



知识处理与专家系统

何新贵 编 著

国防工业出版社

知识处理与专家系统

何新贵 编著

国防 業 出 版 社

内 容 简 介

本书把近年来互相独立发展的各项知识处理的有关理论与技术,系统地形成为一门学科——知识处理学。全书共分三部分。第一部分知识处理学基础,共八章。介绍了知识处理学的形成与发展、研究对象、研究方法和环境。定义了各种基本概念并重点地研究了知识的表示、知识的获取、知识的运用与处理等核心问题。第二部分知识处理语言,共五章,分别介绍了五种典型的和最有影响的知识处理语言:函数型语言Lisp,逻辑型语言Prolog,面向对象的语言Smalltalk,框架表示语言FRL,以及产生式语言OPS5,对各种语言的语法、语义及特点等作了较详细的叙述与评价。第三部分专家系统及其开发工具与环境,共六章,介绍了专家系统的一般原理、构造方法、开发过程以及工具与环境等,对一种特殊的专家系统——智能决策支持系统作了详细介绍。本书最后还展望了新一代专家系统技术以及智能计算机。本书可作知识工程师,专家系统开发人员,软件专业大学教师及理论工作者的工具与参考书,亦可作软件专业研究生和高年级本科生的教材使用。

2P50/12

知识处理与专家系统

何新贵 编著

*

国防工业出版社出版、发行

(北京市海淀区紫竹院南路23号)

(邮政编码100044)

新华书店经售

国防工业出版社印刷厂印装

*

850×1168毫米 1/32开本 20.25印张 500千字

1990年9月第一版 1990年9月北京第一次印刷 印数:0001—3000册

ISBN 7-118-00754-4/TP·98 定价:15.00元

前 言

70年代中期一些称为“专家系统”的计算机程序的研制成功打破了人工智能领域的沉闷空气，使一度处于低谷的人工智能研究重新获得了生机。随后在世界范围内掀起了一个研制各种领域的专家系统的热潮，使得关于知识处理的研究受到越来越多学者们的重视。人工智能研究的这种转机首先应归功于费根鲍姆（Edward A. Feigenbaum）及其研究小组。费根鲍姆等人的主要功劳其实就在于他们注意到了人类专家们（而不是万能博士们）解决其领域问题时的方式和方法，指出了专门领域知识在解决各种领域问题中的重要作用。为了解决特定领域的一个具体问题，除了需要一些公共的常识，例如推理方法和数学知识等之外，更需要应用大量与所解问题领域密切相关的知识，即所谓的“领域知识”。特别在科学技术日新月异的今天，专业分工越来越细，在人类中什么都会万事通晓的“万能博士”已经不复存在，更何况一架由人类发明和制造的计算机，怎能期望它来解决一切问题呢？历史经验已经表明，目前要在现有的计算机上实现一个万能的通用解题程序是不可能的。专家系统正是在批判了原来人工智能的一些学者们那种好高骛远的学术思想之后才获得成功的。

从分析人类专家们解题的过程中我们可以发现，在解题中除了必须运用大量的领域专门知识之外，还有另一个明显的特点，即采用所谓“启发式”的解题方法（或称试探性的解题方法）。为了解一个问题，特别是一些问题本身就很难用严格的数学方法描述的问题，往往不可能借助一种预先设计好的固定程式或算法来解决它们，而必须采用一种不确定的试探性解题方法。在用这种方法解题时，专家们要在不断地试探中不断获得“反馈”，并

在不断获得“反馈”的基础上不断进行评估，然后根据评估结果启发提示下一步的解题策略和方向等。这种启发式的解题方法已在实践中表明是行之有效的。

在专家们解题时，“演绎的方法”或“形式逻辑的方法”自然是必需的。但是“演绎”只能发现其实已经客观存在着的，但仍蕴含着的知识，或叫做只能证明知识的正确性，而并不能创立新知识和深化旧知识。所以人类专家解题的第三个特点就是在解题中除了运用“演绎方法”之外，必须求助于“归纳的方法”和“抽象的方法”。因为只有运用归纳和抽象才能创立新概念，推出新知识，并使知识深化。例如，人们可从千千万万个具体的母亲都爱她们的孩子的千万件实例出发，抽象出“抽象的母亲”和“抽象的孩子”的概念，并用一种“抽象的母爱”把这两个抽象的概念联系起来，从而归纳出“母亲爱孩子”这样一个抽象命题，这种抽象了的认识显然比对千万个具体实例中的任何一个的认识大大深化了一步。

