

知识工程与机器发现

孙怀民 胡岗



北京航空航天大学出版社

《知识工程与机器发现》

孙怀民 胡 岗 编著

北京航空航天大学出版社

(京)新登字166号

内 容 简 介

本书是为计算机软件专业高年级学生和研究生讲授知识工程理论基础的教材。全书分成两篇。第一篇讨论形式系统理论。介绍了一阶形式系统理论基础、哥德尔不完全性定理以及形式系统的判定问题。第二篇探讨了知识工程中的两个核心问题：机器推理与机器发现（机器学习）如何形式化。

知识工程与机器发现

ZHISHI GONGCHENG YU JIQI FAXIAN

编 著 孙怀民 胡岗

责任编辑 肖之中

北京航空航天大学出版社出版

新华书店总店科技发行所发行 各地新华书店经销

通县觅子店印刷厂印装

787×1092 1/32 印张：6.75 字数：157千字

1992年4月第一版 1992年4月第一次印刷 印数：2000册

ISBN 7-81012-285-1/TP·053 定价：1.80元

序 言

本书是针对计算机软件专业的学生而编写的一本教材。

对于知识工程导论的教学，传统的习惯是侧重介绍知识表示、知识库和推理机的建造等方面的方法和技术。我们感到，这样做有三个问题。

第一个问题是：作为导论性的教材，很难对上述内容作深入具体的介绍，因此，对于那些准备以知识工程作为其主攻方向的学生来说，这门课失之浮浅。对于其它软件专业的学生，他们会感到这门课对自己没有什么用处，因而兴趣不大。

第二个问题是：知识工程系统的设计技术，本身还处于不成熟的阶段，它还在不断地发展和更新。因此，过多地陷入技术性细节以及对已有知识工程系统的介绍，往往会使学生感到意义不大。

第三个问题是：不易从方法论的角度对知识工程的理论基础进行探讨。

因此，在本书中我们作了一个新的尝试。我们把注意力的焦点集中于知识工程的理论基础——形式系统理论。对一阶语言的形式系统作了比较详尽的介绍，并在此基础上介绍了机器定理证明的理论和技巧，作为形式系统与知识工程关系的一个实例，我们还介绍了刻划机器发现(机器学习)的形式系统，后者是本书的作者们所承担的国家863高科技计划项目及国家自然科学基金项目的一些初步研究结果，在此我们

对资助者表示谢意。

我们希望，这样做能为那些立志致力于钻研知识工程的软件专业学生提供一个比较扎实的理论基础。由于形式系统理论对于计算机科学的各个领域例如软件工程、数据库技术等都是非常重要的，所以我们希望这样做对其他软件专业的学生也能有所裨益。

孙怀民同志编写了本书的第一章和第六章，胡岗同志编写的本书的第五章，胡岗，苏梅兰，江帆共同编写了本书的第二、三、四章，全书由孙怀民同志定稿。

由于编著者的水平所限，本书谬误之处在所难免，同时，由于我们在本书中企图另辟新径，也必然会产生许多问题，我们诚恳地盼望同行们予以批评、指正。

孙怀民 1990.12.24

目 录

第一章 绪论——知识工程、形式系统理论与机器发现

- § 1.1 知识工程对传统软件工程的挑战 (1)
- § 1.2 形式系统理论——知识工程的理论基
础 (3)
- § 1.3 知识工程能从科学哲学中学习什么 ... (5)
- § 1.4 Lakatos的证伪方法论 (8)

第一篇

一阶形式系统理论

第二章 形式系统理论基础

- § 2.1 谓词演算 (12)
- § 2.2 一阶理论 (41)

第三章 哥德尔不完全性定理

- § 3.1 引论 (68)
- § 3.2 可表达性 (70)
- § 3.3 递归函数和递归关系 (79)
- § 3.4 哥德尔数 (88)
- § 3.5 不完全性的证明 (93)

第四章 可计算性和判定问题

- § 4.1 图灵机 (100)
- § 4.2 判定 (108)

第 二 篇

机器推理与机器发现

第五章 机器定理证明

§ 5.1 Herbrand定理	(122)
§ 5.2 归结原理	(136)
§ 5.3 一阶树推理方法	(146)
§ 5.4 一阶线性归结、Horn归结及PROLOG 系统	(158)

