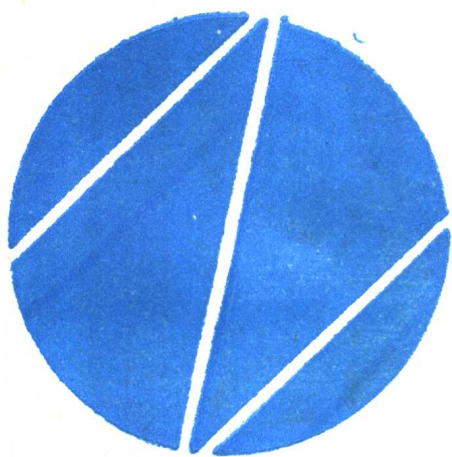


知识工程及其应用

李冠英 李吉桂 吴伟民 编 著



18
34

华南理工大学出版社

知识工程及其应用

李冠英 李吉桂 吴伟民 编著

华南理工大学出版社

内 容 提 要

知识工程是研究知识获取、知识表达、知识接口、知识应用等知识信息处理的综合性科学、它是计算机应用领域的扩展,是人类应用计算机的又一里程碑。

本书第一章简述知识工程发展史;第二章为便于读者阅读有关例题,简要介绍PROLOG语言;第三章介绍问题求解的基础知识;第四章至第七章介绍知识表示;第八章介绍知识库的基本概念、方法和维护;第九章介绍知识工程的主要应用内容——专家系统;第十章介绍了我们近年来的工作,以供读者参考。

本书深入浅出,有计算机基础知识的读者均能顺利阅读。通过本书阅读不但可以掌握知识工程的基本理论、方法和技术,而且能窥见其当前发展的主要动向。本书可作为大学计算机科学高年级学生、研究生教材或教学参考书;也可供科技人员(特别是产业界的计算机专业人员)参考。

知识工程及其应用

李冠英 李吉桂 吴伟民 编著

责任编辑 黄敏

华南理工大学出版社出版发行

(广州 五山)

各地新华书店经销

广东省韶关新华印刷厂印刷

开本 787×1092 1/16 印张 9 字数 213千

1989年12月第1版 1989年12月第1次印刷

印数 1—2000册

ISBN 7—5623—0137—9/TB·20(课)

定价 1.90元

前 言

自1956年美国学者John Mccarthy在一次专题讨论会上提出人工智能(Artificial Intelligence)一词以来,人工智能作为一个学科正在蓬勃地发展。其中一个十分引人(特别是产业界)注目的方向,就是美国斯坦福大学的E. A. Feigenbaum于1977年提出的“知识工程”(Knowledge engineering)。所谓知识工程就是建造知识系统的基本理论和基础技术。当前其理论、方法论以及开发技术正得到急速的发展。可以说,专家系统的最早教科书《Building Expert System》(Frederick, Hayes—Roth, 1983)的发表,标志着知识工程这一人工智能的重要领域已开始成熟并已进入了工业应用的新阶段。

知识工程研究的主要课题是知识表示、知识使用和知识获得等。知识表示就是将关于问题领域的知识和专家所掌握的经验知识记为计算机能够识别的记号;知识使用就是对于给定的事实和目标,选择必要的知识,依赖于借助推理生成的假设进行问题求解;知识获得就是分析存在于专家、说明书、操作手册、维护报告等所谓知识源的知识,抽出必要的部分和将其存于所谓知识库中并实行管理等。

本书第一章简述知识工程发展的经纬;第二章为了便于读者阅读有关例题,简要介绍了常用的PROLOG语言;第三章介绍问题求解的基础知识;第四至七章介绍知识表示;第八章介绍知识库的基本概念、方法和维护;第九章介绍知识工程应用的主要内容——专家系统;第十章介绍了我们近年来的工作,以供读者了解专家系统产业化的一斑。

本书深入浅出,具有计算机基础知识的读者应能顺利地阅读本书。通过阅读本书不但可以掌握知识工程的基本理论、方法和技术,而且能够窥见其当前发展的主要动向。为了便于读者进入这些领域,书末附有较为详细的文献目录。本书可作为大学计算机本科或研究生的教材或教学参考书,也可供科技人员(特别是产业界的计算机专业科技人员)参考。