专家解题的第四个特点是他们必须处理问题的模糊性、不确定性和不完全性。因为专家们要解决的是现实中的问题，而现实世界就是充满模糊性、不确定性和不完全性的。这确定了解决这些问题的方式和方法也必须是模糊的和不确定的，并能处理不完全性。所以在知识处理研究中必须十分重视关于事物的模糊性、不确定性和不完全性的描述和处理的研究，特别是模糊逻辑、模糊推理和模糊匹配等的研究。

解题的过程是不断运用知识的过程。首先运用已有的知识进行启发式的解题，并在解题中不断修正旧知识获取新知识，从而丰富和深化已有的知识，然后再在一个更高的层次上运用这些知识求解问题，如此循环往复，螺旋式上升，直到把问题解决为止。这种观点与毛泽东同志在《实践论》中指出的“实践，认识，再实践，再认识，这种形式，循环往复以至无穷”的论点是完全一致的。由上面的分析可见，这种解题的过程中，人们所运用和操作的对象主要是各种知识（当然也包括各种有关的数据），因

此也就是一个知识处理的过程。

其实，一个具体专家系统的建造和完善的过程也就是一个不断地重复“获取（或抽取）知识→表示知识→运用知识→进一步获取或修正知识”这个循环的过程。要让计算机来模仿人类解决问题必然需要知识，知识从何而来？需要从人类专家的脑中获取，或从积累的书本中抽取，因而很自然，研究各种获取知识的方法和途径成了知识处理中第一个需要解决的问题。然而，知识是一种抽象的东西，要把它告诉计算机或在其间进行传递（或传授他人），还必须把它以某种形式逻辑地表示出来，并最终编码到计算机中去。这就是知识处理中要研究的“知识的表示问题”。正如记数法是数据处理的基础一样，知识的表示也是知识处理的基础。因为只当有了知识的表示以后才谈得上对它进行处理。而且不同的知识需要用不同的形式和方法来表示。它既能表示事物间结构关系的静态知识，又能表示如何对事物进行各种处理的动态知识；它既要能表示各种各样的客观存在着的事实，又要能表示各种客观规律和处理规则；它既要能表示各种精确的、确定的和完全的知识，还应能表示更加复杂的、模糊的、不确定的和不完全的知识。因此，一个问题是否能以合适的知识表示方法往往成为知识处理（解题）成败的关键。综上所述，“知识的获取”，“知识的表示”以及“知识的运用”也就成了知识处理学的三大要素或主要研究内容。

此外，正如自然语言既是表示人类知识的工具，也是描述如何处理和运用知识的手段一样，要在计算机上进行知识处理同样也需要有一种语言来完成这些功能。传统的程序设计语言，例如FORTRAN和COBOL等，由于它们主要是为表示和处理数据而设计的，一般都不适用来表示复杂的知识和描述运用知识的规则和方法。因此，在知识处理中，知识处理语言的研究和设计成了一个至关重要的课题。知识处理语言应该是上述知识处理三要素研究的结晶和具体体现。既要利用语言以表示知识，又要利用语言以描述对知识的处理与运用，还要利用语言实现与各种知识源

(专家或其它知识宝库)进行交互和处理以修正旧知识,获取新知识和创立新概念。这样,知识处理语言把知识的表示、处理和获取联系了起来统一于一身,由此可见一种合适的程序设计语言作为知识处理基础工具的至关重要性。为了适应不同层次用户的使用,知识处理还应有更高级的工具,或对一般用户更友好的工具,乃至一个完备的工具箱和完全的知识处理环境。

费根鲍姆等人把建造专家系统的技术称为“知识工程(Knowledge Engineering)”,企图来说明这是与机械工程和土木工程等一样的一门工程学。但是应该说事实并不是这样,至今它仍还不是一门工程。众所周知,一门工程首先应有其理论基础,应有一套成熟的设计方法与规范,应有一套严格的工序和工艺要求,以及各种工具与设备等等。可是对这门新兴的技术而言,可以说这些都还不具备或不完备,作为其理论基础的知识处理学也还很很成熟,尚未最后形成一个独立的学科。很多问题还仅仅是孤立的研究,尚未系统化和理论化。所以作者认为,如今称它为“知识工程”仍然为时过早。

写本书的一个主要目的就是想将在将知识处理现有成果的系统化与理论化方面做一些工作,力图把它形成为一门独立的学科。所以,这本书最合适的书名也许应该叫《知识处理学》,但是为了强调它在专家系统以及其它基于知识的系统中的应用才取了现在的名字。书中包括了作者近年来在这方面的工作,特别是在模糊数据库、模糊知识表示以及模糊逻辑研究中取得的成果,综合了国内外许多学者们最新的有关研究成果和这门技术未来的发展趋势。所以它从内容上讲是很新的。它的基本内容曾在北京软件研究生院作为专业课程《知识处理与专家系统》进行讲授,后来几经充实和修改才成了本书。

