网络学习算法，又把连接主义的研究推向高潮。在这一时期，符号主义的研究也取得了很大进展。在90年代，必将是人工智能技术获得广泛应用的时代。

1.2 人工智能的应用

1.2.1 知识工程

实用的人工智能称为知识工程，知识工程的概念产生于70年代中期。1977年美国斯坦福大学的Feigenbaum教授在第五届国际人工智能会议上提出了"知识工程"这一概念。他以"人工智能的艺术：知识工程的课题及实例研究"为题，对知识工程作了全面的论述。他认为："知识工程是人工智能的一种技艺，它运用人工智能的原理和方法，对那些需要专家知识才能解决的应用难题提供求解的手段。恰当运用专家知识的获取、表达和推理过程的构成与解释，是设计基于知识的系统的重要技术问题。"知识工程这一概念的提出，标志着人工智能的研究进入实际应用的阶段。

目前，知识工程研究的主要内容包括：

(1) 专家系统　模拟人类专家的问题求解过程，解决那些只有专家才能解决的专门问题。

(2) 知识库系统　对人类的知识进行存储、加工、管理，并根据需要对知识进行处理和应用。

(3) 决策支持系统　利用模型和知识，通过模拟和推理等手段，为人类的活动进行辅助决策。

(4) 自然语言理解　理解人类的自然语言，以实现人和计算机之间自然语言的直接通讯，从而推动计算机更广泛的应用。

(5) 智能机器人　具有感觉、识别和决策功能的机器人。

(6) 模式识别　模拟人类的听觉、视觉等感觉功能，对声音、图像、景物、文字等进行识别。

(7) 自动程序设计　由计算机完成程序的验证和综合，实现程序设计自动化。

当前，知识工程最主要的应用就是建造专家系统。

1.2.2 专家系统

专家系统(Expert System)简称ES，是目前在人工智能的应用方面最成熟的一个领域。

1. *专家系统的产生*

早在60年代初，人们就已开始研究用人工智能技术解决实际问题。通用求解器GPS就是一个典型的例子。但是人们很快发现，客观世界是相当复杂的，企图用一种普遍适用的模式去解决所有的问题是不可能的，因此开始转向比较狭窄的专门领域。1965年，Feiqenbaum研

制了第一个专家系统DENDRAL，从而开创了专家系统的研究。目前，专家系统的应用领域已扩展到数学、物理、化学、医学、地质、气象、农业、法律、教育、交通运输、机械、艺术以及计算机科学本身，甚至渗透到政治、经济、军事等重大决策部门。

2. 专家系统的特点

什么是专家系统？Feigenbaum认为："专家系统是一种智能的计算机程序，它运用知识和推理步骤来解决只有专家才能解决的复杂问题。"也就是说，专家系统提供了一种新型的程序设计方法，可以解决传统的程序设计方法所难以解决的问题。专家系统的特点大致有以下三点：

(1) 专家系统所要解决的是复杂而专门的问题。对这些问题，人们还没有精确的描述和严格的分析，因此还没有确定的算法来解决。解决这些问题需要专家的知识，包括理论知识和实际经验。

(2) 专家系统突出了知识的价值。通常要推广和应用专家的知识，要通过培训的方法，这需要若干年的时间。专家系统则大大减少了知识传授和应用的代价，使专家的知识迅速转变成社会的财富。

(3) 专家系统采用的是人工智能的原理与技术。如符号表示，符号推理、启发式搜索等等，这是与一般的数据处理系统不同之处。

总之，专家系统注重于知识本身而不是确定的算法。

3. 专家系统的分类

专家系统从其完成的功能分类，可包括诊断、预测、解释、调试、修理、规划、设计、监督、控制等多种类型。这些功能又可分为两大类：分析型和综合型。分析型专家系统所要解决的问题有明确的、有限个数的解，系统的任务在于根据实际的情况选择其中一种或几种解。这种专家系统的功能可包括分类、诊断、预测、修理以及部分设计、规划等方面。综合型专家系统的任务是根据实际的需要构造问题的解，包括设计、规划等类问题。另外，也可以根据知识的特征和推理的类型对专家系统进行分类。

4. 几个有名的专家系统

DENDRAL系统是1965年研制的第一个专家系统。它用于根据有机化合物的分子式和质谱图判断分子结构，当给出一个有机化合物的分子式时，其可能的分子结构往往有几千种。对于给定的有机化合物的分子式和质谱图，系统根据化学家的知识和质谱仪的知识，删掉不可能的结构，从几千种可能的分子结构中挑选出一个正确的分子结构。这个系统已经达到了化学博士的水平。

