知识工程

童频　沈一栋　编著

科学出版社

知 识 工 程

童 颋 沈一栋 编著

科学出版社
1992

（京）新登字 092 号

内 容 简 介

本书系统地阐述了知识工程这一领域内的基本概念、原理与方法。主要内容有：知识工程学科领域综述，人工智能问题求解模型，智能问题求解的搜索方法、逻辑推理、规划方法，知识表示，知识库系统组成原理，人工智能程序设计风格及语言，专家系统，知识获取与机器学习，面向知识处理的系统结构等。

本书可作为大专院校计算机专业研究生或高年级学生的教材或参考书，亦可供有关专业领域的科技人员自学参考。

知 识 工 程

童 频 沈一栋 编著

责任编辑 刘晓融 唐兆亮

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100707

中国科学院印刷厂印刷

新华书店北京发行所发行 各地新华书店经售

*

1992 年 7 月第 一 版 开本：787×1092 1/16

1992 年 7 月第一次印刷 印张：20 1/4

印数：1—2 800 字数：459 000

ISBN 7-03-001610-6/TP·115

定 价：17.70 元

前　言

计算机作为人类历史上出现的划时代的信息处理工具，正在改变着人类社会的面貌，并把人类从工业社会引向信息社会．计算机的应用已经深入到基于知识处理的人类智能活动领域．特别是从60年代中期到70年代，相继研究、开发成功了像 DENDRAL 和 MYCIN 这样一些有代表性的“专家系统”以来，计算机作为知识处理工具的作用日益突出．在这样的背景下，E. 费根鲍姆(E. Feigenbaum) 在1977年首先提出了“知识工程(Knowledge Engineering)”这一概念，从此逐渐形成了一个学科交叉的崭新研究领域，并在80年代获得了很大发展．目前，知识工程已成为国际计算机界的一个“热门”研究方向．

众所周知，人类智慧的泉源是知识．同样，一切人工智能信息处理系统也将建立在知识库和知识处理的基础之上．知识工程的研究目标就是探索关于知识的表示、获取(包括学习)、保存(记忆)、交换(变换)、运用(包括检索、推理及其它各种形式的加工)的理论、方法及实现技术．知识工程的研究将从心理学和思维(或认知)科学获得启发、借鉴，同时，又和数据库理论/技术、人工智能理论/技术、语言学、逻辑学等学科密切相关．智能信息处理系统的实现，又有赖于软件工程方法、工具、环境和面向智能信息处理的多处理机系统结构的支持．

本书旨在系统地阐明关于知识工程的基础理论、基本方法和实现技术．它是在作者为研究生和专业科技人员讲授知识工程课程而编写的讲义的基础上，经修订改写而成的．

本书可作为大专院校计算机专业高年级学生或研究生的教材或参考书．采用本书作为教材的参考教学时数为48至60学时．

顺便指出，作为两门独立课程，知识工程和人工智能的教学内容有部分交叉、重叠，但侧重点是不同的．然而，从应用的角度来看，这两门课程的教学目标基本上一致．在需要同时或先后开出这两门课程时，应互相协调教学内容，避免不必要的重复．

本书第一和六—十一章由童频执笔；第二—五章及第七章 7.7，7.8 节部分内容由沈一栋执笔．童频统编全稿．武汉大学陈莘萌教授对书稿进行了全面审阅，不少学员和同行专家也对本书稿提出了很好的建议，在此一并向他们表示诚挚的谢意．作者还特别感谢国家自然科学基金委员会（NSFC）对与本书内容有关的研究工作的资助，以及科学出版社对本书编辑出版的大力支持．

由于知识工程是一个新的学科领域，发展特别迅速，作为知识工程课程的教材，其内容的取舍，也还没有形成一个公认的、成熟的准则，因此，书中难免有遗漏，甚至错误．建议教师采用本书进行教学时，根据具体情况增删部分内容，也殷切希望广大读者批评指正，以便再版时改进．