全书共分三部分。第一部分是知识处理学基础,较系统地介绍了知识处理学的各种基本概念和有关理论,较详细地论述了知识处理的三大要素。第二部分收进了当前最著名的五种可供知识处理用的程序设计语言,它们都是根据第一部分中介绍的基本理

论设计的，如今已经在许多实际应用中起着重要作用。第三部分是知识处理学在专家系统和其它基于知识的系统中的应用，介绍了专家系统及其开发工具的原理、结构以及建造过程。书的最后一章展望了新一代基于知识的系统，论述了专家系统的未来发展趋势。

作者希望把本书写成一本《知识处理学》，但是一方面由于知识处理技术本身发展和变化迅速，另方面其中不少问题仍然研究不够充分和深入，再加上作者水平所限，本书能否起到期望的作用尚很难说。此外，由于书的内容广泛，文字较多，疏漏与错误可能在所难免，十分欢迎同行们给予批评指正，以求不断完善。

最后还应指出，本书是与郭荣和严冬两位合作的结果，没有他们的密切合作，本书的完成是不可能的。本书在出版和印刷过程中，得到科海培训中心的大力支持，作者对此表示感谢。

目 录

第一部分 知识处理学基础

第一章 从数据处理到知识处理	(2)
1.1 知识处理学的形成与发展	(2)
1.2 知识处理学的研究对象	(6)
1.3 知识处理学的研究方法	(11)
1.4 知识处理的工具与环境	(13)
1.5 结束语	(14)
第二章 知识与智能	(16)
2.1 基本概念	(16)
2.1.1 数据	(16)
2.1.2 信息	(17)
2.1.3 知识	(18)
2.1.4 智力与智能	(19)
2.1.5 知识库和知识处理器	(20)
2.2 知识的属性	(20)
2.2.1 真理性	(20)
2.2.2 相对性	(21)
2.2.3 不完全性	(21)
2.2.4 模糊性与不精确性	(22)
2.2.5 可表示性	(22)
2.2.6 可存储性 (可记忆性)	(23)
2.2.7 可传递性	(23)
2.2.8 可处理性	(23)
2.2.9 相容性 (无矛盾性)	(23)
2.3 知识的分类	(24)
2.3.1 事实	(24)

2.3.2	规则	(24)
2.3.3	规律	(24)
2.3.4	推理方法	(25)
2.3.5	解题方法或算法	(26)
2.3.6	理论与知识空间	(27)
2.3.7	通用知识与领域专门知识	(27)
2.4	智能与大脑	(27)
2.5	人工智能	(29)
2.6	智能计算机	(30)
第三章	知识的表示	(32)
3.1	引言	(32)
3.2	一阶谓词逻辑表示	(34)
3.2.1	一阶谓词逻辑的定义	(34)
3.2.2	一阶谓词的形式描述及语义	(36)
3.3	关系表示(或特征表表示)	(37)
3.3.1	关系与谓词间的对应关系	(37)
3.3.2	用关系表示规则及元组函数	(38)
3.4	产生式表示(或规则表示)	(40)
3.4.1	产生式的定义	(40)
3.4.2	产生式的与/或树表示	(41)
3.4.3	产生式的形式描述及语义	(42)
3.4.4	一个例子	(43)
3.5	语义网络表示	(44)
3.5.1	语义网络的定义	(44)
3.5.2	语义网络的形式描述及语义	(46)
3.5.3	常用的语义联系	(47)
3.6	框架表示	(51)
3.6.1	框架的定义	(51)
3.6.2	框架的形式描述及语义	(57)
3.6.3	框架系统的预定义槽	(59)
3.7	过程表示	(60)
3.7.1	过程知识	(61)

3.7.2	过程语言	(62)
3.8	脚本表示	(63)
3.8.1	表示行为用的原语	(63)
3.8.2	脚本	(64)
3.9	面向对象的知识表示	(66)
3.9.1	面向对象的方法学	(66)
3.9.2	对象类的定义方式	(67)
3.9.3	对象表达式	(68)
3.9.4	原子对象类	(70)
3.9.5	结构对象类	(71)
3.9.6	结束语	(72)
3.10	知识表达式——一种统一的知识表示模式	(73)
3.10.1	知识原子	(73)
3.10.2	知识因子	(74)
3.10.3	更复杂的知识因子	(75)
3.10.4	知识项	(77)
3.10.5	知识表达式	(79)
3.10.6	知识表达式的BNF表示	(81)
3.10.7	与其它知识表示模式的关系	(82)
3.10.8	应用举例	(83)
3.11	混合的知识表示模式	(86)
3.11.1	混合表示的意义	(86)
3.11.2	一些混合表示的方案	(87)
3.12	选择或设计知识表示模式的准则	(88)
第四章	模糊数据库与模糊知识表示	(90)
4.1	世界的模糊性与模糊知识	(90)
4.2	模糊数据	(91)
4.2.1	模糊性与随机性	(92)
4.2.2	模糊数的表示	(92)
4.2.3	模糊数间的语义距离[9]	(94)
4.2.4	模糊数上的运算	(96)
4.3	模糊数据库的数据模型	(98)

