

CONSTRUCTING EXPERT SYSTEMS

专家系统设计

吴信东 编著

中国科学技术大学出版社

TP18
11

专家系统设计

吴信东 编著

中国科学技术大学出版社

1990·合肥

2086/3506

内 容 简 介

本书以作者近年来发表的有关科研论文和实际实现的知识工程工具 KEshell 为主要素材, 详细论述专家系统和专家系统开发工具的各个模块的设计技术。内容共分八章: 绪论, 智能程序设计与 PROLOG 语言, 知识的内部表示, 自动知识获取程序的设计, 推理机制的设计, 不精确推理实现的问题和求解, 一个跟踪解释程序的设计, 知识库管理程序。附录中给出了 Turbo-PROLOG 的内部谓词。

本书是一部学术和实用并重的技术书籍。书中各主要章节都给出了具有良好设计风格的程序, 可直接移植到 IBM-PC 系列机或兼容机上实现。

本书可供从事人工智能、知识工程、专家系统领域研究和开发的工程技术人员、大专院校教师、高年级大学生和研究生学习参考。

专 家 系 统 设 计

吴信东 编著

责任编辑: 黄 德 封面设计: 王瑞荣

中国科学技术大学出版社出版

(安徽省合肥市金寨路 96 号)

中国科学技术大学印刷厂印刷

安徽省新华书店发行

开本: 850×1168/32 印张: 6.5 字数: 169 千

1990 年 4 月第 1 版 1990 年 4 月第 1 次印刷

印数: 1—5000 册

ISBN7-312-00070-3/TP·17 定价: 4.00 元

序

八十年代出现的专家系统技术高潮将人工智能从实验室研究推向实用化和商品化，并渗透到社会的各个领域，形成了应用人工智能领域中最活跃的一个分支。人们不仅已经意识到应用专家系统技术构成的计算机程序能解决需要专家知识或专家建议才能解决的问题，更重要的，可望利用专家系统技术能促使非程序人员开发强有力的应用程序，从而提供缓解软件人员供不应求趋势的出路。

专家系统不应该当做一种产品来描述，而是整个概念、技术和问题求解方法上的更新。它的出现和发展将较之七十年代出现的微机高潮对计算机的发展有着更为广泛而深入的意义，因此有人称之为第二次计算机革命。美国人工智能权威专家 Winston 认为：专家系统不仅以其单独的系统出现，而且将小型专家系统象葡萄干一样散布在大型系统中使其具有更重大的价值，这就意味着专家系统技术汇入计算机系统的主流中，将发展成新一代计算机应用系统——智能系统。

80 年代以来，在美国已有 80% 的大公司将专家系统技术应用到许多方面，日本已拥有处于不同阶段的专家系统二千多个，西欧各国专家系统技术的年增长率已平均高达 30% 以上。预计 90 年代，专家系统技术市场将成为各大公司相互竞逐的场所，到 90 年代中期，将出现比 80 年代更为蓬勃的专家系统技术高潮。

从专家系统技术本身来看，一些基本技术，如知识表示、推理与控制、以及知识获取，将随着人工智能的研究发展而继续深化。而从专家系统技术的实用化和商品化的角度来看，为适应社

会发展对规模更大、水平更高的智能系统的需要，正如软件及程序设计逐步形成软件工程和程序设计方法学的发展历程一样，从工程化和方法论的角度来研究建造专家系统的技术，应当是专家系统技术研究的一个很重要的方向。

吴信东同志编著的《专家系统设计》一书，从设计一个实际的专家系统的角度来阐述有关技术，这对推动专家系统技术在我国的应用和发展，将起积极作用。

合肥工业大学人工智能应用研究室 张奠成

1990年1月

前 言

专家系统是一门还在发展之中的高技术，近些年来在我国受到了广泛的关注并得到了迅速的发展。国家自然科学基金和国家“七五”科技攻关项目中，有不少项研究课题面向不同领域的专家系统。许多高校在研究生和本科生课程中列入了专家系统或相应的课程，把专家系统列为研究生课题方向的硕士点和博士点也在不断增多。