目　　录

第一章 引 论

计算机的应用，早已突破了数值计算的范畴而进入更为广泛的非数值处理领域．在非数值处理的应用领域内，人们也早已不满足于仅仅把计算机用于数据处理和控制方面，而把计算机的应用进一步深入到知识处理领域乃至模拟人类的智能活动．

数据处理要涉及大量的数据，知识处理则不仅要涉及大量数据，而且还要涉及人类从自身的活动中积累起来的大量知识．如何运用计算机这个信息处理工具，对这样大量的知识进行有效地存储、管理和利用，便自然地成为社会，尤其是计算机界所关注的问题．知识工程这一信息处理学科领域正是在这样的背景下应运而生的，或者更确切地说，是逐步演化而形成的．

1.1 数据处理、知识处理、智能问题求解

计算机的应用，大体上可以区分为数值计算和非数值处理两大方面．广义地说，从一般数据处理到知识处理，乃至智能问题求解，都属于非数值处理范畴．在非数值处理的场合，计算机的主要处理对象是以字符或字符串表示的数据；计算机所进行的最基本操作是基于逻辑比较或模式匹配的搜索（search）．表面看来，非数值计算似乎比数值计算所要求的基本操作还简单，但实际上，当涉及对文件或数据库的操作时，它却需要完全不同于数值计算的一系列新的操作，如图 1.1 所示．可见其中除了对数据项的基本逻辑操作外，在一般计算机的指令集中，几乎都不包含这些高层次的操作，只有通过宏替换或子程序调用，甚至执行一整套程序来实现．其结果是，不可避免地要付出更多的软件开销代价，从而使系统效能下降．因此，如何尽可能提高计算机用于非数据处理时的效能，始终是数据/知识工程领域内的重要研究课题之一．

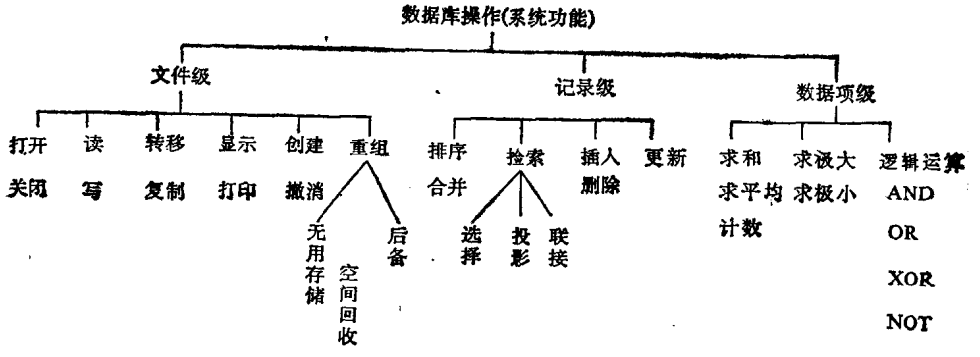

图 1.1 对数据库的基本操作

另一方面，人们早就不仅仅满足于把计算机推广应用于数据处理领域．早在计算机设计的萌芽时期，就有学者思考是否能用计算机来模拟人类的智能活动．人们从著名的

图灵试验(Turing test)得到启示．到 60 年代中期，就有人研究用计算机来下国际象棋并获得成功．1971 年出现了第一个实用的专家系统——DENDRAL，它可以根据质谱仪的实验数据，推导出化合物的分子结构．从此，像 MYCIN，PROSPECTOR 等不同领域的专家系统接踵而来．

人类智能的源泉是知识．因此，把计算机应用推广到更高一层的人类智能活动领域，首先要解决如何实现知识处理的问题．

知识处理有它自己的既不同于数值计算也不同于一般数据处理的特点．概括起来，有以下几个方面：

(1) 从给定的前件和约束条件(输入)出发，在问题空间中可能存在多条达到目标(输出)的求解路径，而且这些路径往往不能事先确定，即不存在确定的算法，往往需要依赖问题本身或求解过程中所提供的信息，也可以说是某种知识，采取启发式 (heuristic) 方法，找出最优或可接受的求解路径．

