

高级 TURBO PROLOG

人工智能程序设计

专家系统

问题求解

模式识别

机器人学

自然语言处理

Herbert Schildt 著
石油大学出版社

高级 TURBO PROLOG

Herbert Schildt 著

庄长淞 向逐骢 叶飞跃 译

马玉书 校

石油大学出版社

内 容 简 介

本书通俗易懂地向读者介绍人工智能各领域的概况，并通过大量程序实例引导读者利用 Turbo Prolog 实现各种人工智能技术。全书共分九章和两个附录：第一章人工智能概述，第二章问题求解，第三章专家系统，第四章自然语言处理，第五章视觉和模式识别，第六章机器人学，第七章机器学习，第八章逻辑与不确定性，第九章模拟人类，附录 A 与其他语言的界面，附录 B Turbo Prolog 工具箱。

本书是学习人工智能程序设计的一本既简洁又实用的教学参考书，书中的每个实例都可作为读者今后进一步开发完整智能系统的起点。可供大专院校计算机专业学生、各类程序设计人员和开发人工智能系统有兴趣的工程技术人员学习、参考。

高级 TURBO PROLOG

Herbert Schildt 著

庄长淦 向逐鹭 叶飞跃 译

马玉书 校

*

石油大学出版社出版

(山东省东营市)

新华书店发行

石油大学出版社激光排版室排版

石油大学印刷厂印刷

*

开本 850×1168 1/32 11.375 印张 296 千字

1991 年 7 月第 1 版 1991 年 7 月第 1 次印刷

印数 1—2000 册

ISBN 7-5636-0171-6/TP·07

定价：5.60 元

前 言

Turbo Prolog 语言的出现,标志着微型计算机应用的一个重大进展。作为具有问题求解和逻辑推理能力的程序设计的理想工具, Turbo Prolog 宣告了人类与计算机相互关系新时代的开始。Turbo Prolog 是以 Prolog (这是用于人工智能研究和应用的主要语言之一)为基础的。用其开发强有力的、直觉的程序,前景可以说是无量的。

《高级 Turbo Prolog》一书首先由 Osborne/McGraw-Hill 在 1987 年出版,其内容主要涉及两部分:一是向读者介绍人工智能领域的概况;二是借助大量程序例题,让读者利用 Turbo Prolog 来掌握人工智能技术。总之,这是一本既简洁又实用的介绍第五代语言工具及其使用的好书。

《高级 Turbo Prolog》一书修订本适用于 1.1 版本的 Turbo Prolog。新增添的附录 B 介绍了 Turbo Prolog 工具箱。该工具箱是一些常用例程汇集,其中包括了许多例题,这对了解 Turbo Prolog 与该工具箱都是很有用处的。

《高级 Turbo Prolog》一书由 Herb Schildt 编著而成。作者颇具名望,曾发表过大量有关程序设计的著作。发表的著作有《方便的 C 语言》、《高级 C》、《方便的 Modula-2》以及《高级 Turbo Prolog》等。

问题求解能力的新应用,及其对程序开发者的有利影响正日趋显著。而《高级 Turbo Prolog》一书对使这些应用变为现实是很有帮助的。

Philippe Kahn

序

本书利用问题求解方法来帮助读者提高 Turbo Prolog 程序设计技能。因为 Turbo Prolog 主要是一种用于人工智能的第五代语言，因此书中的应用特别显示了如何去实现机器智能。此外，为提高 Turbo Prolog 程序设计技能，读者还可以从书中获知有关整个人工智能领域的概貌，譬如：

- 专家系统
- 问题求解
- 逻辑与不确定性
- 自然语言处理
- 机器人
- 机器学习
- 视觉和模式识别

第一章介绍了人工智能的简要发展史，然后严格地阐述了什么是人工智能及其含义；第二章为有关问题求解的内容；第三章讨论了专家系统；第四章论述了自然语言处理；第五、六、七章分别讨论了视觉、机器人和机器学习的问题。最后两章是有关比较逻辑与不确定性的内容，如使计算机模拟人类以及将 Turbo Prolog 与已有的程序进行连接等。附录 A 是有关 Turbo Prolog 与其他程序设计语言之间的连接问题。附录 B 介绍了 Turbo Prolog 工具箱的内容。

书中除介绍各种概念外，其最重要的特点是每个应用都可作为开发一个完整系统的起点。例如，读者可以很方便地利用第三章的例题来开发一个专家系统，或利用第四章的例题来开发一个自然语言分析程序。

