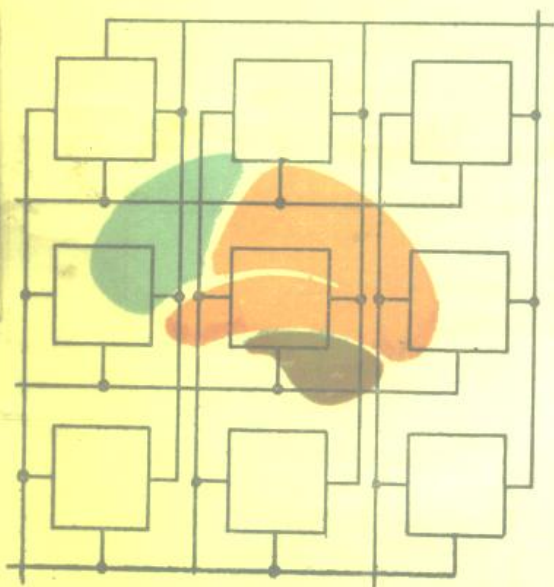


IBM-PC 机编译型 PROLOG 语言

李卫华 周祥和 刘灼光 编译

人·工·智·能·丛·书

武汉
大学
出版
社



IBM-PC 机编译型 PROLOG 语言

李卫华 周祥和 刘灼光 编译

武汉大学出版社

责任编辑 曾慕贞 陈礼琰

IBM-PC 机编译型 PROLOG 语言

李卫华 周祥和 刘灼光 编译

武汉大学出版社出版

(武昌 珞珈山)

新华书店湖北发行所发行 武汉大学印刷厂印刷

787×1092毫米 1/32 10.125印张 227千字

1987年8月第1版 1987年8月第1次印刷

印数: 1—8,000

ISBN 7-307-00010-5/TP·1

统一书号: 13279·38 定价: 2.35元

内 容 简 介

凡内存存在640KB 以上的 IBM-PC 机及其兼容机 均可运行 美国 1986 年研制成功的高速强功能中英文兼用的编译型 Turbo PROLOG 语言。无论是功能上还是运行速度上, 此编译程序都是当今最强有力的最成熟的 PROLOG 系统之一。用之可生产任一实际应用的原型, 实现动态关系数据库, 建造专家系统和专家系统工具, 完成语言翻译, 形成同现有软件的自然语言交互, 解方程、求微积分, 设计智能软件, 管理工业过程以及作情报检索等。

本书根据美国 Borland 国际有限公司 1986 年出版的 Turbo PROLOG 一书以及配书软件编译。书中通过大量通俗易懂富有趣味性的程序实例详介了该编译型 PROLOG 语言内含的关于输入输出、文件管理、窗口显示、类型转换、数据处理、系统交互、模块操作、动态追踪、图形声音等方面的百余个内部谓词, 算术运算函数, 指数函数, 三角函数, 全屏编辑以及同 PASCAL 语言、FORTRAN 语言、C 语言、汇编语言等非 PROLOG 型软件的交互机制。

该书既适于不了解 PROLOG 语言的新手学习, 又适于熟悉 PROLOG 语言的行家参考。新手应先读前两章, 以了解 PROLOG 语言的一些优点, 知晓如何将程序键入, 如何编译已键入程序, 如何执行已编译程序, 如何使用编译型 PROLOG 独有的程序设计环境。然后仔细阅读第三至第十章, 上机测试各章中的程序实例。行家可先读第二章和第十一章, 以了解系统的基本操作并找出本编译型 PROLOG 语言同其它 PROLOG 实现的差别。新老手都应阅读第十二章, 以全面概观编译型 PROLOG 语言。

编译者非常感谢翁寄遥、李子明、何学平、郑红兰、陈卉等人为此书的编译工作所付出的辛勤劳动。

前 言

人工智能是计算机科学研究的重要领域，它涉及专家系统、认知模型、自动推理、自动程序设计、智能机器人、机器学习、自然语言理解和感知等学科。随着这些学科的基本理论、基本技术的深入研究和广泛应用，人工智能不仅正在深刻地影响着计算机科学的其它领域，诸如数据库、软件工程、程序设计方法学的研究方向等，而且正在极大地促进教学、医学、经济管理学、决策科学以及情报学的深入发展，并带来广泛的社会效益。为满足广大科技人员、计算机工作者、研究生、本科生系统学习人工智能的需要，我们计划编辑出版一套人工智能丛书。它包含：《IBM-PC 机编译型 PROLOG 语言》、《自动推理引论及其应用》、《智能机器人》、《自然语言理解》、《人工智能方法与技术》、《大脑与行为模拟》等。