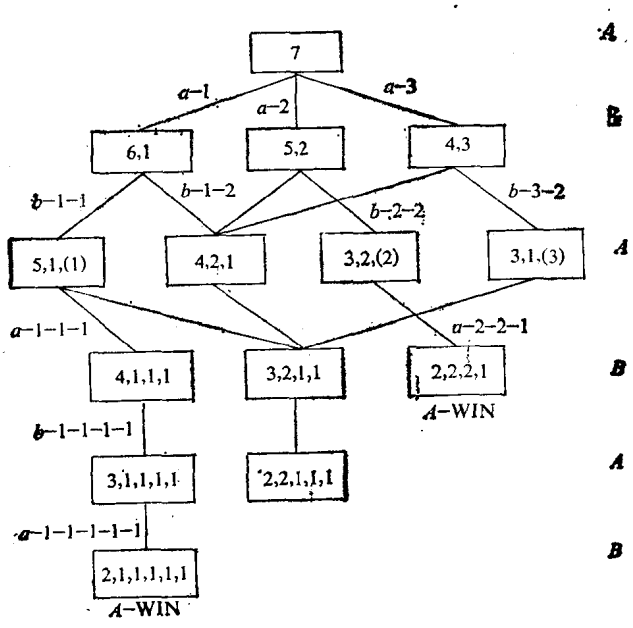

图 1.2 “分堆”游戏的问题空间及求解路径

在人工智能 (Artificial Intelligence, AI) 的研究中，常常把博弈作为智能问题求解的典型例子．例如，图 1.2 表示一种两人对弈的“分堆”游戏的问题空间．假设起始状态是给定 7 根火柴(输入)．约束条件是每轮到两个对弈者之一 *A* 或 *B* 动作时，他将把其中的一堆火柴分成两堆，但必须是火柴数目不相等的两堆．目标是，*A*，*B* 双方都力图使对方轮到动作时，不能再分而取胜．图 1.2 给出了这个问题的所有可能求解路径和合法状态的图 (graph)．可见，在这样一个简单的问题空间中，也明显地存在着达到目标的多条可能路径．例如，对于 *A* 来说，可能取胜的路径有：*a*—1，*b*—1—1，*a*—1—1—1，*b*—1—1—1—1，*a*—1—1—1—1—1 和 *a*—2，*b*—2—2，*a*—2—2—1 两条；对于 *B* 来说，可能取胜的路径则有 5 条之多．

于是，求解这类非确定型问题的基本模式可归结为：在问题空间中进行启发式搜索 (heuristic search)，力求找到引向目标的较好路径．可见这种模型本身就带有不确定性，

完全不同于确定的算法模型.

实际的知识处理所面临的问题空间往往十分巨大，如果不依靠基于知识的启发式方法缩小问题的搜索空间，几乎不可能求解. 例如，西洋棋的问题空间共包含10^{40}个结点，国际象棋达10^{120}，而围棋则高达10^{700}. 如果企图用遍历法搜索整个问题空间来解决问题，显然是不可思议的. 然而人的智慧却能够在有限时间内，通过朝前看若干步的思考，选择最有希望取胜的走法. 即使是当代性能最好的计算机上的象棋程序，也未见得能够胜过棋艺高超的大师(尽管也有计算机象棋程序战胜业余棋手而夺冠的记录).

(2) 人类知识中，存在着许多模糊概念，不能用二值逻辑来表达. 对于不精确的、不肯定的、不完全的知识，或者反映主观愿望、猜测、可能性、因时因地而变的情况等常识性的知识，都难以用经典逻辑来表达.