专家系统的研究可以用两种类型进行归类，一类是基础研究，一类是应用研究。基础研究主要指计算机科学技术和自动化有关专业人员所进行的专家系统技术本身的研究，包括专家系统中正在探索的有关理论问题和由于应用引起的一些新问题的研究，也包括一些解决一类应用领域中有关共性问题的专家系统开发工具（或称为知识工程工具）的研究。应用研究主要指为解决实际领域的具体问题而进行的专门专家系统的开发或设计。当然，在一些实际的科研项目中，往往既包括基础技术研究又包括应用技术研究，它们并不能截然分开，只是不同的项目有不同的侧重而已。

就目前专家系统的发展来看，虽然有了很大进展，但还有许多实质性问题有待解决。国外人工智能、专家系统的怀疑论者们还不时在提出“何以专家系统失败”、“人工智能将何处去”之类的问题，否定论者提出的问题则更为尖锐。作者认为：人工智能、专家系统的技术现状确实并非象有些推崇者们所描绘的那样令人欢欣鼓舞。我们知道，启发式问题求解和逻辑推理（或称符号推理）是人工智能研究的两个核心，作为人工智能一个应用分支的专家系统研究当然也不例外。围绕着这两个核心问题导致了人工

智能的三大基本技术：知识（尤其是启发式知识）的表示、获取和推理。这些技术对解决那些可以用逻辑结构描述的非定规类问题（如定理证明及有关解释、诊断、规划、设计型问题等），确实实现了用基于算法设计的传统程序很难实现甚至无法实现的求解。但是，更多的实际问题，尤其是专家系统所面临的工程应用型问题，只用逻辑结构和启发式方法进行求解又显得极为脆弱。到目前为止，真正能够投放市场实现了商品化的实用知识工程软件并不太多，经营专家系统的公司还没有象原来预料的那样在 80 年代以每周就有若干家开张的速度发展。E.A.Feigenbaum 教授在 AAAI'88 的特邀报告上说，世界范围内真正实用的专家系统在 1988 年还不足 2000 个。作者认为：导致这种情况的重要原因之一，是人工智能、专家系统研究未能充分重视传统的算法式程序设计方法的结合，未能充分重视数值（包括概率数、模糊数）的有效处理。各种已为人工智能研究所称道的描述性知识表示技术对表达实际问题求解中常常不可缺少的数值计算还显得不够灵活。这是涉及人工智能专家系统基础研究的一项重要课题。

在我国，专家系统研究存在的另一个重要问题是：设计中的重复工作太多。不少单位的专家系统选题是以研究生毕业设计为线索的，常常没有实际科研项目的支撑。研究生往往为了毕业论文的需要，从头设计一个可初步运转起来的系统雏形，当毕业论文写完，设计工作也就结束了，能够提供他人继承的内容常常几乎是空白。非计算机专业的专家系统方向的研究生（也包括其他研究人员）从头设计的专家系统结构常常不清晰，程序体冗长，且没有良好的设计风格（因为没有经过计算机专业训练而单纯靠几十学时的智能程序语言的了解就试图熟练地掌握智能程序设计技巧确实很困难）。导致重复的重要原因之一是没有足够的规范的专家系统软件可以借鉴，从而使得许多研究生把大量的精力集中到了程序编写上。我国近几年出版了为数不少的人工智能、专家系统书籍，但专门论述专家系统设计者不多，能起示范

