

关惠玲 韩捷 著

设备故障诊断 专家系统 原理及实践

UZhang ZHENDUAN ZHUANJIA XITONG YUANLI JI SHIJIAN



机械工业出版社
China Machine Press

设备故障诊断专家系统 原理及实践

关惠玲 韩捷 著



机械工业出版社

本书主要阐述设备故障诊断专家系统的基本原理和各主要部分的建造技术。中心内容包括：故障诊断基本原理、方法及其在专家系统中的应用；故障征兆的全自动识别技术；专家知识的计算机表示；专家系统的推理机制和学习系统；专家系统的综合模型；并以单机组和多机组故障诊断专家系统实例说明原理的具体应用。

本书适合从事设备故障诊断专家系统研究与使用的工程技术人员及硕士研究生使用。

图书在版编目 (CIP) 数据

设备故障诊断专家系统原理及实践/关惠玲等著. —北京：机械工业出版社，2000. 10

ISBN 7-111-08219-2

I. 设... II. 关... III. 设备-专家系统诊断法 IV. TP277

中国版本图书馆 CIP 数据核字(2000)第 68648 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)

责任编辑:张亚秋 版式设计:霍永明 责任校对:陈延翔

封面设计:李雨桥 责任印制:郭景龙

北京京丰印刷厂印刷·新华书店北京发行所发行

2000 年 11 月第 1 版第 1 次印刷

1000mm×1400mm1/32·5.25 印张·200 千字

0 001—3 000 册

定价:16.00 元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话 (010) 68993821、68326677-2527

前 言

专家系统诞生于 70 年代, 80 年代其研究达到高潮, 90 年代进入应用发展期。

作者从 80 年代起以离心式压缩机组为对象开始故障诊断专家系统的研究, 应用系统于 1991 年安装于某大化肥厂的 CO_2 压缩机组并取得成功。从 90 年代起, 作者相继承担了国家、省、部级多项重大纵、横向项目, 研制了针对诸如汽轮机、压缩机、风机、发电机等大型机组的故障诊断专家系统。这些系统安装于生产现场, 在工程实际中接受了检验。作者在实践中积累了丰富的诊断知识和经验。本书旨在总结这些经验, 探索专家系统技术及其工程实践。

专家系统技术着重解决启发式和不确定式的知识类问题。故障诊断专家知识往往是经验的, 难以用数学公式描述。专家系统主要是研究启发式推理以便从功能上模拟专家。不断有学者用模糊数学去把握人类的非精确思维过程; 用灰色理论去模拟人们对非确定知识的处理过程; 用神经网络去模拟人脑的学习和思维过程; 力图把握不确定性推理和知识。专家系统实质上是采用不同于传统数学方法的算法模拟人类的思维过程, 相关内容属专家系统的范畴。由于设备故障诊断专家系统的诊断结果依赖于故障征兆, 而准确提取故障征兆的前提是监测系统无遗漏地捕捉异常信号并进行合理的信号处理, 所以与专家系统密切相关的动态检测、数据采集、信号分析等系统的设计也属于故障诊断专家系统的内容。

本书在写作过程中得到了作者所在研究所以及各应用企业的大力支持。作者所在课题组的郝伟、梁川、杨金才、董辛旻、李金光、王丽雅等老师从不同角度给予了作者无私的帮助, 在此一并致谢。本书第 1 及第 3~9 章均由关惠玲编写, 第 2 章由韩捷、关惠玲编写。全书由关惠玲统稿。