勿庸讳言，我很高兴编写这本书（说实话，对于我所编写的书，我都很喜爱，而对本书尤为偏爱）。当我还是一个大学生时，就很喜欢去探索开发机器智能。读研究生后，我便有机会在这一领域做了较为深入的研究（多数是使用 LISP 语言）。由于当时依靠编写人工智能应用软件是很难谋生的，因而我的兴趣只能停留在爱好上。同时，我发现根据 LISP 句法必须使用很多括号，这是十分麻烦和令人讨厌的。

我是在 1982 年第一次接触 Prolog 的。当即就被它那简洁的句法、精致的结构以及它那独特的功能所吸引，于是 Prolog 便立即成为我所选中的人工智能语言。基于上述原因，当我第一次得到一份 Turbo Prolog 软件时是十分兴奋的。

Turbo Prolog 之所以如此引人入胜，正是由于它将趣味性结合在程序设计之中。同时它也为众多的程序员打开了开发智能程序的大门。毫无疑问，作为一个程序设计者，我们都站在微型计算机革命第二次浪潮的起点上。我衷心期望读者能怀着与我写这本书时一样的心情来阅读它。然而重要的是，我更期望读者能够编制出一些更高水平的程序来。

—— HS

目 录

前 言

序

第一章 人工智能概述	1
§ 1 人工智能简史	2
§ 2 什么是智能程序	5
§ 3 人工智能的主要领域	8
第二章 问题求解：解的搜索	11
§ 1 表示方法和术语.....	11
§ 2 组合爆炸.....	13
§ 3 搜索技术.....	15
§ 4 深度优先搜索技术.....	18
§ 5 广度优先搜索技术.....	26
§ 6 加上启发式.....	29
§ 7 爬山搜索技术.....	30
§ 8 最小代价搜索技术.....	37
§ 9 搜索技术的选择.....	39
§ 10 寻求最优解	40
§ 11 农夫、狐狸、鸡和谷物	43
§ 12 回到丢失钥匙的问题	58
第三章 专家系统	60
§ 1 什么是专家系统.....	60
§ 2 专家系统如何工作.....	62
§ 3 建立通用专家系统.....	68

§ 4	知识工程	94
第四章	自然语言处理	97
§ 1	什么是自然语言处理	97
§ 2	自然语言处理的实现方法	98
§ 3	受限语言	99
§ 4	状态机 NLP 分析程序	100
§ 5	上下文无关递归下降 NLP 分析程序	112
§ 6	噪音剔除分析程序	128
第五章	视觉与模式识别	140
§ 1	滤波、反差和色调	140
§ 2	二维系统	141
§ 3	三维系统	143
§ 4	共同的识别问题	147
§ 5	二维模式识别	150
§ 6	综合性系统	188
第六章	机器人学	189
§ 1	机械手	189
§ 2	工业机器人	192
§ 3	自主机器人	195
§ 4	建立一个机器人模拟程序	196
§ 5	程序	199
第七章	机器学习	230
§ 1	两种学习	230
§ 2	如何学习类型描述	233
§ 3	知识表达	238
§ 4	实现命中——近距脱靶过程	241
第八章	逻辑与不确定性	251
§ 1	逻辑	251

§ 2 不确定性	272
第九章 模仿人类	295
§ 1 技巧	296
§ 2 有什么好处	297
§ 3 人和机器	297
§ 4 看病	299
§ 5 类人计算机的实现	318
附录 A 与其他语言的界面	320
附录 B Turbo Prolog 工具箱	330

第一章 人工智能概述

迄今为止，许多人还把人工智能（即 AI）看成是计算机科学的阴暗面。他们认为人工智能程序员，就象 Frenkenstein 试图创造生命一样，在试图创造思想。人工智能的研究者曾经被矛盾地认为既是计算机科学的“尖兵”，又是一批“狂人”。大多数职业程序员在必须说明什么是机器智能时，都尽力避免使用人工智能这样的字眼，却说成：“许多研究还有待去做”，“在不久的将来，将会有重要的发现”，“至今，在这一领域所取得的成就还很小。”等等。不过，现在有关人工智能的观点已发生了根本性的转变。

由于晶体管的发明，在短短的五年中，人工智能已从计算机科学的冷板凳上变为最热门的课题。这种迅速的变化有如下四个方面原因：一是专家系统的成功应用（这是第一个真正成功的商业化人工智能产品）；二是日本对人工智能的广泛宣传和赞助；三是出现了相当好的人工智能语言：Prolog；四是人工智能的时代已经到来。