MYCIN系统是一个医疗诊断系统，用于在不知道原始病原体的情况下，判断如何用抗菌素来处理血液细菌感染病患者。系统输入的原始数据是患者的症状、一般情况、病史和化验结果。系统根据专家的知识和输入数据判断出是什么病菌引起的感染，再作出抗菌素的配方。MYCIN于1974年基本完成，是第一个功能较全面的专家系统，它不仅具有很高的性能，而且具有解释功能和知识获取功能，它可以用简单的英语和用户对话，回答用户提出的问题，还可以学习专家的知识。

PROSPECTOR系统是一个地质勘探专家系统，用于寻找矿藏，一开始它装入了三种矿藏的知识，到1981年已拥有15种矿藏的知识。该系统的特点是很好地协调了多个专家的多种矿藏的知识模型。它已存入了20多名第一流的经济地质专家的知识。该系统曾发现了美国华

盛顿州的一处钼矿，据说其开采价值超过了一亿美元。

AM系统是一个机器学习系统，用于模拟人类归纳推理、抽象概念。它可以由一组有限集合论方面的简单概念抽象出自然数、因子、质数、乘法等概念，并发现了一些定理和猜想，如质数分解唯一性定理，哥德巴赫猜想等等。此外，还归纳出了一些人类还不曾发现的数学概念。

XCON系统是设计计算机总体配置的系统，也称为R1系统。它根据用户的需要，配置DEC VAX计算机系统，它可以发现用户要求中的错误之处，配置计算机系统的各个部件，规划各部件在地板上的位置等。这个系统每年可为数字设备公司(DEC)节省二千万美元的开支。

5. 专家系统的建造

专家系统主要由知识库和推理机构成，知识库用于存放专家知识，推理机根据知识库中的知识进行推理，作出决策。因此，建立专家系统的工作量主要就集中在这两个方面。

建立知识库的过程包括获取领域知识，将其按专家系统要求的形式构成知识库，并在运行过程中不断调整、充实知识库，使之达到实用的程度。这一过程称为知识获取，是研制专家系统最为困难的阶段，通常称为建造专家系统的瓶颈问题。为解决这一问题，已建立了一些知识获取工具，但还未达到普遍应用的程度。这个问题是知识工程研究的重要课题。

建造推理机的过程就是程序设计的过程。为了加速这一过程，已经出现了象LISP、PROLOG那样的人工智能程序设计语言，用这些语言设计专家系统比用类似C、PASCAL那样的通用程序设计语言更为简洁、方便。另一种途径是利用专家系统构造工具，象EMYCIN、EXPERT等骨架系统。利用骨架系统可以摆脱繁琐的程序设计，把注意力完全集中于知识库的建造，但是骨架系统通常具有通用性差的弊病，难以应用到所有的专家系统。一个折衷的方法是采用通用的知识表示语言，如ROSIE、OPS-5等。它们比骨架系统的限制少，适应面更广，但应用过程比骨架系统复杂。综上所述，对于选择软件工具问题，建议采取如下方法：

(1) 寻找适用的骨架系统，如果有适用的骨架系统，可以最快捷地实现一个高质量的专家系统。如果没有合适的工具可用，再考虑如下方法。

(2) 选用通用的知识表示语言，或者LISP与PROLOG类型的人工智能程序设计语言，可以较迅速地建立自己设计的专家系统。

(3) 选用通用的程序设计语言如C语言，虽然工作量最大，但系统性能好，功能强，并易于移植。

本书从人工智能原理开始，为以各种途径建立专家系统的读者提供充实的材料。

第二章　人工智能程序设计语言

在人工智能领域，最具特色的程序设计语言大体包括三种：以LISP为代表的函数型语言，以PROLOG为代表的逻辑型语言，和以Smalltalk为代表的面向对象的语言。目前，采用LISP语言和PROLOG语言进行人工智能程序设计的比较多，因此本章对这两种语言进行具体的介绍。采用面向对象的程序设计方法是当前的一个趋势，因此本章也将对面向对象的程序设计的原理进行介绍。

2.1　LISP语言

2.1.1　LISP语言概述

LISP语言是1959年由McCarthy领导的MIT的人工智能小组创立的，在人工智能的程序设计中获得了广泛的应用。

LISP语言有多种版本，包括Mac Lisp、Zeta Lisp及Inter Lisp等，1983年在这些版本的基础上出现了一种新型的LISP语言Common Lisp。Gold Hill Computer公司在IBM-PC及其兼容机上实现了Common Lisp，它和Common Lisp的核心部分兼容，称为Golden Common Lisp，简称GCLISP。本节就是介绍GCLISP。