由于作者知识面与水平所限, 难免有不少不足之处, 恳请各位读者批评指正。

作者
2000 年 8 月

目 录

前 言

第一章 故障诊断专家系统概论	1
第一节 应用专家系统提高故障诊断技术水平	1
第二节 人工智能与专家系统	1
第三节 专家系统的关键概念	3
第四节 专家系统的结构特性及适用性	4
第五节 专家系统的应用概况及适用范围	13
第六节 设备故障诊断专家系统	16
第二章 设备故障诊断的基本原理	19
第一节 设备故障诊断的基本内容	19
第二节 设备故障诊断表的应用	21
第三节 设备故障的灰色诊断技术及应用	36
第四节 设备故障的模糊诊断技术	42
第五节 设备故障的神经网络诊断技术	51
第三章 设备故障征兆的获取和全自动识别	56
第一节 智能捕获设备故障信号的监测系统	56
第二节 子区域映射识别故障征兆	59
第三节 故障征兆的数字特征量	64
第四节 故障征兆的灰色识别	66
第五节 非线性振动征兆——混沌振动图像的识别	68
第六节 慢变参数的特征识别	70
第四章 知识表示和知识库	73
第一节 产生式规则表示法	73
第二节 基于灰色系统理论的知识表示	75
第三节 基于模糊的知识表示	76
第四节 基于神经网络的知识表示	77
第五章 推理机制	79
第一节 推理方法	79
第二节 非精确推理	97
第三节 基于数学模型的推理	101
第四节 基于实例的推理	104
第六章 学习系统	107

第一节 故障历系统	107
第二节 基于实例的学习	107
第三节 神经网络学习	108
第七章 故障诊断的综合模型	122
第一节 模糊与人工神经网络结合的故障诊断模型	122
第二节 规则与神经网络的结合模型	126
第三节 规则与模糊的结合模型	129
第八章 数据库和配置系统	131
第一节 数据库	131
第二节 配置系统	132
第三节 配置系统运行实例	133
第九章 故障诊断专家系统实例	136
第一节 二氧化碳压缩机组故障诊断系统	136
第二节 大化肥六大机组故障诊断专家系统	145
参考文献	159

第一章 故障诊断专家系统概论

第一节 应用专家系统提高故障诊断技术水平

术语“专家系统”不是表述一个产品，而是表示一整套概念、过程和技术。这些新概念、过程和技术能够使工程技术人员以多种不同的有价值的新方法使用计算机去更有效地解决工程问题。其中设备故障诊断问题是这些新方法的一个重要应用领域。

在过去的近 40 年中，美国近 600 家大公司中，有 80% 已经在研究专家系统技术。一些公司正在实验研究，而有许多公司已得出结论：专家系统的应用将明显地改善他们的工作方式和能力。

实际上，专家系统技术能够使计算机帮助人们分析和解决只能用自然语言描述的复杂问题。这样就扩展了计算机一般能做的计算和统计工作，使计算机程序具有了思维能力。这些具有思维能力的程序能够与决策者进行对话，并应用推理，建议不同的可能行为过程。将人类专具有具有的知识和推理技能编制进一个专家系统中，人们就可建造一个能像人类专家一样诊断故障和提供建议的程序系统。

同样，专家系统技术能使不具有编程能力的人们建立功能强大的程序系统。这样，对编程一窍不通的工程技术人员能够把他们的知识输入专家系统。其他同样缺乏编程能力的工程技术人员通过对话能够很容易地检查这些在系统内部的知识，并在必要时，修改这些知识。

因此，与专家系统有关的概念和技术是革命性的，专家系统能将本领域众多专家的经验汇集在一起，使人类共享知识成为可能。将专家系统应用于故障诊断技术领域必将极大地提高故障诊断的技术水平。

第二节 人工智能与专家系统

人工智能（Artificial Intelligent 简称 AI）像物理学一样，是一门研究学科。该学科是计算机科学中研究、设计和应用计算机去模仿和执行各种拟人任务的一个分支。人类的许多活动，如解题、猜谜语、进行讨论等等都需要智能。人类的智能还没有被全部了解和掌握。目前，人工智能研究者只能研究应用计算机去解决

一些特殊的智能问题。一些 AI 研究者关心编制识别人类声音的计算机程序，关心建造能从一种语言翻译为另一种语言的程序；一些人关注发展使机器人识别物体和推理不同动作顺序的技术；还有人专心于开发能像人类专家一样推理的程序，等等。在近 40 年中，人工智能的许多概念、过程和技术已显示出巨大的商业价值，已有一些产品走向市场。人工智能最活跃的研究领域包括：①自然语言理解；②机器人；③自动智能程序设计；④问题求解程序；⑤人工神经网络；⑥机器学习等；⑦专家系统。专家系统是其最成功、实用性最强的一个领域。

(1) 自然语言理解 自然语言是指任何一种人类所讲的语言，如英语、汉语等。一些人工智能研究者正努力建造一种能与人以一种自然语言对话的计算机硬件和软件。一个经理拥有一个自然语言介面，即可像询问助手一样向计算机询问情况。