随着 Turbo Prolog 的出现，人工智能程序设计已为所有程序员所接受。以前，完整的人工智能语言只能在大型计算机上使用。Turbo Prolog 使人们在微机上编写实用的人工智能应用程序已成为现实。由于优异的通用硬件的支持，如彩色、声音和图形。使 Turbo Prolog 能够很容易地开发出专业化的程序。

在学习本书时，读者将会完成许多 Turbo Prolog 所能完成的任务。通过开发人工智能领域的实际应用程序，将会提高编写 Turbo Prolog 程序的能力。人工智能领域有许许多多的课题，在下面几章中要讨论的是最重要和最常用的课题。但是在讨论这些课题之前，为得到一个总体印象，先回顾一下人工智能的历史，然后再说明究竟

什么是“智能”程序。在本章的最后，还将阐述人工智能的几个主要领域，这也正是本书后面各章要讨论的内容。

§ 1 人工智能简史

很难说出人工智能出现的确切时间。也许它的出现应该归功于发明了程序存贮式计算机的 A · M · Turning。记得，最初的计算机是一台实实在在的机器，当要解决不同的问题时，就必须接通不同的线路。Turning 的杰出贡献在于他把程序也当作数据存放在计算内存中，然后再来执行。这是现代计算机的基础。程序的存贮使得计算机功能可以通过运行一个新的程序很快、很容易地得到更新。这一存贮能力的潜在意义则在于：计算机也许能改变其自身的功能——即可以学习或思维。

不过，通常人们都认为人工智能起始于 1960 年前后。麻省理工学院的 John McCarthy 发明了第一个人工智能语言 LISP。LISP 和 Prolog 有许多相似之处，例如：它们都能方便地处理表结构并且都以递归作为子程序控制的主要形式。但是 LISP 缺少许多 Prolog 的优点，句法也不那么简洁。

人工智能这一名词的产生要归功于麻省理工学院的 Martin Minsky。他于 1961 年曾发表了一篇题为“步向人工智能”的论文（《无线电工程研究院论文集》第 49 期，1961. 1）。在六十年代人们普遍认为使计算机具有思维是完全可能的。在那十年中，人们看到了第一个会下棋的计算机；看到了第一个用计算机来完成数学定理的证明；看到了著名的 ELIQA 程序的诞生。ELIQA 是由麻省理工学院的 Joseph Weizenbaum 编写的，它很类似于一个心理学家。在做心理分析时，医生往往站在被动立场上，只是简单地听取病人的述说，而不是有意引导病人讲话。这正是计算机容易做到的一件事情。当时，该程序曾造成相当大的混乱。很多人提出：“计算机该不该这样使用？

计算机医生是不是真比精神病医生还高明？是否允许计算机来做诊断？”即使 ELIEA 的编制者 Weizenbaum 在他所写的《计算机的能力与人类的明智》（载于旧金山：W. H. “自由人和公司”，1976）中也不大信任自己的程序（请注意：六十年代也正是人人都惧怕自动化的时代，二十年前人们的感情有时是很难理解的）。

由于人工智能的明显成功，因此看起来编写一个程序使它具有类似人的智能，就是人工智能研究的最终目的。其实，事情并非如此简单。

在六十年代，人们还未能充分理解把这些成就综合成一个灵活的智能程序的困难（本书将讨论有关的一些理由）。其原因是当程序员试图提高某些程序的通用性时，常常需要更多的计算机资源而无法得到满足。例如，内存很快就用光了，或运行速度变得太慢。

到了七十年代中期，大内存计算机已变得很普遍，计算机运行速度也得到了迅速提高。既使如此，许多陈旧的 AI 方法，由于其本身效率太低，还是实现不了。要理解这一点，考虑一下简单的一组数的分类问题。如果你使用冒泡分类法，分类时间大约与 N^2 成正比，其中 N 是分类元素的数目。这意味着，假设将 10 个元素排序需 1 秒钟，将 100 个元素排序要花 100 秒钟……到一定的程度，你会发现一组数的排序时间将比人的平均寿命还长。不论计算机算得多么快，这种算法（时间与 N^2 成正比）将很快变得无能为力。更好的办法是改变分类算法，使其效率得到改善。例如使用快速分类法，它的分类时间与 $N^{1.2}$ 成正比——这是一个很有意义的改进。这两种曲线的差别如图 1-1 所示。正如新的分类算法能改进分类的时间一样，可以使程序执行得更快，或者更有效地利用内存的一些例程，使得许多人工智能问题得到解决。事实上，Prolog 语言的发明就是一个突破。在第二章中将要讲到这个问题。