1．LISP语言的特点

LISP语言是一种符号处理语言，也可以说是一种表处理语言。LISP就是List processing language的缩写。为什么LISP语言特别适合于人工智能呢？因为人工智能就是设法用计算机来模拟人的思维过程，而人的思维过程往往可以用语言来描述，而语言可以用符号来表示，因此LISP这种符号处理语言就特别适合于人工智能。具体地讲，LISP语言有如下特点：

(1) LISP语言是函数型语言，它的一切功能都由函数实现。因此，执行LISP程序主要就是执行一个函数，这个函数再调用其它函数。用LISP语言进行程序设计就是定义函数。

(2) LISP语言的函数和数据的形式是一样的，都是S-表达式一种形式。这样会给程序设计带来很大的方便。

(3) LISP语言中程序的运行就是求值。LISP的函数除完成一定的功能外，每个函数都有一个值，因此执行一个函数就可以理解成对函数求值。函数所完成的功能可以看成是它的副作用，在对函数求值的过程中实现函数的功能。

(4) LISP语言的一个主要控制结构就是递归。递归的使用，使程序设计简单易懂。

2．LISP的基本数据类型

LISP语言的程序和数据采用同一个数据类型S-表达式(Symbolic expression)，在Common Lisp中也称为对象(Object)。S-表达式由原子和表两种类型组成。

(1) 原子(Atom)　原子可分为文字原子、数原子和串原子。

文字原子也称为符号(Symbol)，它是由字母开头的字母数字串，可用作变量名、函数名、特性名等，如a，ab2等。

数原子也称为数(Number)。数原子又分为整数(Integer)和浮点数(Float)。字符(Character)是整数的子类，可表示为#\字符名。如字母A表示为#\A，在机内存储它的ASCII码65。

串原子也称为串(string)，是由双引号"开头和结尾的字符串。

(2) 表(list) 表由括号及其中的若干元素所组成，元素可为任何数据类型。如(a b c)及((a) (bc) ((( ))))都是表。

表由空表和非空表(Cons)组成，空表就是( )，也可表示为nil，它既是表，也是文字原子。

LISP的基本数据类型及其相互关系如图2.1所示。另外还有一些其它的数据类型如数组(Array)、函数(Function)、流(Stream)等，遇到时再加以说明。

(3) 表在机内的存储 表在机内以链表的形式存储，链表是指一组内存单元，每个内存单元分成两部分，每个部分有一个地址，称为指针。右边的指针把各个单元串在一起，左边的指针指向表内各元素。

例如，若文字原子example的值是表(This is a list)，表中共有4个元素，故链表由4个单元构成。每个单元的右边指针指向下一个单元，最后一个单元的右边指针为空(nil)，每个单元的左边指针指向各元素如图2.2所示。

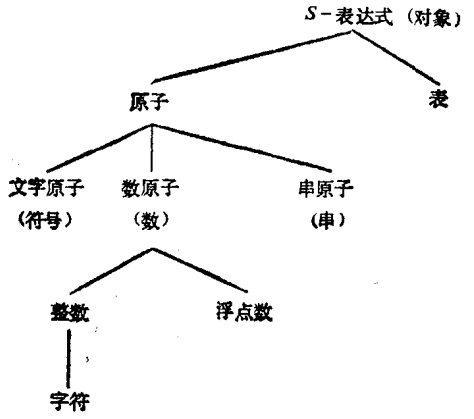

图2.1 LISP的基本数据类型

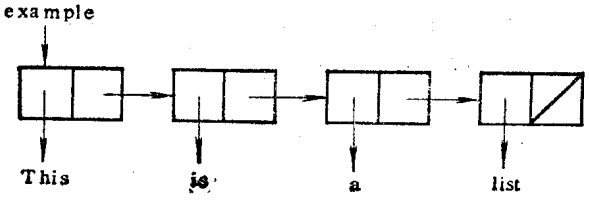

图2.2 表(This is a list)的链表形式

若文字原子L的值为表(4 (3 7) a (5) b c)，表中有6个元素组成，因此链表应包含6个单元。但第2、4个元素又是表，它们又应分别由2个和1个单元组成，因此链表应如图2.3所示。