(3) 知识处理主要是通过推理 (inferencing or reasoning) 来求解问题. 推算或计算都是为推理服务的辅助手段.

(4) 对问题域中的知识处理，主要是在语义这一层上进行，知识的表达必然与语义紧密联系.

(5) 面向问题的非过程的语言，更适合于表达知识及对知识处理的要求. 运用这类语言的 AI 程序设计风格和面向过程的程序设计风格有很大的不同 (将在第七章中详述).

(6)一般数据处理，主要是对数据库的操作，知识处理则主要是对知识库的操作，或者说，是运用知识库所提供的知识进行推理.

由于知识处理具有以上这些主要特点，传统的面向数值计算的体系结构，已经难以适应.

所以，如何根据知识处理的特点和要求来设计新的计算机系统结构，已经成为举世瞩目的研究课题. 在日本、美国、欧洲诸发达国家，都有相当规模的关于"新一代计算机系统"的研究计划. 在我国，也已把研究开发所谓智能机列入国家高技术发展规划.

1.2 数据库、演绎数据库、知识库

如前所述，作为数据处理系统的核心是数据库 (data base)；作为知识处理系统的核心是知识库 (knowledge base). 关于数据库，大家都已比较熟悉，不必多说，需要略加说明的是：什么是知识库？知识库和数据库的根本区别何在？它们有什么联系？

1.2.1 知识库和数据库的区别和联系

简单说来，知识库和数据库的根本区别在于，知识库中不仅包含明显地表达的事实 (facts) 和关系 (relationships)，而且还包括明显地表达的包括常识、经验等在内的领域知识和推理规则等. 例如:

(1) "技术员曾国庆出生于 1965 年 10 月 1 日"——这是某工厂数据库中关于职工档案文件中的一条记录中的一部分，反映了职工曾国庆是一名技术员，出生于 1965 年 10 月 1 日这样一个事实. 其中"技术员"、"曾国庆"、"1965 年 10 月 1 日" 等分别是数据项，它们分别是属性 (attributes) "职务"、"姓名"、"出生日期"的值 (value). 把属于同一名职工的数据项组织在同一条记录内，体现了这些数据项之间的依赖关系 (dependencies).

(2) 年龄$=\begin{cases}\text{当前年}-\text{出生年}-1, & \text{若(当前月}-\text{出生月)}<0\\ \text{当前年}-\text{出生年}, & \text{若(当前月}-\text{出生月)}\geqslant 0\end{cases}$

则可以说就是关于如何计算年龄的知识，它可作为附加过程（attached procedure）纳入数据库系统中.

(3) 年龄在35岁以下18足岁以上者是青年. 这也是知识，它可以用一条规则表述如下：

若 $18\leqslant X$ 的年龄 <35，则 X 是青年.

或者，也可以用谓词逻辑的形式表述为：

青年 $(X)\leftarrow$ 年龄 $(X), \geqslant 18$，年龄 $(X), <35$

(4) 青年人富有进取心，但缺乏经验.这是常识，也是一种知识. 其中"青年人"、"富有"、"进取心"、"缺乏"、"经验"等概念全是模糊概念，可以用模糊集（fuzzy set）隶属函数来表示程度，也可以用所谓置信度（Certainty Factor, CF）来描述. 例如，某人为18岁，则他是青年的 CF = 1；若某人是35岁，则他是青年的 CF = 0.1 等等.

(5) 已知曾国庆的父亲是曾大江，曾国庆的女儿是曾雪梅，则可以推断曾雪梅的祖父是曾大江，或者曾大江的孙女儿是曾雪梅. 这也是一种知识，是一种隐含继承性（heritance）的知识. 在知识库中可以用"ISA"联系构成语义网络来表达. 当然，也可以用其它方式来表达. 例如，可在上述数据库中加入一条规则：

若 X 是 Z 的父亲与 Z 是 Y 的父亲，则 X 是 Y 的祖父.