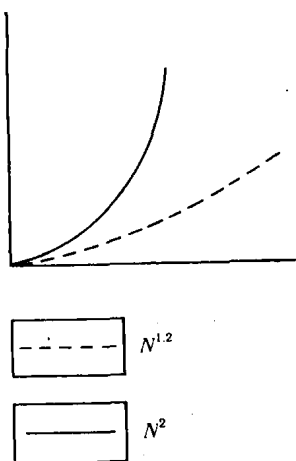


图 1-1 N^2 和 $N^{1.2}$ 曲线的比较

到七十年代末，人工智能在一些特定领域中获得了成功——如自然语言处理、知识表达、问题求解。从而到了第一个商业化产品——专家系统——的诞生时代。专家系统是含有某一领域知识的程序，当向它提出问题时，它能象一个专家一样地给予回答。第一个专家系统是 MYCIN，由斯坦福大学编制，是帮助内科医生诊断疾病的程序。本书后面将较深入地讨论专家系统。

八十年代以前，在美国，人们都忽视了七十年代发生的人工智能的一个最重要的事件。这就是 1972 年，法国马赛 Alain Colmerauer 创造了 Prolog 语言。与 LISP 一样，Prolog 也是用来帮助解决有关人工智能方面问题的语言。而它具有许多特点：如内部数据库、句法简洁。直到 1980 年，LISP 仍为美国人所选中的人工智能语言，而 Prolog 却广泛地应用于欧洲。这种状态一直保持到 1981 年日本宣布他们将用 Prolog 作为他们所炫耀的“第五代计算机”的基础，这才引起了美国程序设计者们的关注。

虽然从 1981 年以来，Prolog 已在美国逐渐普及，但大多都只在

大学中使用,这是由于微机上还没有一个完整的 Prolog。就是这些 Prolog,也没有一个完整的标准。因此, Turbo Prolog 的发行,也是人工智能历史上的一个里程碑。Turbo Prolog 给众多的程序设计者提供了在自己的个人计算机上开发人工智能程序的机会。同时也提供了一个标准的 Prolog,使程序员们可在各种各样的机器上使用。毫无疑问,这只是人工智能实际应用的开始,它的发展将会越来越迅速。

§ 2 什么是智能程序

在用 Turbo Prolog 进行人工智能研究之前,首先必须知道什么是智能程序,智能程序与“非智能程序”有什么区别。当然要讲清楚它们之间的关系是很不容易的,因为目前对此仍有争论。一种意见认为所有的程序都是智能的,而另一种意见却完全相反,认为所有的程序都是非智能的。这一节我们将为智能程序下一个合理的定义。

要确定什么是智能程序,首先得确定智能的含义。在一本字典中曾将智能定义为:“理解事实、命题和它们间关系的能力及对其进行思考的能力。”这个定义引出了这样一个问题:“思考是什么意思?”而问题就出现在这里。人们都能够说出他在想什么,但没有一个人能说得清楚他是如何思考的。事实也是如此,没有人能够真正理解另一个人是如何进行思考的(如果有人能够知道,那么让计算机去做到这一点就不是什么困难的事了)。

如果对这个定义只局限于字面的理解,那么很有理由说所有的程序都是智能的。智能定义的第一部分是对事实、命题和关系的理解能力。实际上,计算机正好适合于完成这一类任务。例如:关系数据库可以存贮(理解)信息,接受查询(命题),且正如其名字所隐喻也能表示关系。当然,有些类型的信息(如图形)计算机比较

难以理解，但是智能的定义并没有指定什么形式的信息，只要做到理解就可以了。这样计算机通常所做的工作：收集、存贮和访问信息正好满足了智能定义第一部分的要求。

那么数据库能否思考呢？这是智能定义的第二个要求。也许答案要取决于思考的定义是什么。当你认为数据库对信息的操作是思考时，那么数据库就是一个智能程序。这说明大多数计算机程序都是智能程序，因为大多数的计算机程序都是以合理的、逻辑的方法来处理信息的。因此这种形式的“思考”是能够称得上为智能的。

不过，这个结论是很难被接受的。由这个结论可以推断出所有的程序都属于人工智能的范畴。这是一个不正确的推论。直觉和经验告诉我们智能程序和一般的程序是有区别的。但这个区别是什么呢？当你试图证明不能接受关系数据库是一个会思考的程序时，你也许会说由于数据库程序做事的思考方法与人的思考方法不一样，所以它不可能是一个会思考的程序。但是你又面临着这样一个事实：数据库程序确实做了一个档案员需用他的智能才能做的相同的工作。因此出现了如下矛盾：数据库完成这项工作不是思考，而由人来完成这项工作就是思考。这个问题部分原因是由于人类的自尊心在作怪。作为人类，你也许认为大脑是人与动物分界的标准。人类唯一具有认识思维，一些较高级的哺乳动物可以有些简单的思想甚至可以在很低的水平上进行思考，而人类却是远远超过这一水平的。但当说到一台纯粹的计算机能够在任意水平上进行思考时，那就感到很不舒服了。因此当一些优秀的程序员编出了一个杰出的程序时，潜意识却说：“哼，这算不了什么，只不过看起来象是挺好的罢了。”如果不这么说，就等于承认思维能力并非人类所独有。