限于篇幅,本书有些地方没有具体指出引用内容的出处,在此特向有关编著者致意。区益善教授在百忙中认真审阅了本书并提出了宝贵意见,在此深表谢意。

由于知识工程这一学科发展迅速,本书难免挂一漏万,不当之处恳望读者指正。

编著者

1989年3月

目 录

第一章 绪 论	1
§ 1.1 知识工程的发展	1
§ 1.2 知识系统和知识工程	3
第二章 PROLOG语言	6
§ 2.1 PROLOG语言的基本概念	6
§ 2.2 基本语法	8
§ 2.3 PROLOG的基本算法.....	11
§ 2.4 内部谓词	14
第三章 基于知识的问题求解	19
§ 3.1 问题求解中知识的典型运用	20
§ 3.2 搜索空间与问题表示	25
§ 3.3 基本的启发式搜索方法	32
第四章 知识的逻辑表示	55
§ 4.1 知识表示概述	55
§ 4.2 谓词逻辑	56
§ 4.3 归结原理	58
§ 4.4 归结方法的控制策略	60
§ 4.5 逻辑表示的优缺点	63
第五章 产生式系统——知识的规则表示	64
§ 5.1 产生式系统的结构	64
§ 5.2 置信分解的策略	65
§ 5.3 产生式系统的推理方向	66
§ 5.4 产生式系统的特征	69
第六章 知识的语义网络表示	71
§ 6.1 语义网络	71
§ 6.2 性质的继承	72
§ 6.3 语义网络的推理过程	73
§ 6.4 状态框架	75
§ 6.5 语义网络的特征	75
第七章 知识的框架表示	78
§ 7.1 框架的数据结构	78
§ 7.2 槽的扩张与过程附加	80
§ 7.3 框架结构的推理机构	82

第八章	知 识 库	84
§ 8.1	知识、知识表示与知识库	84
§ 8.2	知识库的建造	86
§ 8.3	知识库运行、测评和维护	97
§ 8.4	元知识的使用	99
第九章	专家系统	102
§ 9.1	概述	102
§ 9.2	专家系统的结构	104
§ 9.3	专家知识的获取	106
§ 9.4	推理机制	111
§ 9.5	不精确推理	115
§ 9.6	专家系统的评价	125
§ 9.7	专家系统的工具系统和程序设计语言	128
第十章	专家系统的实例	133

第一章 绪 论

1977年,在美国麻省理工学院召开的第五届国际人工智能协会(International Joint Conferences on Artificial Intelligence, 简称IJCAI)上,斯坦福大学的Edward A. Feigenbaum教授提出了“知识工程”这一概念,他把以积极地使用人类具有的经验知识为前题的、人工智能应用的有关研究领域称为知识工程(Knowledge Engineering, 简称KE)。1983年1月,美国著名的知识工程学者Frederick Hayes-Roth等发表了世界上第一本专家系统的著作《Building Expert System》,标志着知识工程这一人工智能的重要领域的成熟,并标志着工业知识工程的诞生

§ 1.1 知识工程的发展

知识工程的发展,一般可表示为图1-1所示的几个阶段。

1956年, Shamon, Mccarthy, Minsky, Simon和Newell等人,将利用计算机实现智能功能的有关研究领域称为人工智能(Artificial intelligence),并从事了开创性的研究工作。当时,有代表性的范例是用计算机求解“猜谜”和“对策”等初等问题。

进入19世纪60年代,从对“博弈”与“猜谜”等问题的研究中发展起来的“搜索技术”继续得到发展;另一方面,对以GPS (General Problem Solver)的通用问题求解程序的研究,也开始盛行起来,这一时期的另一个特征是出现了面向人工智能的程序设计语言——由Mccarthy 在1960年开发的LISP语言,至今仍是使用较多的一种人工智能程序设计语言。

19世纪60年代中期的主要研究课题是定理证明和编制机器人的操作程序,由J·A·Robinson于1965年提出的“归结原理”,将定理证明研究领域向前推进了一大步。

1967年, Newell提出产生式系统,而Quillian在1968年提出了“语义网络”的概念。这二项工作把一般解题策略与专业知识、实践经验总结起来,研究专门的求解,把专门领域专家知识、推理策略和解决问题的方法存贮到计算机中,使计算机能象专家一样解