第六章 机器定理发现

§ 6.1 机器定理发现与机器定理证明	(172)
§ 6.2 二阶树推理系统	(174)
§ 6.3 二阶树推理与数学发现	(179)
§ 6.4 二阶线性归结	(189)
§ 6.5 二阶线性归结与逻辑程序自动设计	(195)
参考文献	(209)

第一章 绪 论

——知识工程、形式系统理论与机器发现

§ 1.1 知识工程对传统软件工程的挑战

知识工程追求的目标是使计算机去解决那些原来需要由人类专家凭藉其专门知识去解决的问题，或者简称之为知识处理问题。这意味着计算机的应用有可能获得进一步的扩大，从处理日常生活中事务性数据信息发展到知识性信息处理。这不仅需要专门领域的理论和经验知识，而且还需要高度的智能性的技巧。因此，知识工程的核心问题和人工智能(AI)其它领域是一样的，即如何使人类的智能行为在计算机上再现的问题。

但是知识工程对待这个问题的观点却不同于所谓AI认知学派，后者认为解决问题的方法是使计算机能模拟人的智能行为。基于这种观点，他们往往认为他们所发展的技术——例如知识表示、推理机制等方面的程序是对人类智能行为的刻划。有些人甚至宣称他们的程序揭示了一些重要的人类思维规律。从人工智能卅多年的发展情况来看，这条道路目前是困难重重的。当然，我们这样说并不排斥，通过对认知学、心理学、思维科学各方面深入的研究与了解后，人们最终

能制造出能模拟人类智能行为的机器，但是这恐怕将是比较遥远的事。在目前计算机软硬件技术基础上，要想真正模拟人类智能行为是十分困难的。关于这方面，美国哲学家H·Dreyfus在他的“计算机不能做什么”一书中有深刻而精辟的（虽然有时是失之偏颇的）论述，我们在这里不准备详细讨论这个问题。

知识工程的研究者们不同于此。他们主张立足于今天的软硬件技术基础，研究如何能使计算机在功能上表现出某些人类智能行为，通俗地说，就是不管怎么干，只要能解决问题就行。因此，也有人把知识工程的研究者称为AI的工程学派。

但是，不管计算机用什么方法去解题，这些方法归根结底得靠程序来实现。从这个意义上说，今天所谓的知识工程，实质上只是传统软件工程的延伸。只不过，由于知识处理本身的复杂性和特殊性，所以它对传统的软件工程技术提出了更高的要求。下面我们将较为详尽地探讨一下这个问题。

近十多年来，知识工程的研究取得了显著的进展，但是同时也碰到了巨大的困难。这主要是在由实验室阶段向实用阶段过渡时所碰到的困难。例如，实验研究阶段的一些专家系统运行效果十分令人满意，但要进一步发展成实用系统，情况就不同了。因为实验系统的开发往往是针对问题的简化模型进行的。它的知识库很小，知识工程师和领域专家可以通力合作对解题方法及程序实现进行精雕细琢。而一旦放到实用环境中，知识库的规模将成百倍地增加，对系统性能的每一次改进都需要对庞大的知识库进行修正和重组。特别是，在知识工程师的知识和领域专家的知识之间有着很难逾

越的鸿沟，而知识工程系统实质上是前者的软件知识和后者的专门领域知识的有机的结合体，这大大增加了上述问题的难度。往往系统开发到后来，无论知识工程师还是领域专家都无法将它把握。近年来国际上有人提出所谓“专家系统危机”，就是针对这种情况而言的。

这种困难使人们逐渐认识到，知识工程，作为软件工程的延伸，有其本身的特殊性，因而提出了一系列新的要求。大致来说，一个能满足实用要求的知识工程系统，必须具备下述两个特点。

它的程序结构应该是开放式的，它的开发过程应该是探索式的。这是因为，解决专门领域问题的知识很难一次性地整理完善。领域专家会忽略某些环节，他可能认为这些环节是不言而喻的，也可能由于他自己凭直觉甚至灵感处理这类环节，无法把它们表示成有规律的形式。知识工程师也可能会曲解领域专家的意思，因而未能在程序中反映专家知识的实质。因此，专家系统的开发过程是一种边试用边修改的过程。开发周期和使用周期在很大的程度上是交叠在一起的。