让我们从另一个角度来看这个问题，一只训练有素的狗，如果它能够每天从院子前门口叼回主人的报纸，那么你会说这是一只很杰出的狗。同样一个一岁多的孩子也能做相同的事，那么你也会说这是一个很聪明的孩子。而制造一个由计算机控制的机器人来做

这件工作也是相当容易的，但绝大多数人都不会认为这个机器人具有智能。他们会说这个取回报纸的机器人只不过是在运行由某个程序员编写的程序的机器罢了。机器人并不思考它所做工作，只是完成一些动作而已（也有人认为是设计机器人的程序员在一定的时空以外在思考）。

机器人和取报纸的这个例子同样引出了一个老问题——不知道人们怎样进行思考。因为从门口取回报纸这样的程序是很容易编的，所以一般都认为它是可以理解的，不是智能程序。这就是有时被称之为魔术的原因，因为大多数人在感情上都觉得思考有些象魔术。由于人们不理解思考的过程，因此他们倾向于得出这样的错误假设：即任何设备都是由人制造的，也能为人所理解，因而不可能具有智能。实际上大多数人都相信创造出的任何东西都不可能超过创造者。

另外，还存在着自由愿望的问题。历史上，思想一直是与自由愿望的概念联系在一起的。即思考只能由愿意思考的实体来完成。事实上，十七世纪的哲学家 Descarts 曾声称正是思考才证实了他的存在。他曾写过一句至理名言：“我思考，因此我存在。”正是他造成了机器人和取报纸这个例子的困境。很明显，小孩和狗是自己选择去取报纸（他或它也可选择去干别的事），但是机器人是由程序控制它去干这件事的。因为程序让它去取报纸，它必须去，而不能去干别的事情，那么计算机做为一个确定的设备能否有选择地去做某件事情呢？毫无疑问，这个问题将会成为二十一世纪主要的哲学和法律论题。它也使全体程序人员分成了两个派别。一部分程序员坚信：“机器就是机器”，它没有自由愿望，没有思维只有电路，因此计算机不可能有选择地去做任何事情，特别是思考；而意见相反的另一部分程序员的见解也许更有说服力。假设一台计算机正在监视给卡车装砖，当达到一定的重量时，计算机关掉运送砖块的传送带。它有没有做出决定呢？很清楚计算机处于一种控制地位并且在重量上达到了某个水平时，选择去停掉传送带。如果不是计算机做的决定，

那么是谁做的呢？这个观点的拥护者声称：“计算机完成条件分枝的能力也正是做出决定的能力。”

说到这里，可能许多人都相信不可能使一个程序具有智能。首先，计算机是一个确定的设备；第二，存在是否具有自主权的问题；第三，既然几乎没有人能够知道思考是怎样进行的，那么怎样才能使计算机思考呢？其实，正如所有的程序都是智能的这种结论不恰当一样，上述结论也是不合适的。

前面提到，问题的关键取决于对智能的定义。字典上的定义忽略了如下事实：“智能”这个词蕴涵着人类智能的意思。这种蕴涵的联想使人们很难承认机器可以思考，或计算机上运行的程序可能具有智能。因为大多数的程序完成工作的方式与人所使用的方式并不一样，因此，一旦去掉了这个蕴涵的意思，那就很容易说所有的程序都是智能的。如果你理解了这种区别，智能程序的定义就很好下了。作为一个智能程序只需它具有智能的行为——即象人类那样，它的思考过程并不需要完全象人类。由此可得出智能程序的定义如下：

当一个智能程序面对相同的问题时，必须表现出与人类类似的行为。它实际解决或试图解决问题的方式并不需要与人类完全一致。

注意，智能程序并不需要象人类一样思考，但必须象人类一样行动（其实即使人类也不是以同样的方式进行思考的）。

因此，智能程序具有与人类智能类似的行为，而非智能程序却不是如此。

§ 3 人工智能的重要领域

人工智能的研究领域很多，下面是常见的和主要的几个领域：

- 搜索（求解）
- 专家系统
- 自然语言处理