(2) 机器人 建造机器人装置不仅是人工智能研究者关心的事。人工智能仅仅关注机器人中需要计算机程序引导去分析和解决它们将遇到的问题部件。这样的“智能机器人”应用人工智能技术去看和操作它们将遇到的物件。实际上，机器人能“智能”行动，是因为有一个数学模型存在其计算机内，这个模型使机器人能识别事物并能分析机器人动作之后事物将作何种反应动作。

(3) 探索编程或自动编程 探索编程是指应用 AI 概念和技术去研制大型应用程序。这样的概念和技术包括新的编程语言和编程环境、仿真等被应用于增加传统程序员的生产能力。这些技术融汇到传统的编程过程中，正在逐步改进传统编程方法。

自动编程技术应用 AI 技术，通过指定一个目标交给一个自动编程系统，产生该应用的绝大多数专用码，程序员以这种方式研制计算机程序。

(4) 机器学习 学习是一个有特定目的的知识获取过程，其内部表现为新知识结构的不断建立和修改，而外部表现为性能的改善。狭义讲，学习系统把导师或专家提供的信息转换成能被系统理解并应用的形式。

机器学习还有助于发现人类学习的机理和揭示人脑的奥秘。这是一个已经得到重视，但还未达到理想境地的研究领域。

(5) 专家系统 在人工智能研究领域，专家系统最引人注目。这些问题求解系统当初被称为“专家系统”，是指在很高深的专业工作中，它在功能上与人类专家的作用相当。

一个专家系统是一个程序，此程序明显具有来自人工智能近期研究的概念、过程和技术相结合的特性。人们应用这些技术，设计和研制应用知识和推理技术分析、解决工程问题的计算机程序。

专家系统的代表作是 80 年代初斯坦福大学研制成功的 MYCIN 医疗诊断专家系统。它可以在不知道原始病原体的情况下诊断脑膜炎和其他细菌感染。

第三节 专家系统的关键概念

专家系统有三个最主要的概念：①表达知识的新方法；②启发式搜索；③将知识与推理和控制的分离。

一、知识的表达

知识比信息更复杂，但更有价值。通常情况下，知识指有关一个特定主题的经验有序集合体。如果说一个人在某方面知识丰富，意思总是指此人不仅知道这方面的许多事实，而且他还可以对相关的问题进行分析并做出判断。

AI 研究者已经集中精力研究知识的动词和图形表达，而不再是传统软件人员所关心的数学计算。因此，在传统程序员寻找将一个问题分解为一些能用数学术语表示并用算法处理的子问题时，AI 程序员则对用句子和结构表达、用逻辑推理处理知识技术更感兴趣。传统编程技术已经深入研究了能用数学分析解决的问题。如果计算机要进一步解决管理者遇到的更复杂的问题，程序员必须能够帮助管理者分析和解决只能用自然语言表达的问题。将自然语言表达进行编码，用逻辑推理过程操作编码知识，将大大扩展计算机所能解决的问题的类型范围。

二、启发式搜索

传统的计算机计算过程，依赖于对一个问题的每一个元素和每一步骤的详细分析。这限制了计算机所能解决问题的范围。人类在解决许多问题时是凭启发式思维（或经验）进行的。这些经验使他们将大问题缩减为一个较小的问题。完全的启发式思维会导致错误，因为它们不能保证答案总是正确的，它们仅仅增加找到有用答案的可能性。

比如，要在一个公用计算机硬盘上找到某个用户所编制的一个特定文件。硬盘内容很多，要一个一个文件寻找是很费时费事的。有一定计算机使用经验的人都知道，文件通常是按类、按不同的用户存放在不同的子目录下。凭这个经验，就可以将问题缩小，很快进入该用户的子目录下找所需文件。这说明了启发式搜索的关键性质：依赖特定环境知识、来源于实践经验。启发式知识不能保证一定成功。比如，该文件的编程者可能是个新手，他没有建自己的子目录，将文件存到其他地方了。该启发式知识就找不到答案。所以，启发式知识表达的可能性知识，仅仅在可能碰到的各种情况中的一些情况下起作用。