作用、能提供可资使用的专家系统软件的书籍则更少。

近几年来，作者从事了专家系统的专门研究和教学工作，参加了国家“七五”科技攻关项目中有关专家系统课题和其它一些实际专家系统项目的研制工作。作者申请的一项与专家系统研究有关的自动知识获取项目已得到了国家自然科学基金的批准和资助。本书围绕前述的两个问题，以作者实际研制的知识工程工具KEshell为示例，主要介绍作者近年来在专家系统研究方面所取得的一些粗浅的成果和实践，内容包括专家系统及专家系统开发工具各个模块的设计。本书中的几个重点或称特色可概括如下：

1、学术性。针对人工智能、专家系统中现有的各种描述性知识表示的局限性，本书采用由我国学者熊范纶、周金铭最先提出，经过作者改进、扩充和进一步形式化的“规则架+规则体”表示方法，给出基于这种表示的内部编码、知识获取、基于知识的推理、不精确推理、追踪解释及知识库管理的详细论述。这些论述材料许多都直接取材于作者近几年发表的一些中、英文研究论文。

2、实用性。书中各主要章节在介绍技术内容之后都给出了用Turbo-PROLOG实现的程序实例，这些程序具有良好的递归结构，读者可移植到IBM-PC系列机或兼容机上实现。

3、模块性。除了第一章和第二章，其他每章论述一个主题，这个主题除了与第一章中“知识的‘规则架+规则体’表示”有直接联系以外，可以自成体系。每章一般包括现有技术的介绍、作者的研究工作和程序实现三方面内容。读者可以根据需要在相应的章节中寻找、发掘进一步的研究课题。

本书与作者编著的《专家系统技术》（电子工业出版社，1988）一书可以配套使用。读者最好先阅读《专家系统技术》，以对专家系统技术的基本内容能有一个比较系统的了解。但未读过《专家系统技术》也不妨碍本书的阅读，本书以“规则架+规则体”表示为基础，全书自成体系。读过《专家系统技术》的读

者可以跳过第一章的第一节，具备了 PROLOG 程序设计知识和实践的读者可以跳过本书的第二章。

本书的大部分内容作者曾在给合肥工业大学研究生开设的《PROLOG 与专家系统》课程中讲授过。在这次编撰成书的过程中，合肥工业大学多位老师和同行曾提出了许多宝贵意见。书中的有些技术内容是作者与其他老师或同行合作研究的，相关文献已罗列在参考文献中。作者在此向他们一并表示感谢。

书中不足之处恳请读者批评指正。

吴信东

1989 年 10 月于合肥工业大学

目 录

序	张冀成
前 言	(1)
第一章 绪 论	(1)
第一节 专家系统的基本结构	(1)
第二节 专家系统的建立	(4)
第三节 专家系统开发工具	(8)
第四节 知识的“规则架+规则体”表示	(12)
第二章 智能程序设计与 PROLOG 语言	(20)
第一节 逻辑程序设计	(20)
2.1.1 Horn 子句	(21)
2.1.2 消解和定理证明	(22)
2.1.3 并行性	(26)
2.1.3.1 与并行	(26)
2.1.3.2 或并行	(26)
2.1.3.3 流水线并行	(26)
2.1.3.4 搜索并行方式	(27)
2.1.4 标准 PROLOG	(27)
2.1.4.1 语法	(27)
2.1.4.2 语义	(28)
第二节 Turbo-PROLOG 介绍	(29)
2.2.1 Turbo-PROLOG 的三种基本语句	(30)
2.2.2 Turbo-PROLOG 的程序结构	(31)
2.2.3 Turbo-PROLOG 的复合对象— 结构	(33)