3. 读入-求值-显示循环

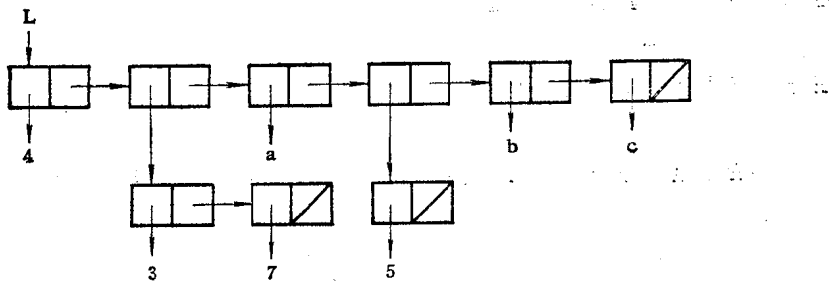

图2.3 表(4 (3 7) a (5) b c)的链表结构

GCLISP是一个解释执行的系统。进入系统后即进入了收听程序(Listener)，完成读入-求值-显示循环。系统的提示符为*。循环的操作为：由用户打入一个表达式，系统将其读入，对其求值，并将值显示在屏幕上。例如在提示符*后打入求和函数(+ 1 2 3)，则显示

```
*(+ 1 2 3)
6
*
```

系统显示了和6后，再显示提示符*，等待下一个读入-求值-显示循环。

退出系统时打入(exit)

4. 函数表示中的记号约定

为函数表示的简洁起见，特作如下约定：

&rest：表示其后的自变量可以有0个或多个

&key：表示其后为可选的关键词

&optional：表示其后为可选的自变量

form：表示可求值表达式

form-result：表示form的值

last-form-result：表示最后一个form的值

[…]：表示括号内为可选项

{…}*：表示括弧内的项可出现0次，1次或多次

|：表示分割两个可选项

⇒：表示表达式的值

大写字母开头的自变量：表示不对其求值

小写字母开头的自变量：表示对其求值

2.1.2 LISP的基本功能

1. 算术函数

(1) (+ &rest numbers)⇒ sum

+表示函数名，本函数自变量的数目可以有0个或多个，number表示自变量为数。函数的功能是，将各自变量求值，各自变量的值必为数，再将这些数相加，其和作为本函数的值。例如

(+ 1 2 3)⇒6

(+)⇒0

若文字原子a的值为4，b的值为3，则

(+ a b)⇒7

注意：文字原子的值应预先赋予，而数的值就是该数本身。

(2) (- number &rest more-numbers)⇒difference

本函数至少应有一个自变量，其功能是实现多个数相减。例如

(- 1 2)⇒-1

(- 1)⇒-1

(- 10 5 2 5)⇒-2

(- 10 5 (+ 2 5))⇒-2

上述最后一个函数中，第三个自变量是一个+函数。对-函数的第三个自变量的求值就是对+函数求值，将其值作为-函数第三个自变量的值。

(3) (* &rest numbers)⇒product

功能是实现多个数相乘。例如

(* 2 3)⇒6

(*)⇒1

(4) (/ number &rest more-numbers)⇒quotient

功能是实现多个数相除。例如

(/ 15 5 2)⇒1.5

(/ 2)⇒0.5

(5) (1+ number)⇒successor

功能是将自变量的值加1，等价于(+ number 1)。

(6) (1- number)⇒predecessor

功能是将自变量的值减1，等价于(- number 1)。

2. 赋值与求值函数

(1) (setq {Symbol form}*)⇒last-form-result

本函数的自变量是成对出现的。奇数个自变量为文字原子，不求值，偶数个自变量为可求值表达式。函数的功能是对偶数个自变量求值，将值赋给它前面的文字原子。赋值从左至右依次进行。例如：

(setq a 4 b 3)⇒3

(setq a 5 b (+ a 4))⇒9

此时若对文字原子a、b求值，则有

a⇒5

b⇒9

系统还规定

(setq)⇒nil

(2) (psetq {Symbol form}*)⇒nil

功能与setq类似，但为并行赋值。

(3) (set symbol form)⇒form-result

本函数对两个自变量都要求值。功能是将form的值赋予symbol的值，其中symbol的值应为文字原子。例如设文字原子a的值为b，则有

(set a 7)⇒7

b⇒7

(4) (let ({Var|(Var value)}*) {form}*)⇒last-form-result

本函数的自变量分为两部分，前面为局部变量表，后面为可求值表达式。功能是先对局部变量Var赋值value，仅有Var时赋值nil，并行赋值。再依次对form求值，返回最后一个form的值。

在函数的求值过程中，Var为局部变量，赋予它的值称为临时值，函数执行结束后其值