由于我们水平所限，书中错误和缺点在所难免，请广大读者批评指正。

人工智能丛书编写组

一九八六年十二月五日

目 录

第一章 引 言	(1)
第二章 启 动	(5)
§2.1 主菜单	(5)
§2.2 例子程序	(6)
§2.3 编辑程序	(9)
§2.3.1 基本操作	(9)
§2.3.2 字块操作	(9)
§2.3.3 搜索替换	(11)
§2.4 追踪	(12)
§2.5 改变窗口设置	(14)
§2.5.1 临时改变窗口	(14)
§2.5.2 保留窗口构型	(15)
第三章 例 子	(16)
§3.1 变量	(18)
§3.2 对象与关系	(19)
§3.3 领域与谓词	(20)
§3.4 复合目标	(22)
§3.5 无名变量	(22)
§3.6 回溯	(23)
§3.7 not 的用法	(26)

§3.8	注解	(29)
§3.9	例子	(29)
	练习	(33)
第四章	领域、对象与表	(37)
§4.1	自由与约束变量	(37)
§4.2	标准领域类型	(39)
§4.3	复合对象	(42)
§4.3.1	复合对象的领域说明	(43)
§4.3.2	复合对象的层次表达	(46)
§4.4	递归与表	(48)
§4.4.1	递归对象	(51)
§4.4.2	表	(52)
§4.4.3	陈述式与过程式程序设计	(55)
	练习	(57)
第五章	搜 索	(61)
§5.1	合一	(61)
§5.2	搜索的控制	(64)
§5.3	阻止回溯的截断元素	(70)
	练习	(74)
第六章	算术运算、输入输出和调试追踪	(75)
§6.1	算术运算	(75)
§6.1.1	算术表达式的求值顺序	(75)
§6.1.2	比较	(76)
§6.1.3	算术函数与谓词	(80)
§6.2	输入输出	(82)

§6.2.1 写	(82)
§6.2.2 读	(86)
§6.3 调试追踪	(89)
练习	(91)
第七章 窗 口	(94)
§7.1 窗口属性	(94)
§7.2 窗口设置	(96)
§7.3 窗口读写	(97)
§7.4 窗口游戏	(103)
§7.5 DOS 交互	(108)
第八章 图形和声音	(112)
§8.1 图形	(112)
§8.2 声音	(120)
第九章 文件和串	(123)
§9.1 文件控制	(123)
§9.2 串	(130)
§9.3 类型转换	(134)
§9.4 findall 和 random	(137)
第十章 应 用	(140)
§10.1 专家系统	(140)
§10.2 路径问题	(146)
§10.3 洞穴探宝	(148)
§10.4 硬件模拟	(152)

§10.5	梵塔问题	(153)
§10.6	音节划分	(158)
§10.7	皇后问题	(161)
§10.8	键盘利用	(164)
§10.9	美国地理	(169)

第十一章 用户指南

(219)

§11.1	系统概貌	(219)
§11.2	语言元素	(220)
§11.2.1	名字	(220)
§11.2.2	程序段	(222)
§11.2.3	领域说明	(223)
§11.2.4	谓词说明	(225)
§11.2.5	子句	(226)
§11.3	存贮管理	(228)
§11.4	编译指令	(230)
§11.4.1	check_cmpio	(230)
§11.4.2	check_determ	(231)
§11.4.3	code	(231)
§11.4.4	diagnostics	(232)
§11.4.5	include	(232)
§11.4.6	nobreak	(233)
§11.4.7	nowarnings	(234)
§11.4.8	project	(234)
§11.4.9	trace 和 shorttrace	(234)
§11.4.10	trail	(236)
§11.5	动态数据库	(237)
§11.5.1	数据库说明	(237)

§11.5.2	事实处理	(239)
§11.5.3	文件型数据库	(241)
§11.6	参量模式	(245)
§11.7	程序设计风格	(247)
§11.7.1	堆栈的考虑与尾递归删除	(247)
§11.7.2	fail 谓词的应用	(251)
§11.7.3	截断元素的设置	(252)
§11.7.4	参考领域	(252)
§11.8	可执行文件的生成	(257)
§11.9	模块程序设计	(258)
§11.9.1	工程	(258)
§11.9.2	全局领域和全局谓词	(259)
§11.9.3	模块的编译和连接	(260)
§11.9.4	例子	(261)
§11.10	同其它语言的接口	(262)
§11.10.1	外部谓词说明	(263)
§11.10.2	调用约定与参量	(263)
§11.10.3	命名约定	(264)
§11.10.4	同汇编语言的交互	(264)
§11.10.5	同 C、PASCAL 和 FORTRAN 的交互	(267)
§11.10.6	低级支撑	(268)
§11.11	编辑程序的随机调用	(271)
§11.12	列目录与格式输出	(272)
第十二章	参考手册	(274)
§12.1	配书软盘	(274)
§12.2	启动文件	(279)