2.2.4 Turbo-PROLOG 中的递归与表	(35)
2.2.5 fail 语句	(40)
2.2.6 Turbo-PROLOG 的数值计算	(41)
2.2.7 模块程序设计	(41)
2.2.8 同其它语言的接口	(44)
第三节 PROLOG 与 LISP 之比较	(44)
第三章 知识的内部表示	(47)
第一节 五种内部表示方案	(47)
3.1.1 形式一	(50)
3.1.2 形式二	(56)
3.1.3 形式三	(57)
3.1.4 形式四	(59)
3.1.5 形式五	(60)
第二节 知识的输入和内部表示转换	(61)
3.2.1 知识的输入形式	(61)
3.2.2 谓词 rule_read2 和 body_read2	(63)
3.2.3 谓词 rule_read3 和 body_read3	(72)
3.2.4 谓词 rule_read4 和 body_read4	(75)
3.2.5 谓词 rule_read5 和 body_read5	(77)
第四章 自动知识获取程序的设计	(81)
第一节 知识的自动获取	(81)
4.1.1 知识获取的自动化问题	(81)
4.1.2 从示例中学习	(83)
4.1.3 类比学习	(85)
第二节 一个交互式知识获取程序的设计	(87)
4.2.1 知识获取步骤	(87)
4.2.2 谓词 skeleton_acquisition 和 body_acquisition	(88)
4.2.3 谓词 knowledge_adapt	(93)

第三节	多目标定值的一种权系数学习模型	(100)
4.3.1	多目标定值问题	(100)
4.3.2	一个实际例子	(101)
4.3.3	学习步骤	(103)
4.3.3.1	初始权系数获取	(104)
4.3.3.2	权向量的调整	(106)
4.3.3.3	实例完善	(107)
第四节	智能系统的技能提高问题	(108)
4.4.1	显式知识获取与技能提高	(108)
4.4.2	人类的技能提高	(110)
4.4.3	实现系统的技能提高	(112)
第五章	推理机制的设计	(114)
第一节	基本正向推理算法	(114)
5.1.1	算法设计	(114)
5.1.2	初始证据获取谓词 <code>fact_acquisition</code>	(116)
5.1.3	谓词 <code>forward</code>	(118)
第二节	一个基于知识排序的线性正向推理 算法	(122)
5.2.1	知识排序谓词 <code>bt_sort</code>	(123)
5.2.2	谓词 <code>linear_forward</code>	(127)
第三节	基本反向推理算法	(129)
5.3.1	算法设计	(130)
5.3.2	谓词 <code>backward</code>	(130)
第四节	一个双向推理算法的设计	(134)
5.4.1	算法设计	(135)
5.4.2	谓词 <code>br_rsr</code>	(136)
第六章	不精确推理实现的问题和求解	(139)
第一节	不精确推理及实现上的问题	(139)
第二节	“规则架+规则体”: 一种求解	(141)

第三节	计算不精确度的谓词 <code>body</code>	(143)
第七章	一个跟踪解释程序的设计	(148)
第一节	跟踪解释的基本思想	(148)
第二节	一种跟踪解释方案的设计	(149)
第三节	跟踪谓词 <code>knowledge_trace</code>	(149)
第八章	知识库管理程序	(152)
第一节	知识库管理的内容	(152)
8.1.1	知识的一致性维护	(153)
8.1.2	知识库组织	(154)
第二节	用 PROLOG 实现的几个知识库管理 谓词	(156)
8.2.1	知识删除谓词 <code>delete_rule</code> 和 <code>delete_body</code>	(156)
8.2.2	规则体调整谓词 <code>body_adjust</code>	(157)
8.2.3	规则组调整谓词 <code>skeleton_adjust</code>	(158)
8.2.4	规则重复检测谓词 <code>repeat_detect</code>	(159)
8.2.5	规则矛盾检测谓词 <code>contradiction_</code> <code>detect</code>	(160)
8.2.6	冗余检查谓词 <code>redundant_skeleton</code>	(161)
8.2.7	规则架插入谓词 <code>insert</code>	(162)
8.2.8	知识库组织谓词 <code>partition</code> 及调用谓词 <code>load</code>	(163)
8.2.9	知识排序谓词 <code>bt_sort</code> 及 <code>td_sort</code>	(163)
第三节	推理网络的回路识别算法	(164)
8.3.1	回路识别算法	(164)
8.3.2	谓词 <code>cycle_find</code>	(166)
8.3.3	一般有向图的回路识别	(169)
附录:	Turbo-PROLOG 的内部谓词	(170)
参考文献		(194)