4.3.1	模糊关系数据模型 (FRDM) ^[8]	(99)
4.3.2	元组属性具有模糊值的关系数据库模型 (FVRDM) ^[8]	(102)
4.3.3	元组属性具有不定值的关系数据库模型 (UVRDM)	(103)
4.3.4	元组属性具有模糊值的模糊关系数据库模型 (FVFRDM)	(104)
4.3.5	属性具有加权模糊值的模糊关系数据库模型 (WFVFRDM) ^[9]	(105)
4.3.6	模糊实体-联系模型 (FERDM) ^[10]	(112)
4.3.7	面向对象的模糊数据库模型 (OOFDM) ^[13]	(117)
4.4	模糊知识表示	(124)
4.4.1	现实世界中的知识	(124)
4.4.2	模糊谓词	(125)
4.4.3	模糊规则 (或模糊产生式)	(125)
4.4.4	模糊框架	(127)
4.4.5	模糊语义网络	(127)
4.4.6	其他模糊表示方法	(128)
4.5	结束语	(128)
第五章	知识的获取	(130)
5.1	知识获取的任务	(130)
5.2	知识获取方法的分类	(131)
5.3	机械照搬式知识获取	(133)
5.4	条件反射式知识获取	(133)
5.5	教学式 (或传授式) 知识获取	(134)
5.5.1	获取方式	(134)
5.5.2	知识工程师的职责与任务	(135)
5.5.3	知识编译器	(135)
5.6	演绎式知识获取	(136)
5.7	归纳式知识获取	(137)
5.7.1	枚举归纳知识获取	(137)
5.7.2	类属归纳知识获取	(138)

5.8	类比归纳知识获取	(141)
5.8.1	外延归纳知识获取	(141)
5.8.2	联想归纳知识获取	(142)
5.8.3	类比归纳知识获取	(143)
5.9	猜想(假设)——验证式知识获取	(144)
5.10	反馈修正式知识获取	(145)
5.11	逆推理归纳知识获取	(145)
5.12	消除归纳知识获取	(147)
5.13	概念的形成	(149)
5.14	知识获取的步骤	(151)
第六章	知识的运用及处理	(155)
6.1	引言	(155)
6.2	知识运用的种类	(155)
6.3	推理	(156)
6.4	搜索	(158)
6.4.1	搜索问题	(158)
6.4.2	搜索方向	(167)
6.4.3	搜索路径	(168)
6.4.4	搜索方法	(168)
6.5	匹配与替换	(187)
6.5.1	匹配与变量替换的进行方式(合一过程)	(188)
6.5.2	冲突的解决办法	(190)
6.6	知识的组织、管理和维护	(191)
6.6.1	知识库的组织	(192)
6.6.2	知识库的管理与维护	(214)
6.7	知识处理语言	(216)
第七章	推理的逻辑基础	(217)
7.1	形式演绎逻辑	(217)
7.2	非单调演绎逻辑	(220)
7.3	三值逻辑	(225)
7.3.1	Kleene的强三值逻辑	(225)

7.3.2	Lukasiewicz逻辑	(226)
7.3.3	Bochvar逻辑	(227)
7.3.4	三值代数——计算三值逻辑	(229)
7.4	多值逻辑	(231)
7.5	可信度逻辑及概率逻辑	(232)
7.6	模态逻辑	(234)
7.7	时序逻辑	(235)
7.8	模糊逻辑	(238)
7.8.1	Zadeh模糊逻辑	(239)
7.8.2	可能的改进	(240)
7.9	加权模糊逻辑 [15]	(241)
第八章	不精确知识与不完全知识的运用及处理	(244)
8.1	引言	(244)
8.2	不精确推理与知识不完全推理	(244)
8.2.1	基于多值逻辑的不精确推理	(244)
8.2.2	专家系统MYCIN的不精确推理	(245)
8.2.3	统计推理	(249)
8.2.4	模糊变换、综合评判和模糊推理	(252)
8.2.5	加权逻辑推理	(255)
8.2.6	假设——验证式推理	(259)
8.2.7	定性代数推理	(259)
8.3	模糊匹配	(264)
8.4	模糊检索与信息不完全检索	(265)

