

高级

turbo
Prolog

邱建设编译
武汉大学出版社

IBM-PC(0520)系统

高级 Turbo Prolog

IBM-PC (0520) 系统

邱建设 编译

武汉大学出版社

1989

高级 Turbo Prolog

IBM-PC (0520) 系统

邱建设 编译

*

武汉大学出版社出版发行

(武昌 珞珈山)

*

湖北省农科院科技杂志印刷厂印刷

*

787×1092毫米 1/32 10.75印张 240千字
1989年9月第1版 1989年9月第1次印刷

印数:1-3000

ISBN 7-307-00553-0 / TP·20

定价:6.20元

前 言

本书内容包括了 Turbo Prolog 版本 1.1 和 Turbo Prolog 工具箱 (Toolbox)。这个工具箱提供了 80 多个工具和 8000 多行源程序代码。

Turbo Prolog 是专为微机 (IBM-PC 及其兼容机) 设计的第五代程序设计语言, 该书通过一些典型的人工智能问题及如何用 Turbo Prolog 来具体实现, 使读者用 Turbo Prolog 编程的技巧提到较高一级的水平。该书为 Turbo Prolog 应用于各个领域, 提供了一个良好的开端, 读者对书中的程序 (例如专家系统), 稍加扩展和增强, 便可应用到自己的专门领域。本书内容包括

- 专家系统
- 问题求解
- 自然语言处理
- 视觉与模式识别
- 机器人学
- 逻辑

本书是以 Herbert Schildt 编写的 “Advanced Turbo Prolog Version 1.1” 为蓝本编译而成的, 是 1987 年美国的最新版本。

全书理论联系实际, 收集了大量丰富的珍贵程序, 处处以这些实用的 Turbo Prolog 程序实例, 阐述人工智能的理论及具体的人工智能程序设计技巧。全部程序均在 IBM-PC 机上通过。

编译者在编译过程中参考了白为民同志的部分译稿,在此表示感谢。

邱建设
1989.8
于武汉大学

目 录

第一章 人工智能——简单回顾	(1)
§1.1 AI 的简短历史	(2)
§1.2 什么是智能程序?	(5)
§1.3 AI 的主要课题	(10)
第二章 问题求解：用搜索寻找答案	(13)
§2.1 术语和表示	(13)
§2.2 组合爆炸	(15)
§2.3 搜索技术	(17)
§2.4 搜索评价	(18)
§2.5 深度优先搜索技术	(21)
§2.6 宽度优先搜索技术	(28)
§2.7 加入启发式信息	(32)
§2.8 爬山法搜索技术	(33)
§2.9 最小成本搜索技术	(39)
§2.10 搜索技术的选择	(42)
§2.11 寻找最优解	(42)
§2.12 农夫、狐狸、鸡子和谷物	(45)
§2.13 回到丢钥匙问题	(58)
第三章 专家系统	(60)
§3.1 什么是专家系统	(60)
§3.2 专家系统怎样工作	(63)

§3.3	建立通用的专家系统	(69)
§3.4	知识工程	(91)
第四章	自然语言处理	(95)
§4.1	什么是自然语言处理	(95)
§4.2	自然语言处理的方法	(96)
§4.3	限制性语言	(97)
§4.4	状态机NLP分析器	(98)
§4.5	上下文无关递归NLP分析器	(109)
§4.6	干扰处置分析器	(123)
第五章	视觉与模式识别	(133)
§5.1	过滤、对比和阴影	(133)
§5.2	二维系统	(135)
§5.3	三维系统	(137)
§5.4	普通的识别问题	(142)
§5.5	二维模式识别	(144)
§5.6	通用系统	(177)
第六章	机器人学	(178)
§6.1	机器人手臂	(179)
§6.2	工业机器人	(181)
§6.3	自律机器人	(185)
§6.4	建立机器人模拟器	(186)
§6.5	程序	(189)
第七章	机器学习	(215)
§7.1	两种学习	(215)
§7.2	怎样学习分类描述	(219)

§7.3	知识表示	(225)
§7.4	成功与近似样品过程的实现	(229)
第八章	逻辑与不定性	(239)
§8.1	逻辑	(239)
§8.2	婴儿Prolog	(247)
§8.3	不定性.....	(258)
第九章	显示人的特征	(279)
§9.1	是骗术吗?	(279)
§9.2	有什么好处?	(280)
§9.3	人与机器	(280)
§9.4	看病	(283)
§9.5	类似人的计算机的含义	(297)
附录A	与其他语言的接口	(299)
附录B	Turbo Prolog Toolbox(工具箱的使用)	(312)