第一章 绪 论

专家系统是一种典型的知识处理系统，它面向现实世界中那些需要由专家来分析、求解的复杂问题或称不良结构问题，强调利用专家的专门知识和推理方法来实现专家水平上的问题求解。专家系统的设计有两个显著特点：一是专家系统对专家的依赖性，一个实际专家系统的设计一般自始至终都离不开一个或多个领域专家的合作，设计出来的专家系统的性能很大程度上决定于专家的水平和合作效果；二是专家系统的设计有不同于传统程序设计的一套技术，如知识获取、知识表示、基于知识的推理等，这些技术已作为知识工程学研究的核心范畴。本章概述专家系统设计的一些基本问题，并介绍作为本书技术基础的“规则架+规则体”表示。

第一节 专家系统的基本结构

专家系统强调利用领域专家的专门知识来求解那些需要专家才能有效解决的现实问题。所以，一个专家系统的两个核心组成部分是存放专门知识的知识库和利用知识库实现实际问题求解的推理机。

专门知识包括问题领域内与问题求解有关的定义、事实（这里指与具体问题无关的共性事实，如医疗领域内正常人有关指标的数值）和各种为领域技术人员所共同同意和一致接受的理论、方法，这样的知识一般可以在教科书或其它文献中查询到，称为共性知识，也包括那些在已发表的文字材料中难以找到、是专家凭经验和直觉获得的经验知识或称启发式知识。这种启发式知识

是专家所特有的个性知识，它们是专家赖以进行高水平问题求解的关键所在。知识的采集和知识库的建立往往是建造一个实际专家系统的中心任务。这涉及到选择合适的知识表示方法组织知识库和运用合适的知识获取手段来获取知识两大问题。许多专家系统建造中的知识获取是通过被称为知识工程师的专家系统设计者与专家的直接接触进行的，但不少专家系统中也开发有专门设置的知识获取模块，以使得专家和／或用户能够以非编程方式与专家系统直接打交道实现对知识库中已存贮的知识进行交互式修改、扩充。少数专家系统中的知识获取模块还能够把系统进行实际问题求解过程中的一些有用的中间结论记录下来、归纳成知识追加到知识库中或者对知识库进行自动调节、完善，这是我们通常所称的自动知识获取能力或自学习能力。而交互式实现的知识修改、扩充，则称之为交互式知识获取。

同时，由于知识库中启发式知识的特点，一方面，许多用户可能对这些知识不了解、甚至不信任，如果专家系统只能给出问题的求解结论而缺乏必要的说明，则其结论容易被视为“黑箱”中的预言而难以被用户接受；另一方面，启发式知识可能不完备、甚至产生相互矛盾，当一个问题的求解得出错误结论或者出现错误的迹象时，产生错误的根源在哪一条或哪几条知识上必须能定位出来。所以，不少专家系统中建立有专门的实现上述目的的解释机制。

知识库、推理机、知识获取模块、解释机制，以及系统与专家 and 用户交换信息的人机接口、用来存放具体问题数据（库）和求解过程相关信息的上下文一起，就构成了一个专家系统的基本结构。不带有知识获取模块和解释机制的专家系统可以纳入知识库系统的范畴。专家系统与一般的知识库系统的区别，除了这种功能上的差异之外，更本质上的区别，还应是系统的性能水平。在传统的算法式程序的基础上，加上一些逻辑推理的成份，甚至开发一个用于逻辑推理的知识库（通常以规则形式组织，称为规

则库) 和一个推理机, 所形成的系统一般只能称为基于知识的系统或知识库系统, 因为这个知识库 (或规则库) 的内容未必是只有专家才能提供的。如果我们不强调系统的性能水平, 那么所有用于逻辑推理的人工智能系统均可称为专家系统, 因为几乎所有人工智能系统都要用到一些定性的推理规则和实现推理的推理机制。