任何一个有成就的专家都拥有许多启发式知识。当遇到一个不能完全分析的情况时，他将利用启发式知识去解决问题。

利用启发式知识，AI 程序员在传统程序员无能为力的情况下建议行动。另一方面，像人类专家一样，新程序将提高他们建议的质量，但有时可能会错。

启发式编程技术迅速扩展了计算机所能做的工作范围，使程序员可以研制更

大、更复杂的程序，能分析解决非常模糊的问题，并做出解答。然而启发式搜索将依赖不确定性技术，应用不确定性技术让专家对自己的判断定出准确度的量值。基于启发式的程序不是给出一个正确答案，而是建议几种可能的选择，并提供每一种选择的可能性估计值。

因此，专家系统技术中能应用传统方法精确处理的内容，一定要用传统的方法处理，只有当问题很复杂、很模糊无法精确时，才能应用不确定技术。如果什么都去模糊，那就本末倒置了。

三、知识与推理和控制的分离技术

传统的编程，有关一个问题的知识与操作这些知识去解决该问题的过程是混合在一起的。当非程序人员看到一个应用程序的源程序时，无法断定程序是关于什么或是解决什么问题的，必须依靠程序员正确地表示知识。这非常象人们用电报通信的状况。由于信息的发出人和接收人不能读懂点和线，他们不得不假定电报操作人员操作正确。这给开发大的应用程序的程序员身上压了一副重担，因为专家无法帮助程序员把知识袋放入程序中。

AI 研究者发明了将知识与应用知识的过程相分离的技术。在功效上，任何一个专家可以检查在一个专家系统里的知识，并判断其正确性。另外，当关于一个问题的知识改变时，专家能找到相应的规则并提前去修改它们。

知识与推理和控制的分离，应该算是人工智能研究的最重要概念。表面上看起来很简单，不仅人工智能语言能实现，传统的编程语言也能实现。这项新成果的功效使非程序员编程成为可能。如果一个程序的研制者既需要去分辨包含于一个程序的知识，又需要断定处理这些知识的确切过程，那么这个程序的研制只能由受过编程技术训练的人承担。一旦创造出能产生处理一个给定知识体的自身算法的程序环境，任何一个能提供此知识体的人即可创造出一个程序。

第四节 专家系统的结构特性及适用性

一、专家系统的基本结构

专家系统主要由知识库、数据库、推理机、学习系统、上下文、征兆提取器和解释器组成，如图 1-1 所示。

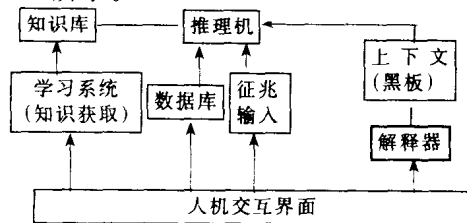


图 1-1 专家系统结构示意图

二、专家系统的基本知识

1. 专家系统各部分简介

(1) 知识库 主要用来存放领域专家的专门知识。在知识库中,知识是以一定的形式表示的。知识的表示方法有许多种,常用的有:产生式规则、网络、框架、决策树、谓词逻辑。其中以产生式规则表示最为常用。

例如,人类专家的经验为如下规则:

- ① if 1x 振幅高 then 不平衡
- ② if 正弦波 then 不平衡
- ③ if 正弦波 then 轴承间隙大

则产生式规则以 prolog 语言相应表示为:

- ① fault (unbalance): ——spectrum (1x)
- ② fault (unbalance): ——wave (sine)
- ③ fault (large clear): ——wave (sine)

(2) 数据库 数据库通常由动态数据库和静态数据库两部分构成。静态数据库存放相对稳定的参数,如离心式压缩机械的设计参数:额定工作转速,额定介质流量、压力,振动红、黄灯值等。动态数据库是运行过程中的机组参数,如某天某时的工作转速、介质流量、振动幅值等。这些数据都是推理过程中不可少的诊断依据,它们以“事实规则”的形式表达为:

Workspeed (14000) ——工作转速 14000 r/min

flued (3000) ——工作介质流量 3000 m³/s

stop (100) ——振动红灯报警值 100μm

alarm (80) ——振动黄灯报警值 80μm

这些数据可以看作没有条件的规则,因此有些专家系统将数据库和知识库合二为一。

(3) 推理机 推理机的功能是根据一定的推理策略从知识库中选择有关的知识,对用户提供的证据进行推理,直到得出相应的结论为止。推理机包括推理方法和控制策略两部分。

1) 推理方法 分精确推理和不精确推理两类:

①精确推理 把领域的知识表示成必然的因果关系,推理的结论是肯定的或否定的。

②不精确推理 在专家给出的规则强度和用户给出的原始证据不确定性的基础上,定义一组函数,求出结论的不确定性度量。其基本做法是,给各个不确定的知识某种确定性因子,在推理过程中,依某种算法计算中间结果的确定性因子,并沿着推理链传播这种不确定性,直到得出结论。

2) 控制策略 主要指推理方向的控制及推理规则的选择策略。推理有正向

推理、反向推理及正反向混合推理等。

①正向推理 由原始数据或原始征兆出发,向结论方向的推理。推理机根据原始征兆,在知识库中寻找能与之匹配的规则,如匹配成功,则将该知识规则的结论作为中间结果,再去寻找可匹配的规则,直到找到最终结论。

②反向推理 先提出假设,然后由此假设结论出发,去寻找可匹配的规则,如匹配成功,则将规则的条件作为中间结果,再去寻找可匹配的规则,直到找到可匹配的原始征兆,则反过来认为此假设成立。

以上两种推理均为单向推理。单纯的正向推理,目的性不强,搜索效率低;单纯的反向推理,初始假设盲目性大,因此通常采用正反向混合推理。

③正反向混合推理 先根据重要征兆,通过正向推理得出假设,再以假设去反向推理,寻找必要条件,如此反复。

例如,在旋转机械故障诊断中,振动信号是主要的诊断依据,大量的诊断知识是以振动理论为基础的。在振动信号中,频谱又是诊断的首要依据。因此,通常旋转机械的故障诊断是以频谱作为正向推理的征兆,得出故障假设集,再以假设集去指导反向推理。

(4)学习系统(知识获取系统) 知识获取过程可以看作是一类专业知识从知识源到知识库的转移过程。知识源有专家、资料和书本等。知识获取过程包括在知识库创建时识别出必要的知识并将其形式化;建成的知识库经常会发现有错误或不完全,所以知识获取过程还包括对知识库的修改和扩充。

早期的专家系统完全依靠专家和计算机工作者把领域内的知识总结归纳出来,然后将它们程序化后建立知识库。此外,对知识库的修改和扩充也是在系统的调试和验证过程中手工进行的。后来,一些专家系统或多或少地具有了自动知识获取功能。然而,基于规则的学习系统灵活性较差,且知识库的维护较难。最近几年专家系统和神经网络结合后,才大大改观了知识自动获取的局面。

(5)上下文(黑板) 上下文即存放中间结果的地方,给推理机提供一个笔记本记录,指导推理机工作,其功能相当于一个工作过程的“记录黑板”。

在 prolog 语言编程系统中,通常用 database 实现,以 assert 谓词往黑板上写,以 retract 谓词擦黑板。

(6)征兆提取器 在故障诊断领域,征兆通常是采取人机交互方式,由人机交互接口送入系统。显而易见,人机交互容易产生因人而异的弊端。同一个专家系统,因操作者水平不同会产生不同的结果。

故障诊断准确的前提是故障征兆正确。故障征兆的识别不仅重要而且难度大,因为现代设备的动态信号不仅包含随机因素、混沌因素等,而且常常存在并发故障的复合因素。因此,故障征兆的自动识别应是故障诊断专家系统必不可少的一个组成部分。

征兆的全自动识别有一种较简单实用的方法，就是特征参数计算。比如正弦波形的识别，可以用被识别波形与正弦波的相似系数为识别的特征参数。

(7) 解释器 透明性是专家系统性能的衡量指标之一。透明性就是专家系统能告诉用户自己是如何得出此结论的，根据是什么。解释的目的是让用户相信自己。

2. 故障诊断专家系统示例 以上 7 大部分是专家系统的基本组成，其中知识库和推理机是核心部分。下面以 prolog 语言举一旋转机械故障诊断专家系统的例子，仅摘取知识库和推理机的一小部分，以说明专家系统的原理。