或者用一阶谓词逻辑子句表述为：

祖父$(X, Y)\leftarrow$ 父亲 (X, Z)，父亲 (Z, Y).

综上所述，从知识库的角度来看，一个知识库的基本内容可概括为：

$$\mathbf{KB}\supset \boldsymbol{F}\cup\boldsymbol{R}\cup\boldsymbol{A}_P \tag{1.1}$$

其中 **KB** 代表知识库；$\boldsymbol{F}$ 代表事实集，用数据库的术语来说，相当于实体及其属性所对应的数据集合；$\boldsymbol{R}$ 代表规则集，其中所包含的与领域有关的语义知识，在数据库中，通常是隐含于数据结构中或者完整性约束中，而在知识库中，则主要是通过某种知识表达方式，与事实一样明显地表达出来；$\boldsymbol{A}_P$ 代表附加过程.

可见数据库与知识库既有区别又有联系，它们中间没有一条泾渭分明的界限. 数据库中所包含的明显表达的关系和约束条件，可以视为某种低层（low-level）知识；知识库中所包含的有明确定义的事实断言和通过规则表达的某些明显的关系，实际上也是数据库的内容. 在用逻辑程序设计语言，例如 PROLOG，来描述时，程序、数据库、知识库这三者在形式上也统一起来了.

关系型数据库的理论和数理逻辑有着密切的关系. 从逻辑的观点来看，只要把反映因果关系的演绎推理规则以逻辑蕴涵的方式加入到数据库中，就能运用这些逻辑蕴涵，从数据库中已有的事实推导出新的事实. 具有这种性质的数据库，叫做演绎数据库（deductive databases），可以看作是从数据库向知识库的自然延伸.

还有一种观点是把知识库看作是由两部分知识所组成. 一部分是所谓内涵知识（intensional knowledge），另一部分是所谓外延知识（extensional knowledge）. 式(1.1)中的 $\boldsymbol{R}$ 和 $\boldsymbol{A}_P$ 都属于内涵知识；组成数据库内容的全部数据，即式(1.1)中的 $\boldsymbol{F}$，则属

于外延知识.推理（inference）或推论（reasoning）的过程,就是运用内涵知识对外延知识操作的过程.

从上述不同角度的论点来看,可以认为数据库、演绎数据库和知识库之间，就其内容来说,存在以下包含关系:

$$DB \subset DDB \subset KB \tag{1.2}$$

其中 DB，DDB，KB 分别代表知识库、演绎数据库和知识库的内容.

1.2.2 数据模型与知识表达

数据模型抽象地反映了数据库的逻辑结构；知识表达则抽象地反映了知识库的逻辑结构.

在数据库技术领域内,众所周知的三大数据模式,即关系型、网络型和层次型,是在较低层次上关于数据库逻辑结构的抽象. 如果在更高层次上进行抽象，可以发现数据模型和知识表达之间存在着许多共同性.

从数据处理和知识处理的角度来看,反映客观世界的基本要素可以抽象地概括为:

(1) 客体型（object type)，即逻辑客体（logic object). 是具有相同或类似属性的实体集（entity）或事实集（facts set)，正是数据或知识处理的基本对象（object).

(2) 联系（association). 反映了客体之间、实体之间和属性之间以及它们自身之间的各种关系,

(3) 状态（state). 这是关于客体及其联系的形态和内容的抽象,

(4) 操作或称算子（operator). 施加于客体的结果将改变状态或标识一个客体-联系子集.

(5) 推理规则（inference rules). 又有演绎推理与归纳推理之分. 用来从已有事实及其联系,推导出新的事实及其联系.

(6) 公理集（set of axioms). 完整性约束（integrity constraint）可视为公理集的一个子集.

所谓数据模型或知识表达，都是研究如何把上述诸要素组合在某种一致的逻辑结构中.