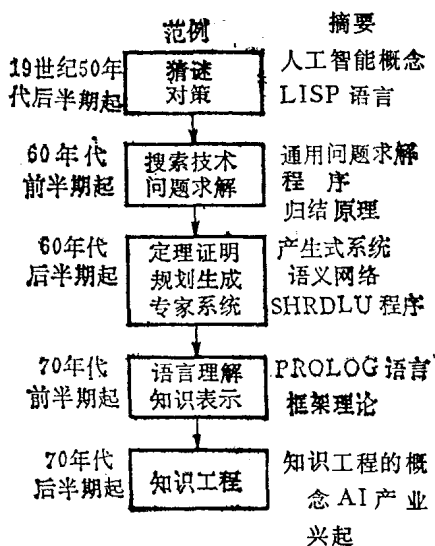


图1-1 从人工智能到知识工程

决有关领域的问题，构成“专家系统”。1965年，E·Feigenbaum 研制成世界第一个专家系统DENDRAL，它能根据质谱数据，识别化合物的分子结构。

进入19世纪70年代，特别是1972年，由Winograd创建了SHRDLU程序。该程序在由各种形状、大小、颜色组成的积木世界中操作，根据英语指令变换积木和实施关于积木世界的变化。此后，对语言理解和知识表示的研究与讨论相当活跃，给人工智能带来了新的转机。这一时期普遍地认为，就智能的本质而言，知识的获取与使用是不可缺少的。这一时期的另一特征是 Colmerauer 等人在1972年提出了逻辑为基础人工智能程序设计语言——PROLOG。

知识表示的研究高峰是从19世纪70年代中后期开始的，以Minsky的框架(frame)理论为代表的新的知识表示形式相当流行。另一方面，基于知识应用的专家系统的研究不断取得进展。如1976年，美国斯坦福大学的E·Shortiffe研制出专家系统 MYCIN，能根据细菌感染进行抗菌素处方。这样，进一步促使人工智能的研究从理论转向应用，从基于推理模型转向基于知识模型，使智能系统的研究转入具体化，这是人工智能研究的一个转折。于是，1977年E·Feigenbaum教授提出“知识工程”的概念。

从上述事实可以看到，知识工程是人工智能的一个专门领域发展起来的。目前已广泛地应用于各个部门，其产品已逐步进入到商品化阶段，表1-1列出了目前知识工程应用的一些实例。

表1-1 知识工程应用的实例

应用领域	系统名称	开 发 者	注 释
有机化合物 物辨识	DENDRAL	Stanford大学	化合物结构推理系统
	META-DENDRAL	ditto	成份分析推理系统
	SU/P	ditto	蛋白质X射线数据解析系统
	MULGEN	ditto	分子遗传学查询系统
技术咨询	SACON	Stanford大学	结构解释系统
	SU/X	ditto	信号分析和理解系统
	KNOBS	Mitre公司	飞机识别仿真器
诊断治疗	MYCIN	Stanford大学	血液传染病与脑膜炎诊断治疗系统
	TEIRESIAS	ditto	构成、维护、使用大规模知识库的中间知识引用系统
	PUFF	ditto	肺功能诊断系统
	MECS-AI	东京大学医院	心率不全诊断、治疗系统
	CASNET	Rutger大学	白内障诊断治疗系统
	EXPERT	Rutger大学	甲状腺等疾病治疗系统
	PIP	MIT	肾脏病诊断系统

续 表

应用领域	系统名称	开 发 者	注 释
问题应答	BUILD, SHRDLU	MIT	积木世界中的机器人对话系统
	ELIZA	ditto	精神分析快速响应系统
	GUS	Xerox公司	旅游规划用快速响应系统
	SCHOLAR	MIT	学习地理的计算机辅助教学系统
	STUDENT	MIT	算术解题系统
	NEWTON	MIT	物理解题系统
	ARIMAS	大阪大学	算术几何解题系统
文章解释 与理解	SAM	Yale大学	对话理解系统
	MARGIE	Stanford大学	英文理解系统
	LINGOL	MIT	英文解析系统
	EXPLUS	ETL	日文解析系统
	PLATOV	京都大学	日文解析系统
	MILISYT	ditto	日语理解系统
声音理解	HEARSAY-I	CMU	声音理解系统
	LUNAR	Harvard大学	
情报检索	PLANES	Illinos大学	情报检索系统
	REQUEST	IBM	
	RADDER	SRI	
翻 译	METED	Montreal大学	翻译系统