第二部分 知识处理语言

第九章	函数型程序设计语言Lisp	(269)
9.1	引言	(269)
9.2	原子	(269)
9.3	符号表达式和表	(270)
9.4	基本函数	(273)

9.4.1	函数CAR	(274)
9.4.2	函数CDR	(274)
9.4.3	函数CAR与CDR的复合	(275)
9.4.4	函数CONS	(276)
9.4.5	函数QUOTE	(276)
9.4.6	谓词ATOM	(277)
9.4.7	谓词NULL	(277)
9.4.8	谓词EQ	(277)
9.5	赋值与求值函数	(278)
9.5.1	赋值函数SET	(278)
9.5.2	赋值函数SETQ	(278)
9.5.3	求值函数EVAL	(279)
9.6	算术与逻辑运算函数	(280)
9.6.1	求和函数PLUS	(280)
9.6.2	求差函数DIFFERENCE	(281)
9.6.3	求积函数TIMES	(281)
9.6.4	求商函数QUOTIENT	(281)
9.6.5	求余函数REMAINDER	(282)
9.6.6	除函数DIVIDE	(282)
9.6.7	其他算术运算函数	(282)
9.6.8	“与”函数AND	(283)
9.6.9	“或”函数OR	(283)
9.6.10	“非”函数NOT	(283)
9.6.11	其他逻辑谓词函数	(284)
9.7	表处理函数及谓词	(284)
9.7.1	函数LIST	(284)
9.7.2	函数APPEND	(285)
9.7.3	求表长函数LENGTH	(286)
9.7.4	成员谓词MEMBER	(286)
9.7.5	相等谓词EQUAL	(286)
9.8	条件分支函数COND	(287)
9.9	函数定义及递归	(289)

9.9.1	函数定义函数DEFINE (或DEFFUN或DEF).....	(289)
9.9.2	递归.....	(290)
9.9.3	函数MAPCAR与APPLY	(291)
9.9.4	函数的LAMBDA定义及求值.....	(292)
9.9.5	LABEL表达式	(294)
9.10	输入/输出函数.....	(295)
9.10.1	函数READ.....	(295)
9.10.2	函数PRINT	(296)
9.11	函数PROG	(296)
9.12	Lisp程序的结构	(298)
9.13	Lisp程序举例	(299)
9.13.1	皇后问题	(299)
9.13.2	符号微分	(301)
9.14	结束语	(302)
第十章	逻辑程序设计语言Prolog	(304)
10.1	一阶谓词演算与Horn子句	(304)
10.1.1	一阶谓词演算	(304)
10.1.2	Horn子句	(307)
10.2	一阶谓词公式到Horn子句的转换.....	(308)
10.3	语言的语法及语义	(311)
10.4	程序的与/或树表示.....	(313)
10.5	程序的执行与控制机构	(316)
10.6	执行控制机构的图形解释	(318)
10.7	程序执行中变量的约束与解脱	(323)
10.8	Prolog中的表结构	(324)
10.9	Prolog的系统谓词	(325)
10.9.1	比较谓词	(325)
10.9.2	算术运算及is谓词	(326)
10.9.3	输入/输出谓词.....	(327)
10.9.4	判别谓词	(329)
10.9.5	控制回溯的谓词	(329)

10.9.6 其他常用的系统谓词	(333)
10.10 Prolog程序举例	(334)
10.10.1 求最大公约数	(334)
10.10.2 汉诺塔问题	(334)
10.11 结束语	(336)
第十一章 面向对象的程序设计语言Smalltalk	(338)
11.1 引言	(338)
11.2 Smalltalk的基本思想及概述	(338)
11.3 字值与标识符	(342)
11.4 消息	(343)
11.5 表达式	(344)
11.6 语句	(345)
11.7 程序块	(346)
11.8 类定义	(347)
11.9 语法图及解释	(350)
11.9.1 类描述	(351)
11.9.2 类描述头部	(351)
11.9.3 冒号变量	(351)
11.9.4 实例变量	(351)
11.9.5 协议	(352)
11.9.6 方法	(352)
11.9.7 方法模式	(352)
11.9.8 临时变量	(353)
11.9.9 语句	(353)
11.9.10 表达式	(353)
11.9.11 级联表达式	(353)
11.9.12 简单表达式	(354)
11.9.13 二元式	(354)
11.9.14 一元式	(354)
11.9.15 初等量	(354)
11.9.16 继续式	(355)
11.9.17 程序块	(355)