较高层次上的抽象数据模型称概念数据模型（conceptual data model). 现在研究和应用得较多的有以下三种:

(1) 实体-关系模型（Entity-Relationship model, 简称 E-R model). 这种模型的基本思想是把客观世界看成是由多个实体及其语义关系所组成. 每个实体则是一个包含一个或多个函数依赖属性的集合. 例如,对一个运输部门来说,职工、车辆、进货、发货、仓库等等都是实体的例子;姓名、性别、工种、出生年月等等,则是"职工"的属性.

在这类模型中，语义既表达在实体与实体间的关系中，也通过函数依赖关系隐含于实体中,或者以完整性约束的方式来表达.

E-R 模型和一阶谓词逻辑知识表达方式之间,存在着某种逻辑上的对应联系，也容易转化为语义网络的知识表达方式.

(2) 数据抽象模型（data abstraction model). 这种模型的基本思想是找出客体或实体之间的性质继承关系（properties inheritance）构成一个 ISA 格（lattice)，也可把

它视为某种特定形式的语义网络，或者语义网络的组成部分.

(3) 客体或对象模型 (object model). 这种模型和面向客体或对象的程序设计 (object-oriented programming) 风格恰好协调一致. 客体模型的基本思想也和框架知识表达 (frame knowledge representation) 一致，它具有以下主要特点:

① 自含描述 (self-describing), 即关于客体的解释性信息和客体本身储存在一起;

② 强类型化 (strongly typed), 即只有定义于客体的操作能够用于本客体;

③ 信息隐蔽 (information hiding), 即程序员只需了解如何利用一个客体而不管该客体内部是怎样处理信息的.

关于知识表达问题，至今仍然是 AI 领域内的主要研究课题之一，目前已经进行了广泛研究的知识表达模式有以下几种:

(1) 产生式规则 (production rules). 这种表达方式的基本结构是:

IF 〈premise〉 THEN 〈co clusion〉

或

IF 〈condition〉 THEN 〈action〉

前者体现了产生式规则的说明性语义，即说明了前提和结论间的因果关系;后者则有助于从过程性语义的角度来理解，即把"条件"视为过程调用(按模式匹配)，"动作"部分视为过程体.

(2) 一阶谓词逻辑 (1st-order predicate logic). PROLOG 语言就是已被公认的实现这种表达方式的一种典型工具.

(3) 语义网络 (semantic net). 是一种基于分类学的推理模式，按照类似于动、植物分类归属的方法，建立 ISA, Part-of 之类语义联系.

(4) 框架 (frame). 这种表达方式类似于数据库的客体模式，比较适合于表达具有固定模式 (pattern) 或典型化的知识. 例如某类事物的性质及变化条件等等. 在一个框架中，可以随意加入附加过程.

(5) 模糊逻辑 (fuzzy logic). 这是正在日益受到重视的一种表达方式，但尚处于探索研究阶段. 由于作为人类知识的结晶——概念 (concepts), 往往具有模糊性. 因此，研究如何应用模糊逻辑或模糊集这一数学工具来恰如其分地表达知识，是十分有意义的.

综上所述，可以看出，现有高层数据模式与知识表达方式之间，存在着共性或相似性. 例如，前面介绍的客体模式类似于框架知识表达方式，而其它几种数据模式，可以说都在不同程度上类似于语义网络知识表达方式.

1.2.3 数据库/知识库管理系统

1. 数据处理系统和知识处理系统的基本结构

图 1.3 (a), (b) 分别表示一个数据处理系统和一个知识处理系统的基本结构. 二者在系统结构上虽有相似之处，但实质上是不同的.

对于数据处理系统，用户主要是通过数据操作语言 (Data Manipulation Language, DML) 或查询语言 (query language) 和数据库打交道. 查询语言通常有交互作用式和嵌入式两种运用方式. 当嵌入某种宿主语言而融合在应用程序中时，还需要有预处理接