§ 1.2 知识系统和知识工程

一般来说,知识系统是一个以电子手段收集、贮存、传播、推广和应用知识的系统,专家系统是知识系统的一个典型例子。因此,可以将各种基于知识的系统简称为知识系统。

人类知识中,有相当大量的一部分是由一些专门技能片段构成的,运用知识时需要用有效的方法将各个片段组成有效的实体。知识系统将那些片段的 知识 收集在知识库中,通过访问知识库以实现 对某些问题作出解答。一个知识系统的结构如图1-2所示,

其中知识库、推理机和用户接口是知识系统的主要部分。

知识系统的研究，逐步形成了一门以知识为研究对象的一套理论、技术、方法和工具，将具体知识研究中共同的基本问题抽象出来，形成具体研制开发各类知识系统的基本理论和一般方法，作为知识工程的核心内容。因此，知识工程是人工智能中以知识为研究对象的一门新兴学科，它是人工智能、数理逻辑、数据库技术和心理学等学科交叉发展的结果，它是一门主要研究如何利用知识以求解人类的智能才能解决的问题的工程学，其内容包括：

- (1)人工智能的应用研究；
- (2)知识获取、表示和利用的技术；
- (3)知识信息的处理技术；
- (4)利用符号操作完成表示知识、推论等的程序设计技术；
- (5)知识系统的开发工具；
- (6)根据领域专家的知识建造专家系统的工程学。

人工智能的符号程序设计、问题求解和搜索三个概念构成知识工程的技术基础。图1-3表示构造一个知识系统所使用的技术。基础层包括符号程序设计、命题演算、搜索、启发式方法。技术的第二层为常用的知识表示形式，包括约束、断言、规则和可信因子。知识系统包含约束是为了表示对允许状态、值或结论的限制。

断言对应真命题，即事实。规则表示特定形式的陈述性或命题性知识。一般地说，陈述性规则描述事物的工作方式，例如“如果病人患感冒，则可能会发烧或流鼻涕”，规则告诉系统相信什么。命令性规则指示了知识系统在运行过程中必须使用的启发式方法，例如，“如果病人发烧和流鼻涕，则应怀疑他可能感冒了”这个规则告诉系统要如何行动。

可信因子指定了知识系统对其数据、规则或结论应赋予怎样的置信度和有效程度。

第三层包括组织与控制、中间结果、解释与证明、优化。知识系统按照体现的结构设计原理来组织控制自己的活动。每一不同的控制模式有其相应的知识库组织方式及搜索该知识库和应用知识的推理机制，因此，控制和组织是密切联系的，一般的知识系统，都会出现中间结果，需要有效地使用中间结果，一般来说，要求知识系统对用户是

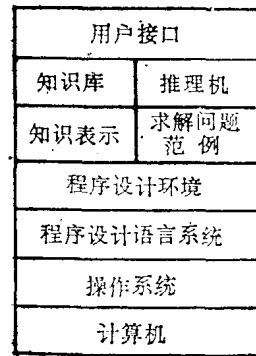


图1-2 知识系统

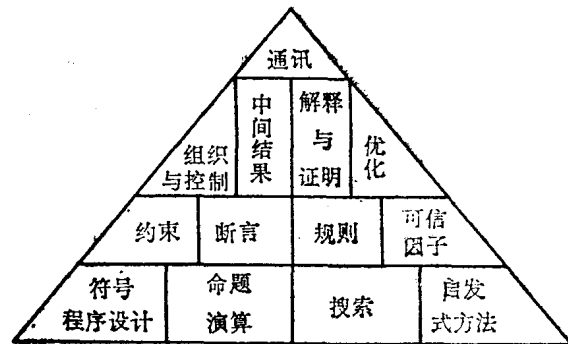


图1-3 知识系统使用的技术

· 透明的，通过解释和论证结果，给用户产生一个合理的印象。在知识系统中，优化技术起着重要的作用，它能有效地节省计算机资源，提高运行速度。

最顶层是通讯技术，知识系统通过通讯能力与知识工程师、专家、端点用户、数据库及其它计算机系统进行通讯。

知识工程为解放人类智力，为建立各种知识系统提供理论基础和技术。目前，知识系统有知识库系统、专家系统、智能机器人、智能辅助教学系统、自然语言理解和决策支持系统等，这些系统被用于医疗卫生、教育、地质勘探、军事等各个部门，已取得了巨大的经济效益和社会效益。随着各种技术的发展和社会的进步，知识系统的应用将会更加广泛，知识工程的理论和技术将会不断的完备。