§12.3 主菜单.....	(280)
§12.3.1 运行命令Run	(280)
§12.3.2 编译命令Compiler 和选择菜单Opitons...	(281)
§12.3.3 编辑命令Edit	(282)
§12.3.4 文件菜单Files.....	(288)
§12.3.5 设置菜单Setup	(290)
§12.4 退出命令.....	(292)
§12.5 语言语法.....	(292)
§12.6 系统限制.....	(297)
§12.7 出错信息.....	(297)
文献	(305)
索引	(306)

第一章 引 言

过去十多年来,硬件销售价格每四年减少一半。但是,软件成本却逐年增加,而且已占系统预算的一大部分。1970年,软件开销大约占系统总成本的10%,1975年占50%,1985年占80%。如此迅速增加的成本迫使人们开发新的程序设计工具,研究将大量工作交机器完成的方法,以便更容易、更迅速、更便宜地开发程序。

PROLOG 就是这类工作的多年的研究结晶。PROLOG 版本于七十年代初期由法国马赛大学 Alain Colmerauer 等人开发作为一种方便的逻辑程序设计工具。它比 PASCAL、BASIC 之类的其它著名程序设计语言能力强、效率高。例如,用 PROLOG 来写求解一给定应用的程序比用 PASCAL 写要少花十倍的程序行数。

目前,在人工智能应用程序设计中,在专家系统的开发中,PROLOG 是一种非常重要的工具。很多著名的专家系统外壳都用 PROLOG 编写,如 ESP/Advisor、APE 和 XI 等。适于更多用户掌握的命令和智能程序是 PROLOG 日增普及的另一理由。一旦交给 PROLOG 一些必要的事实和规则,陈述式语言 PROLOG 就可使用演绎推理去自动求解一些程序设计问题。与此相反,过程式语言 PASCAL、BASIC 和其它系统程序设计语言的程序员必须一步一步地告诉计算机如何求解。正因为 PROLOG 是陈述式而不是过程式语言,使 PASCAL 和 BASIC 中诸如循环之类的错误

来源在 PROLOG 中从一开始就堵绝。而且，PROLOG 还教程序员对问题作结构良好的描述，因此在实践中，PROLOG 可作为一种规范工具。

虽然 PROLOG 使程序设计非常容易，但它对计算机也作了许多要求。本书介绍的编译型语言 PROLOG—Turbo PROLOG 实现在 IBM-PC 机及其兼容机上，它能力极强，内存要求很少，是一门十分成熟的编译程序。它有一个菜单型交互式程序设计环境，能作普通算术运算、三角函数、指数对数函数计算、逻辑推理，还有图形和声音功能。由 Turbo PROLOG 产生的编译程序执行起来非常快，且不象其它解释型 PROLOG 语言那样吞占内存。

1982年，日本宣布了一个宏伟的设计与生产第五代计算机的国家工程计划，计划中选用 PROLOG 作为基本系统语言，它对应于当前的结构中的汇编语言。Turbo PROLOG 能在价值2000美元的计算机上运行。1984年曾用本系统的早期版本同日本第五代计算机原型机中的 PROLOG 语言作过比较，结果是 Turbo PROLOG 的执行速度快。

Turbo PROLOG 有着广泛的实际应用：

- 生产任一实际应用程序的原型。某种萌芽思想可以很快实现，基于此思想的模型可以“现场”测试。
- 控制和监督工业过程。Turbo PROLOG 能完全接近计算机的输入/输出端口。
- 实现动态关系数据库。
- 变换语言，或作人类语言转换，或从一种程序设计语言转换成另一种程序设计语言。例如，将 BASIC 语言转化为 UNIX 操作系统下的 C 语言。
- 构造同现有软件的自然语言交互，使得更为广泛地利用现

有系统。用Turbo PROLOG 很容易将窗口包括在这种交互之中。

- 构造专家系统和专家系统工具。
- 构造符号处理包，以解方程、求微积分等。
- 构造定理证明与人工智能包，并用 Turbo PROLOG 的演绎推理能力去测试不同理论。

Turbo PROLOG 与系统程序设计语言有许多差别：

- Turbo PROLOG 是陈述性的。 Turbo PROLOG 程序不规定计算机求解问题的每个步骤，而是由一问题的描述所组成。此描述有三个部分，其中第一部分和第二部分对应于 PASCAL 程序的说明部分：