第一章 人工智能 —— 简单回顾

直到现在,还有许多人把人工智能领域,看作是计算机科学的阴暗面。他们认为:与 *Shelley* 的 *Frankenstein* 博士试图创造生命相似, AI (artificial intelligence) 程序员为创造思想而工作。人们甚至把人工智能研究人员矛盾地看成既是计算机科学界的精英,又是“精神错乱的极端分子”。当被问及有关机器智能的可行性与实用性时,大多数挣工资的程序员,通常都会谨慎地避免提及 AI,而总是说“仍需要进行大量的研究”,或者“在未来世界的某个时候,总会有重大发现的,但目前还没有什么建树”。有关人工智能的观点已发生了根本变化,这是确定无疑的。

在不到五年的时间内,人工智能在计算机科学中,已经从死气沉沉、默默无闻的状态转变为自晶体管时代以来该领域中最热门的东西。这个快速的变化主要有四方面的原因:专家系统的成功,这是第一个从经济上讲完全成功的 AI 产品;日本人对 AI 的广泛宣传;AI 技术缓慢但稳定地引入现有的应用程序中;最后一个原因是人工智能时代已经到来这一事实。

随着 Turbo Prolog 的出现,人工智能程序设计已逐渐为广大程序员所理解。长久以来,全面体现 AI 特点的语言,只能在大的主机上才能实现。Turbo Prolog 使实际的 AI

应用,在微机上得以实现。由于 Turbo Prolog 有很好的常用硬件支持的特点,象颜色、图形和声音,使它很容易为专业程序员开发软件服务。

通过本书,你可探索 Turbo Prolog 较适宜执行的各种类型的工作。通过实际开发应用 AI 的各个领域,将磨炼你的 Turbo Prolog 编程技巧。有几个方面和专题形成了人工智能的基本内容。本书其余各章,讨论了其中最主要最常用的内容,在你开始这些研究之前,本章首先重温了 AI 的历史,为你阅读以后各章打下基础。然后向你明确阐述什么是“智能”程序。最后,本章还简述了本书以后各章的 AI 的主要专题。

§ 1.1 AI 的简短历史

要指出通常所说的 AI 的精确起始日期是很困难的,或许 AI 诞生的荣誉应给 *A. M. Turing*,是他发明了存贮程序的计算机。第一台计算机实际上是专用机,要解不同的问题,必须重新布线。*Turing* 认为程序可以当作数据存贮在计算机的存贮器中,然后执行程序。这个思想构成了所有现代计算机的基础。存贮程序使得计算机可以快速方便地改变其功能,这只要通过运行一个新程序就可做到。这个能力意味着计算机可以自己改变自己的功能——学习或思考。可是,通常所认为的 AI 开始于 1960 年左右,那时 MIT 的 *John McCarthy* 创立 LISP——第一个用于人工智能研究的语言。LISP 的某些地方类似于 Prolog:象对表容易处理,并利用递归当作程序控制的主要形式。但是,LISP 却缺少 Prolog

的许多新特点和简明的语法。一般说来, *artificial intelligence* 这一术语是由 *Marvin Minsky* 首次使用, 他也是 MIT 的学者。1961 年, 他写了一篇论文, 题目是 “*Steps Towards Artificial Intelligence*”。六十年代人们对使计算机能思考持广泛的乐观态度, 因为毕竟在六十年代, 人们看见了第一台会下棋的计算机, 第一次由计算机进行了数学证明, 并且在这期间还有一个广为宣传的非常著名的程序 ELIZA。这个程序是由 MIT 的 *Joseph Weizenbaum* 在 1964 年写成的。程序 ELIZA 就象一个精神分析专家。在这种形式的分析中, 精神病专家起到一个被动的作用, 他通常只是重复病人自己的话, 而不是引导进行对话。重复的工作方式计算机是容易实现的。那时这个程序引起了很大的轰动。人们问, “计算机可以这样用吗? 计算机精神病专家能比人类精神病专家强吗? 这样的程序允许吗?” 就是 *Weizenbaum* 本人, ELIZA 的编制者, 也写了一本题为 “*Computer Power and Human Reason*” 的书, 对自己的程序表示怀疑(记住, 六十年代也是人们对自动化普遍感到惊恐的时代, 对仅仅二十多年前人们的心情有时也是很难理解的)。

由于 AI 取得了明显的成功, 因此产生一个具有类似人类智能的程序的目標就要实现了。然而情况并非如此。

六十年代人们还没有清楚地认识到这些个别成功的程序推广为灵活的智能程序的困难, 本书将要追寻其中的一些根由。程序员如果想增加某个程序的一般性, 就需要有比现在正使用的程序有更多的计算机资源, 这样, 或者是内存很快耗尽或者是执行时间慢得无法忍受。