例 1.1 旋转机械故障诊断专家系统其知识库内容为：

如果 1 倍频高 则不平衡
 如果 1 倍频高 则轴承间隙大
 如果 2 倍频高 则不对中
 如果 2 倍频高 则转子裂纹
 如果波形为正弦波 则不平衡
 如果波形为正弦波 则轴承间隙大
 如果波形为削波 则摩擦
 如果为正弦波且转子两端振动差别大 则轴承间隙大
 如果为正弦波且转子两端振动差别小 则不平衡
 如果轴向振动大 则不对中
 以此知识库为基础编写的程序如下：

```
domains
    specdom = spec (string, real)
    wavedom = wave (string, real)

database
    fault (string, string, real, real)    /* fault      */
    ch (string)                          /* channel    */
    spec (string, real)                   /* spectrum   */
    wave (string, real)                   /* wave       */
    pp (string, real)                     /* P - P value */
    app (string, real)                    /* P - P value */

predicates
    sdiagnose (string, real, real)
    single - ch - def (string)

goal
    single - ch - def (CH).
```

```

clauses
/ * ..... * /
/ * diagnosing by single channel * /
/ * ..... * /
single - ch - def (CH): —
    ch (CH),
    sdiagnose (CH, —, —),
    fail,
    !
single - ch - def (—): —
!
/ * ..... / *
/ * The rule about spectrum. / *
/ * ..... / *
sdiagnose (CH, CF, SF): —
    spec ("2x", A),
    CF1 = A/25. 0,
    SF1 = A/40. 0,
    assertz (fault (CH,"misalignment", CF, SF)),
    assertz (fault (CH,"crack", CF1, SF1)).
sdiagnose (CH, CF, SF): —
    spec ("1x", A),
    CF = A/25.0,
    SF = A/40.0,
    assertz (fault (CH,"unbalance", CF, SF)),
    assertz (fault (CH,"lose", CF, SF)).
/ * ..... * /
/ * The rule about wave. * /
/ * ..... * /
sdiagnose (CH, CF, —): —
    wave ("sine", A),
    CF = A,
    fault (CH,"unbalance", —, SF1),
    assertz (fault (CH,"unbalance", CF, SF1)),
    assertz (fault (CH,"lose", CF, SF1)).

```

```

sdiagnose (CH, CF, -): —
    wave ("clip", A),
    fault (CH,"rub", -, SF1),
    assertz (fault (CH,"rub", CF, SF1)).

/* ..... */
/* The rule about addition. */
/* ..... */

sdiagnose (CH, CF, -): —
    pp (CH, A1),
    pp (COCH, A2),
    A1/A2 < 1.3,
    A1/A2 > 0.7,
    fault (CH,"unbalance", CF1, SF1),
    CF = CF1 + 0.3,
    assertz (fault (CH,"unbalance", CF, SF1));
    pp (CH, A1),
    pp (COCH, A2),
    A1/A2 > 1.8,
    A1/A2 < 0.5,
    fault (CH,"lose", CF1, SF1),
    CF = CF1 + 0.3,
    assertz (fault (CH,"lose", CF, SF1)).

sdiagnose (CH, CF, -): —
    app (CH, A1),
    fault (CH,"misalignment", CF1, SF1),
    CF = CF1 + 0.4,
    assertz (fault (CH,"unbalance", CF, SF1)).

征兆识别的结果为 (动态数据库中存的内容):
spec ("1x", 30),
wave ("clip", 0.6)
pp ("ze001", 32),
pp ("ze003", 26),
app ("ze001", 25),

```

3. 专家系统的开发工具 为了提高专家系统的设计效率, 缩短系统的研制周期, 已出现了许多专家系统的开发工具。这些工具大致分为以下几类:

(1) 面向人工智能的通用程序设计语言 如 lisp、prolog 等。由于这些语言可以将符号直接写在程序中,能够以接近自然语言方式表达知识和规则,因此对建立专家系统特别有效。

(2) 通用知识表示语言 这是针对知识工程发展起来的程序设计语言,这些语言并不与具体的体系和范例有紧密联系,也不局限于实现任一特殊的控制策略,因而便于实现较广泛问题的处理。