第二章 PROLOG语言

PROLOG语言也是一种计算机程序设计语言。从名字上看, PROLOG是Programming in Logic的缩写, 这种语言是1972年前后法国Marseille大学的A.Colmerauer设计的, 并用Algol—w完成了它的解释程序。直至1981年10月, 日本公布的第五代计算机研制规划指定以PROLOG为它的核心语言以后, PROLOG语言便成为计算机界众所瞩目的计算机程序设计语言。

PROLOG语言具有递归、回溯、模式匹配、数据库等功能, 特别适用于非数值处理, 如人工智能、专家系统等领域。为了本书描述算法的需要, 以下介绍PROLOG语言的基本概念、语言规则、基本算法和基本谓词等内容。为便于初学的读者理解这些内容, 各部分相应地列举了一些例子。为了方便阅读, 假定我们使用一个中文PROLOG系统。

§ 2.1 PROLOG语言的基本概念

PROLOG语言由以下三部分组成:

1. 事实 即对客观事物的对象和关系的描述。
2. 规则 即定义有关对象和关系的规则。
3. 询问 即给对象和关系进行询问。

以下我们用一个例子来说明上述组成部分, 这个例子本身就是一个PROLOG语言程序, 为了后面说明方便, 在每个语句前面加上一个行号。

例2.1 PROLOG程序

1. *colour*(‘苹果’, ‘红’).
2. *colour*(‘桔子’, ‘黄’).
3. *colour*(‘玉米’, ‘黄’).
4. *colour*(‘草莓’, ‘红’).
5. *colour*(‘西红柿’, ‘红’).
6. *colour*(‘胡萝卜’, ‘红’).
7. *fruit* (‘桔子’).
8. *fruit* (‘苹果’).
9. *fruit* (‘草莓’).
10. *vegetable*(‘西红柿’).
11. *vegetable*(‘胡萝卜’).
12. *corn*(‘玉米’).
13. *like*(‘张三’, *X*), —

- ```

colour(X, '红'), vegetable(X).
14. like('李四', X); —
 colour(X, '黄'), fruit(X).
15. ? —like('张三', 'X').

```

这段程序共15行，每行都是一个PROLOG语句。前12行是事实，是对事物的对象和关系的描述。给出了一些水果、蔬菜和谷物的颜色，这些事实即构成关系数据库。第13和14行给出规则，规则指定张三和李四喜欢食物的颜色及类别，这些规则加到数据库中，即构成——知识库。第15行是询问，即提出要求解的问题，根据这个提问，PROLOG即搜索知识库得出第一个解是：

$X = \text{西红柿}$

第二个解是：

$X = \text{胡萝卜}$

下面进一步讨论上例中三种类型语句的格式。

## 一、事实

第1行至第6行包括了二个对象：食物(如草莓、苹果、桔子、西红柿、胡萝卜、玉米)和它们的颜色(红、黄)，且有一种颜色属性关系colour，在PROLOG的语句里，写成如下的标准形式：

colour('苹果', '红')或colour(apple, red),

colour('桔子', '黄')或colour(orange, yellow)等等。第7行至12行包括了一个对象：食物(如桔子、苹果等)和它的类型属性关系fruit、vegetable、corn，用PROLOG语言写成如下形式：

fruit('桔子')或fruit(orange)

vegetable('西红柿')或vegetable(tomato)

corn('玉米')或corn(maize)

等等。

从上述一些句子中，可以看到表达事实的PROLOG语句的规则是：

(1)对象和关系必须以小写字母开头，如果使用中文，必须包含在单引号内。例如，对象苹果、apple、红、red。

colour('苹果', '红')。

colour(apple, red)。

fruit('苹果')。

fruit(apple)。

(2)语句以关系开头，所有对象写在圆括号内，各对象之间用逗号“,”分开。例如关系colour：

colour('苹果', '红')。

colour(apple, red)。

(3)语句结束，必须写上实心句点“.”。

## 二、规 则

要表示张三和李四爱吃的东西的颜色和类别，可分别用13及14行语句来表示。在PROLOG中，规则分为头部和体部两部分，这两部分之间用冒号和连字符“:—”连接起来，“:—”可理解为如果的意思。例如，13行语句规则的头部是：*like*(‘张三’, *X*)，体部是：*colour*(*X*, ‘红’), *vegetable*(*X*)。可理解为：如果食物是红色的且是蔬菜，则它就是张三爱吃的食物。