到了七十年代中期, 大容量的计算机已很普通了, 运算

速度也增加很快。然而,即使有了这么多的进步,许多老的AI方法仍然不能成功,因为这些方法本身的效率不高。例如,对数组排序这样一个简单的问题,如果你用冒泡排序算法,排序时间将与 N^2 成正比,这里 N 是数组元素的个数。这就意味着如果对 10 个元素的数组排序需要 1 秒钟的话,对 100 个元素的数组排序就要 100 秒钟,可以看出,到了某一点,数组排序的时间将会比一个人的平均寿命还要长。不管计算机变得多快,对与 N^2 成正比的排序算法来说,它随着 N 的变大将迅速变慢到不能忍受的地步。最好是改进排序算法,提高它的效率。例如,快速排序算法的排序时间与 $N^{1.2}$ 成正比——这是一个很显著的改进。图 1.1 表明了快速排序与冒泡排序算法曲线的差距。与需要新的排序算法来提高排序速度一样,许多 AI 问题也需要改进其程序,以使程序执行的速度足够快或能更好地利用内存,这样这些 AI 问题才能得到解决。事实上,正如第二章所述,Prolog 的产生是其中的突破之一。

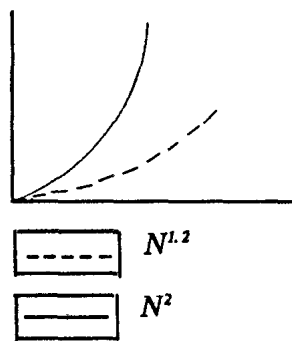


图1-1 N^2 曲线与 $N^{1.2}$ 曲线的比较

到了七十年代末,在 AI 的几个特别领域,如自然语言处理、知识表述和问题求解等,取得了一些成功。这些成功奠定了第一个商业化 AI 产品专家系统引入的基础。专家系统是一个包含某一领域信息的程序,当向其提问时,能象人类专家一样回答问题。较早的专家系统 MYCIN,由斯坦福大学开发成功,能帮助内科医生诊断疾病。本书将包括后来的专家系统的深入发展。

虽然自 1981 年以来,Prolog 在美国已得到普及,但由于一个完整的 Prolog 很难在微机上运行,所以多数仍限于在大学使用。在这些稀少的可完全运行的版本中,还缺少标准化,Turbo Prolog 的出现,是 AI 历史上一重要事件。Turbo Prolog 为广大用户在自己的微机上开发 AI 程序,提供了广阔的天地。同时它也为用户在许多机种上提供了一个标准 Prolog。可以确信,这仅是 AI 具体应用的开始,前途是无量的。

· §1.2 什么是智能程序?

在用 Prolog 进行 AI 领域的开发之前,你必须理解,对计算机来说智能程序意味着什么。智能程序是如何区别于“非智能程序”的。为了讨论的方便及与传统提法一致,本书将把思考程序称为智能程序,然而仍有很多争论,这种争论并不那么容易解决,因为这部分地取决于智能定义的解释。两种结论各有令人信服的证据支持。在争论中讨论的问题之一,就是智能程序与非智能程序有什么区别,本节将给智能程序一个合理的定义。

定义什么是智能程序,意味着人们知道智能的含义。有一部字典是这样定义智能的:“(智能是)理解事实、命题及它们之间关系并对其进行推理的一种能力。”这样的定义就产生了问题,“推理的意思是什么呢?”在这里推理意味着思考,这就是麻烦所在!很久以前,有人提出人们不能解释他们是怎样思考的,但可以说出他们在想什么!事实是人类并不真正明白他们是怎样思考的(如果他们明白,也许让计算机来进行思考也就不会有这么巨大的一项工作了)。

如果严格按照智能定义的解释,你也许会说所有程序都有智能。你可以这样考虑:智能定义的第一部分是理解事实、命题及关系的能力,计算机非常适合于做这类工作。例如,关系数据库可以存贮(理解)信息,接受查询(命题),并正如其名字所表示的那样,可以表达一个关系。当然,某些类型的信息,如视觉图形,要计算机来理解就比理解其它信息更为困难,但智能的定义并未要求理解必须采取某一特定的方式,它只要求能理解就行。这样,计算机通常所做的——收集、存贮和访问信息——就满足智能的第一个要求。

那么,数据库会用这些事实来进行推理吗?——这是智能的第二个要求。或许会,答案取决于你接受的推理的定义。如果数据库对信息的操作——如搜索、存贮、查询处理、文件管理等——可以称为推理,那么就可称数据库是一个智能程序,这也就意味着大多数计算机程序都是智能的。因为大多数计算机程序都是按合理的、符合逻辑的方式操纵信息,因此,这种形式的推理就有资格称作智能。