由于应用目的不同,知识类型不同,其知识表达方式也不同,所以开发了若干种流行的知识表示语言,如产生式语言系统 OPS—5 以及基于框架理论的知识表示语言 RLL、KPL-UNITS 等。

(3) 专家系统外壳 专家系统外壳提供推理机制、解释功能及知识获取模块。只要领域专家根据知识模块的提问,将知识教授给系统,这些外壳就变为专家系统了。这类系统的典型代表有 EMYCIN、KAS 和 EXPERT 等。

(4) 组合式专家系统研制工具 它向用户提供多种知识表示方法和多个推理机制,用户根据需要,选择合适的知识表示模块和推理机制组合成自己所需的专家系统。

从以上分析可看出,基于规则的专家系统有以下弱点:

(1) 知识库管理困难。非结构化的知识库中,规则与规则之间的关系是不明确的,仅仅利用简单的规则加入、删除和修改等操作建立起来的知识库,只是规则杂乱无章地堆砌,规则之间的重复、矛盾等现象难免发生,因此知识库的维护成了一大难题。

(2) 符号匹配的推理效率太低。

由于这些原因,我们首次建立了基于数学模型的离心式压缩机组故障诊断专家系统,尝试用数学模型去整体把握知识库,从根本上解决知识冲突、矛盾和冗余问题。计算机最大的长处是计算,用数学模型去建造专家系统,可以将大部分推理转化为数值计算,大大提高推理效率。

三、专家系统是非过程式程序

编程示例被分为过程式和非过程式的。

所谓过程式就是通常的编程形式,如用 Fortran, Turbo C 等语言的编程。一个算法是以有限步骤解决一个问题的方法。包含在程序中的一个算法的执行是过程式的。算法编程、过程编程和传统编程是同义词,都指非人工智能型程序。过程式程序是顺序执行的,一个语句接着一个语句,除非碰到分支说明。所以过程式程序通常也称作顺序程序。然而,因为顺序编程包含有太多的限制,现代编程语言都支持循环,严格的顺序是不存在的。

过程式编程最突出的特性是程序员必须严格指明一个程序的结果是如何得到的。而非过程式编程,程序员指明目标是什么,让系统决定如何去执行。非过程

式程序的主要类型如图 1-2 所示。

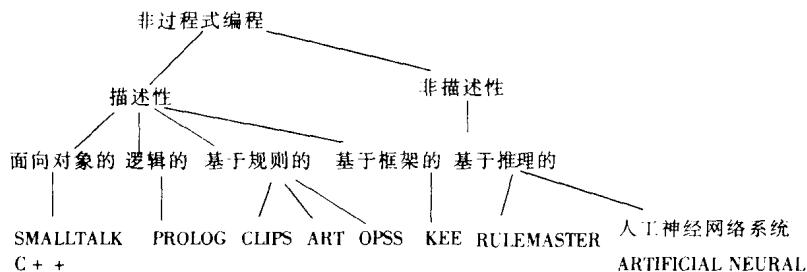


图 1-2 非过程式编程的类型

专家系统被认为是描述性的非过程式程序，比如一个基于规则的专家系统，其知识库中的任何一条规则只要前提条件与事实匹配成功，就可以被激活，被放入记事簿中。规则被送入的顺序不影响其被激活的顺序。因此，程序语句的顺序没有指定一个严格的控制流程。

下面以一个工资表为例进行比较。

$\text{Salary} = \text{rate} * \text{hours}$.

rate 是每小时报酬，hours 表示工作的小时数。

用过程式程序实现如下（以 C 语言为例）

```

const    MAX = 20
float    salary [MAX], rate [MAX], hours [MAX]
char     * name [MAX] [5]
main ( )
{
    for (i = 0; i ≤ 20; i + + )
        {
            salary [i] = rate [i] * hours [i];
            procout out << salary [i] << rate [i] << hours [i];
        }
}

```

尽管用的是循环，但程序的执行顺序是定死的，算完第一个人，算第二个人。

每一次执行此程序都是如此过程。而且必须是已知每小时工作酬劳和工作的小时数，求出工资额。

PROLOG 编的专家系统程序如下

```

database
    rate (symbol, real)
    hours (symbol, real)
predicates
    salary (symbol, real)

```