一个规则的头部描述这个规则要定义的事实，规则的体部是一些必须满足的子目标(事实)。各子目标之间用逗号“,”分开，只有这些子目标全部满足时，头部才能为真。为进一步说明这个问题，下面再举一个例子。

### 例2.2 定义一个祖父的规则

```
father(a, b).
father(b, c).
grandfather(X, Z):—
father(X, Y), father(Y, Z).
```

程序第一句的意思是：*a*是*b*的父亲，第二句的意思是：*b*是*c*的父亲，都是表示某种事实。而第三句是表示一个规则，规则的头部是描述要定义*X*是*Z*的祖父这一事实，规则的体部给出两个子目标，它描述了*X*是*Y*的父亲及*Y*是*Z*的父亲。这个规则的含义是：如果*X*是*Y*的父亲且*Y*是*Z*的父亲，则*X*是*Z*的祖父。

## 三、询 问

上面读到过，PROLOG中所有事实的集合实际上构成一数据库，在数据库的基础上加上一些规则，则构成知识库。当PROLOG系统内存贮了知识库以后，程序员可以进行询问，要求系统回答。如果知识库中的知识能解答所提问题，则输出解的结果，否则，PROLOG回答“no”，表示系统不能求解所提问题。询问语句由一个问号“?”和一个连接字符“—”开头，跟着一个询问的事实。例如对例2.2中的知识库提出问题：

```
? —grandfather(Q, c).
```

即问*c*的祖父是谁？PROLOG将自动搜索知识库，寻求匹配而回答：

```
Q = a
```

即*a*是*c*的祖父。如果提问：

```
? —grandfather (Q, a).
```

即询问*a*的祖父是谁？因从知识库中的事实和规则不能推出*a*的祖父是谁，因而PROLOG回答“no”。PROLOG如何搜索知识库，回答询问呢？将在第三部分中加以说明。

## § 2.2 基本语法

上面的二个例子，大致描述了PROLOG程序一般的语句规则，以下我们用巴科斯

一诺尔范式来描述PROLOG语言的语法规则。巴科斯—诺尔范式是以巴科斯和诺尔的名字命名的一种形式化的语法表示法，这种表示法使用了一些符号，其中：

“ $::=$ ”表示“定义为”的意思。

“ $|$ ”表示“或者”的意思。

“ $\langle \rangle$ ”尖括号，用尖括号括起来的字符序列是非终结符。

“ $\{ \}$ ”花括号，用花括号括起来的部分为选或不选取项。

利用上述符号，构成PROLOG程序的语法可表示为：

#### 1. PROLOG 程序

$\langle \text{程序} \rangle ::= \langle \text{语句} \rangle . \{ \langle \text{语句} \rangle . \}$

即说明PROLOG程序由若干语句组成。

#### 2. 语句

$\langle \text{语句} \rangle ::= \langle \text{目标} \rangle$

$| \langle \text{目标} \rangle : - \langle \text{目标} \rangle \{ , \langle \text{目标} \rangle \}$

$| ? - \langle \text{目标} \rangle \{ , \langle \text{目标} \rangle \}$

即是PROLOG程序的语句定义为目标(如例2.1中的1至12行)或规则(如例2.1中的13、14行)或询问(如例2.1中的15行)。

#### 3. 目标

$\langle \text{目标} \rangle ::= \langle \text{谓词} \rangle ( \langle \text{参数} \rangle \{ , \langle \text{参数} \rangle \} )$

$| \langle \text{参数} \rangle \langle \text{谓词} \rangle \langle \text{参数} \rangle$

$| \langle \text{谓词} \rangle$

$| \langle \text{空} \rangle$

即目标可定义为四种形式，上述最后一种是空，除空外，有三种基本形式：

(1) 前缀形式 这种形式谓词放在参数之前，参数用圆括号括住，各参数之间用逗号分开，例如：

*colour*(*apple*, *red*).

其中，*colour*是谓词，*apple*和*red*是二个参数。

*father*(*x*, *y*).