对许多人来说,这个结论是难以接受的,它意味着所有的程序都属于人工智能领域——而这并不那么正确。你的直

觉及你对特定的基于 AI 的程序的经验告诉你这里面是有差别的。但差别是什么？如果你想证明关系数据库不是一个会思考的程序，你或许会说关系数据库不可能是会思考的，因为你相信数据库程序所做的与人们的思考方式不一样。但是紧接着你就面临这样一个事实，那就是同样的任务由文件保管员来做时，很显然需要这个职员有智能。这样，就产生了一个矛盾：如果是数据库程序执行任务，就不算思考，而如果由人来做同样的工作，则人就在思考。作为一个人，你或许会想正是你的大脑，使你显得特别，只有人才会有认识性思维。你会认为高等哺乳动物会思考一些简单问题，甚至能在原始水平上进行推理，但人类能做的远远不止于此。但是，一台机器会思考，不管在什么水平，都是一个令人不愉快的想法。这样，当某些有才华的程序员编制出了聪明的程序，人们就趋于这么说，“实际上程序并不聪明，它只是做起来显得聪明罢了。”如果不这么说，它似乎是承认了人类在思维领域里的垄断的消失。

这里还可以从另一角度来看这个问题。如果一条狗能从前厅衔回其主人的报纸，人们就会说这条狗很聪明；如果一个一岁的小孩会做同样的事，人们甚至也会说这孩子很聪明。尽管建造一个能执行同样任务的由计算机控制的机器人不是一件困难的事，但是大多数人并不会仅仅因为机器人会取来报纸就说机器人有智能。这其中的原因是人们会说取报的机器人仅是一台机器，这台机器运行由程序员编制的程序，它并不是在思考任务，而仅仅是在执行任务（有些人也许会说这实际上是程序员在思考——只是在时间与空间上有一段距离罢了）。

上述机器人与报纸的例子带来一个不同的问题,但同样使我们回到了前述的事实,即人类并不知道他们是如何思考的。由于从前厅取回报纸的程序是容易被人们理解的,因此人们就会趋向于说程序不可能是智能的,因为它是可理解的!这有时也可称为魔术原则:从感性上说,大多数人都认为思维过程类似于魔术。因为他们不知道思维的过程,所以他们就错误地认为任何由人类制造并能被人类理解的装置都不可能是智能的,实质上,他们认为创造是不可能超过创造者自身的。

还有一个令人心烦的问题就是自由意志。在历史发展的进程中,思维总是与自由意志的概念联系在一起的:只有有意愿思考的实体才能进行思考。十七世纪的著名哲学家 *Descartes* 写下著名的哲学名句“*I think, therefore, I am,*”表明意识证实了存在。使这个问题令人不安的原因是,在上述例子中,狗和婴儿是选择了取报纸这个行动(相对于另外的事),但机器人是由程序控制这么做的——事实上,机器人必须去取报纸,因为它的程序就是这么编的——它不可能做其他事。然而计算机,这个表面上看来能决策的设备,是否会选择执行要做的事?毫无疑问,这个问题将会是二十一世纪的主要哲学与法律问题之一,这个问题也会使大批程序员分化。有一批程序员坚定地认为“机器就是机器”,计算机不可能有自由意志,因为它没有头脑,只有线路,这样计算机也就不可能选择要做的事,特别是进行思维,这是非常有力的论据。可是,其他持反对意见的程序员们的论据可能更有说服力。设想计算机正在砖场监视卡车上的衡器,当卡车装入的重量到了某一点,计算机就关闭砖头传送装置。这里是不是

计算机决定关掉传送装置的？是的！很清楚是计算机在控制着局势，当重量达到一定水平时它就作出了关掉传送装置的选择。如果计算机未做这个选择，那么是谁作的？这个论点的支持者认为计算机执行条件分支的能力就构成了其决策的能力。

计算机能否进行思考？正如上面举的例子所表明的，双方都有自己的意见，争论仍在继续。

关于机器智能问题，许多人都认为要确定机器是否能思考或程序是否有智能是不太可能的！然而，某些事例清楚地表明人可以使计算机的行为与人类似，这里的关键是某些程序显得具有智能（事实上，这就是 AI 的主题）。前一节所举例子的难点在于对智能的定义上。字典上对智能的定义所遗漏的是：“智能”这个问题实际上指的是人的智能，这样，按照这个定义要承认机器会思考或在计算机上运行的程序具有智能就很困难，因为大多数计算机在执行同样一件任务与人的做法并不一致，从另一方面说，如果把这个含义去掉，就很容易说有智能程序存在。如果你理解了这个区别，智能程序的定义就很直接了当了。要使一个程序有智能，就要使它执行起来象是有智能一样——这也就是说，它应能象人一样行动，而它的思维过程并不一定要总是与人的思维过程一致。下面给出智能程序的定义：

智能程序是一个在面临同一问题时能执行与人的行为相似的动作的程序，并不要求程序在实际解决问题时在具体方式上与人所做的一致。

程序不必要真象人一样思考，只要程序看起来象是如人一样在思考就行（即便是人，其思维方式也并不总是一样）。