结构更为完整些的专家系统还应该包含以下几个方面的处理功能: (1) 人机接口上含有用于系统同人 (专家、用户) 交换信息的自然语言处理模块和 / 或图形、表格处理模块, 使得专家和用户能以尽可能自然、易实现的方式实现信息传输和结果反馈; (2) 知识库的组织和维护有一个专门的知识库维护模块或知识库管理模块 (见第八章), 实现对知识库中知识的矛盾、冗余检查和新知识的加入对已有知识的影响调节以及知识的存储、共享等。

图 1.1 给出了专家系统的一个结构示意图。

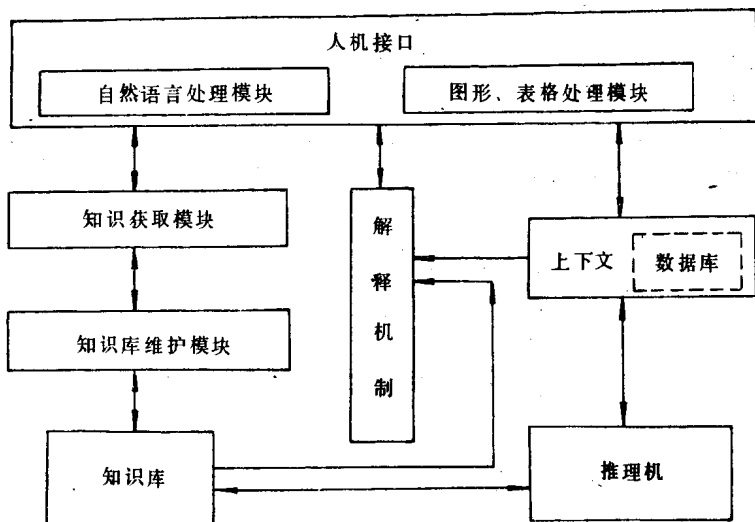


图 1.1 专家系统结构示意图

从图 1.1 我们可以看出专家系统结构同基于算法设计的传统程序的一些主要区别：(1) 专家系统中知识库级知识和运用知识库级知识的控制级知识（推理机）是相对独立的，这样便于实现知识库的修改和扩充，而在传统程序中两者合而为一、融合在程序中，从而传统程序系统的完善只能靠修改程序的方式进行；

(2) 专家系统中知识获取模块和相应的知识库维护模块能够在非编程状态下不断改进知识库、进而改进系统的性能；(3) 专家系统中解释机制能够给出问题求解结论的求解过程说明，增强了系统对非专家用户的透明性；(4) 专家系统的使用主要面向非计算机专业的用户，它追求自然、易于接受的人机接口的设计；(5) 专家系统强调符号推理和启发式知识的使用，这也是整个人工智能问题求解同基于算法设计的传统程序的区别。当然，实际的专家系统建造中我们也发现，许多不良结构问题的求解也常常避免不了数值计算，如专家的一些经验公式、专家对不确定现象和模糊现象的不精确处理，尤其在一些实际的工程领域，缺乏数值计算功能的系统建造常常被认为是不可行的。因为不管是工程设计还是工程规划，总少不了多个方案的核算、比较和选择，不经过定量处理是难以实现的。而这样的定量处理又未必可以用单纯的有限元分析或其它严谨的数学模型就能完成，随机应变的数值处理方法常常也是必需的。

事实上，专家系统与传统程序的最本质区别还在于专家系统所面向的问题特征和所能实现的性能水平。

第二节 专家系统的建立

专家系统的核心是知识，一般说来，知识库中知识的数量与质量是一个专家系统性能是否优越的决定性因素。从而专家系统建立的关键就在于知识库的建立。涉及知识库建立的两项关键技