*father*是谓词，*x*和*y*是两个参数。

(2) 中缀形式 这种形式是谓词放在参数之间，这种形式一般是二元谓词。如：

$x = y$ ;  $x + y$

等，其中*x*和*y*是参数，等号“=”和加号“+”是谓词。

(3) 独立形式 这种形式中，谓词不带参数。例如，内部谓词(将在§2.4中说明) *nl*。

#### 4. 谓词和原子

$\langle \text{谓词} \rangle ::= \langle \text{原子} \rangle$

$\langle \text{原子} \rangle ::= \langle \text{小写字母} \rangle \{ \langle \text{尾部} \rangle \}$

$| \langle \text{谓词符号} \rangle$

$| \text{'任何字符串'}$

$\langle \text{尾部} \rangle ::= \langle \text{字母} \rangle$

| <数字> | <汉字>

<字母> ::= <小写字母>

| <大写字母>

<大写字母> ::= A | ... | Z | \_

<谓词符号>是表示谓词的某些特殊符号, 如: ., !, =, +, - 等。

#### 5. 参数和项

<参数> ::= <项>

<项> ::= <常量> | <变量> | <复合项> | <表>

#### 6. 常量

<常量> ::= <原子> | <整数>

<整数> ::= <数字> { <数字> }

#### 7. 变量

<变量> ::= <大写字母> { <尾部> }

<大写字母>中包括下划线字符“\_”。在 PROLOG 中, 变量表示可以取不同的赋名或不需赋名的量, 变量与原子的区别是: 变量以大写字母或下划线符开头, 原子只能以小写字母开头。单独一个下划线符表示无名变量, 如果一个句子中出现二个或二个以上的无名变量, 它们可以取不同的赋名。例2.2中的大写字母X、Y、Z和Q都是变量, 而小写字母a、b和c都是常量。

#### 8. 复合项

<复合项> ::= <函数名> (<项> { , <项> } )

| <项> <算符> <项> { <算符> <项> }

<函数名> ::= <原子>

<算符> ::= <原子>

如例2.1中的colour(‘苹果’, ‘红’)可看作复合项, 其中 colour看作是函数名而‘苹果’和‘红’可作项。例2.2中的father(a, b), father可以看作是函数名, 而a, b可以看作是项。又如

2 \* a \* X

其中2和a是常量(是项), X是变量(是项), “\*”是算符, 故2 \* a \* X是复合项。

#### 9. 表

<表> ::= [ ]

| [ <项> { , <项> } ]

| [ <项> | <表> ]

| [ <复合项> ]

| [ ‘任何串’ ]

#### 例2 3

(1) [ ] 是一个空表;

(2) [a] 是一元素组成的表;

- (3)  $[a, b]$  是二元素组成的表  
 (4)  $[a, b, [C, D]]$  是三元素组成的表;  
 (5)  $[X+Y, a+b]$  是二元素(复合项)组成的表,  
 (6)  $[-, x, a, b]$  是四元素组成的表。

将表的第一个元素叫表头, 表中除第一个元素以外的元素的表叫表尾。

**例2.4** 下表中分别列出表的表头和表尾。

| 表                  | 表头         | 表尾              |
|--------------------|------------|-----------------|
| $[ ]$              | 无头         | 无尾              |
| $[ a ]$            | $a$        | $[ ]$           |
| $[ a, b ]$         | $a$        | $[ b ]$         |
| $[ [ a, b ], c ]$  | $[ a, b ]$ | $[ c ]$         |
| $[ a, b, [C, D] ]$ | $a$        | $[ b, [C, D] ]$ |
| $[ X+Y, a+b ]$     | $X+Y$      | $[ a+b ]$       |

在 PROLOG 中, 将一个表分割为表头和表尾是一种常用的操作, 这个操作可以使用算符|实现。其形式为:

$[X|Y]$

表示  $X$  是某表的表头,  $Y$  为某表的表尾。

**例2.5** 分割算符|的应用

假定有如下数据库:

$\rho([1, 2, 3])$

$\rho([a, b, c, [d, e, f]])$

询问

$? \text{---} \rho([X|Y])$

PROLOG 将回答:

$X = 1 \quad Y = [2, 3]$

再询问:

$? \text{---} \rho( \text{---}, \text{---}, \text{---}, [ \_ | x ] )$

PROLOG 将回答

$X = [e, f]$

## § 2.3 PROLOG的基本算法

PROLOG 系统要回答询问, 主要是对知识库使用模式匹配和回溯两种算法来完成。