1. 待解问题中牵涉到的对象名字、对象结构以及对象所属领域。
2. 存在于对象之间的关系名。
3. 描述这些关系的事实与规则。

Turbo PROLOG 程序中的描述用来规定在给定输入数据与由那一输入生成出来的输出之间存在什么关系。Turbo PROLOG 使用事实和规则。Turbo PROLOG 程序不作初始说明，而由一张逻辑语句表组成，格式可为诸如“今天下雨”这样的事实，也可为“如果天下雨，且你忘了带伞，那么你会淋湿”之类的规则。Turbo PROLOG 可作演绎。给定事实：

约翰喜欢玛丽

汤姆喜欢萨姆

和规则：

如果汤姆喜欢 X，那么珍妮特喜欢 X

Turbo PROLOG 便可推出结论：珍妮特喜欢萨姆。你可给 Turbo PROLOG 程序一个目标，如：

找出喜欢萨姆的每一个人。

Turbo PROLOG 将用其演绎能力找出问题的全部解。

- Turbo PROLOG 程序的执行是自动控制的。在执行一道 Turbo PROLOG 程序时，系统试图找出满足给定目标的所有可能的值集。在执行期间，结果可显示，或提示用户键入某些数据。一旦找到某个解，Turbo PROLOG 就用回溯机制使 Turbo PROLOG 重新评价任何假定，以弄清某些新的变量值是否产生新解。
- Turbo PROLOG 的语法简明扼要。学习 Turbo PROLOG 的语法比学习其它复杂传统程序设计语言的语法容易得多。
- Turbo PROLOG 能力很强。同 PASCAL 相比，Turbo PROLOG 是一门更高级的语言。如前所述，用 Turbo PROLOG 编写的求解问题程序较用 PASCAL 编写的求解同一问题的程序小十倍。其所以这样，是因为 Turbo PROLOG 有一内部模式匹配机制，以及一种简单和有效的处理递归结构的方法。
- Turbo PROLOG 是编译程序，它允许交互式程序开发。程序员可在任何时刻测试一程序的个别片段，改变程序的目标，且不必附加新代码。这相当于能试用 PASCAL 程序中的任一过程，即使在程序编译之后也如此。

以上扼要地叙述了 Turbo PROLOG 的特征。随着你更加深入地学习这本书，更具体地练习编写程序，你就会发现 Turbo PROLOG 的更多能力。

第二章 启 动

本章描述 Turbo PROLOG 系统的基本操作，包括如何让系统回溯，如何使用菜单，如何运行 Turbo PROLOG 程序，以及如何用 Turbo PROLOG 编辑程序建立一程序文件。

第十二章给出了放在配书软盘上的一张文件表，以及用 Turbo PROLOG 系统时需要的文件。Turbo PROLOG 在 IBM-PC 机及其兼容机上运行。如果你不满意某些默认值（如显示颜色、窗口位置大小等），可用下推菜单改变之。

§2.1 主菜单

将系统盘放入默认驱动器后，敲 PROLOG 便可启动系统。这时，你可在屏幕上见到版本信息，Turbo PROLOG 在你所用计算机上的构型，以及提示信息。

照提示信息按空格键后，Turbo PROLOG 在屏幕顶部显示主菜单，在屏幕中部显示四个系统窗口，在屏幕底行显示状态行。

主菜单向你显示命令和下级菜单。为选一个下级菜单，你可按一个发亮的大写字母，也可先用左右向箭头键将亮条移到所选菜名处并按回车键。各窗口的作用在本章中介绍。屏幕底行的状态文本描述了功能键或光标键的用法，这些功能键的意义依追踪、编辑或运行状态的不同而异。

§2.2 例子程序

现用一道简单的 Turbo PROLOG 程序来说明如何建立、运行和编辑 Turbo PROLOG 程序:

```
predicates
    hello
goal
    hello.
clauses
    hello:-makewindow(1,7,7,
                      'My first program',
                      4,56,10,22),
        nl, write('Please type your name '),
        cursor(4,5),
        readln(Name), nl,
        write('Welcome ', Name).
```

先用左右向箭头将亮条移至单词 **Edit**、再按回车键,或者单按 **E** 键以选编辑状态。此时编辑窗口发亮,而在屏幕底行的状态文本反映了功能键的新意义。

为弄清如何改错,故意将上述程序第一行敲为:

```
predivates
```

为将字母 **v** 改为 **c**, 用左向箭头将光标移至错误字母 **v**, 然后按 **Del** 键。仔细看看编辑窗口内的变化, 然后按 **c** 键便可看出, 错误业已改正。接着键入上一程序文本的头八行, 但在敲第 6 行至第 8 行时, 别按回车键。

当键入行 **makewindow...** 的尾部时, 该行首部文本将卷