其知识库和程序必须是多层次的，不但需要有解决专门领域具体问题的知识，还需要有如何运用这些知识的知识，即所谓元知识或方法论知识。不但需要有解决上述具体问题的程序，还需要有修改、更新甚至设计这些程序的程序，即所谓元程序或高阶程序。从这个意义上说，知识工程既是传统软件工程的延续，同时也对传统软件工程提出了新的挑战。

§ 1.2 形式系统理论——知识 工程的理论基础

上面的讨论告诉我们，必须使知识工程的方法与现代形

式系统理论紧密结合起来。这样才能为知识工程打下一个坚实的理论基础。事实上,我们无论要用计算机去解决什么问题,首先都需要把该问题形式化。知识工程当然也不例外。我们已经有了一个完善的和有力的形式系统理论,即数理逻辑中的一阶理论。它原是为研究数学的逻辑基础或所谓元数学而发展起来的。由于一阶理论具有很强的推理能力,而且提供了使推理过程机械化的手段,所以在人工智能领域中一直起着非常重要的作用。知识工程当然也不例外。以专家系统为例,专家系统中的知识库,实质上就是某个专门领域问题的一阶理论。专家系统中的推理机,实质上就是一阶形式系统推理规则的程序实现,过去做得不够的是,仅仅把形式理论中的局部成果——例如机器定理证明(自动推理)用来作为解决某些问题的技术,未能自觉地以形式系统理论为指导去建立知识工程的方法论。为了建立知识工程的方法论,我们必须对现代一阶理论的成果有深入的了解。只有这样做,我们才能对知识工程中的根本性的问题建立深刻认识。例如,某个任务能否形式化?所构成的形式系统能具备怎样的功能?能否满足实用的要求?有着怎样的局限性等等。只有这样,我们才不致对知识工程抱盲目的乐观态度或盲目的悲观态度。

知识工程的形式系统理论恐怕还不能只限于一阶。前面曾经谈到,知识工程系统的开发是开放式和探索式的。知识工程系统必须在运行中通过人机合作的方式不断地更新它的知识,这种任务的形式化就进入了二阶的范畴,它是如何将问题形式化的方法的形式化。例如,知识工程系统在使用过程中不断地修正和完善,如果每一次修正都需要由知识工程师去修改程序,那事实上是很难做到的。理想的方式是用户或领域专家能直接通过人机界面干预并改善程序的性能。一

般地说，我们不应希望用户和领域专家能用改写程序的方法去改进系统。他们只能用接近自然的方式，向系统输入一些信息，系统应该能根据这些信息，去改善它的知识库和推理方式。譬如说，当系统对一个问题给出错误的解答时，领域专家可能不知道造成错误的具体程序环节，他仅能给出正确的解答。我们希望系统能自动地去找造成错误的原因，根据这些原因和专家给出的正确实例去改善它的知识库或推理性能。这些功能的形式化相对于原有知识的形式化说来，就属于更高的层次了。这就表明，知识工程的理论不应仅包括一阶形式系统理论，还需要研究能否把人类的一些高级智能行为，如学习和发现等的某些方面，加以形式化，正是由于这个原因，当前知识工程的研究，才对机器学习和机器发现，付于极大的注意。

二阶形式系统的理论，是一个很困难的问题。我们只能希望把它们局限于某些局部的问题上。这些问题应该比较容易形式化的，但同时又是对知识工程的发展有实际意义的。一般地说，我们不应希望建造一个完全自足的机器发现或机器学习系统，而只能寄希望于经过仔细选择的、针对某个专门领域的人机合作系统，这样的人机合作系统应能对领域专家或知识工程师起到有效的助手作用。

机器发现或机器学习的研究，不仅需要现代形式系统理论的指导，还需要现代科学方法论的指导，我们将在下一节讨论这个问题。

§ 1.3 知识工程能从科学 哲学中学习什么

回顾过去人工智能界对机器发现和学习问题所做的工

作，我们就会发现，以前我们在这方面的研究基本上是依靠一种经验内省式的程序方法。也就是说，从一些特例出发，根据对人们解决这些问题的方法的朴素体会去建立经验模型，然后用精巧的程序技术加以实现。从方法论上看，这种作法仍然停留在培根的朴素归纳法阶段。现代科学哲学的成果在我们过去所作的工作中很少得到反映。为了建立机器发现和机器学习的方法论，我们有必要了解现代科学哲学在这方面所作的探索。

人类的知识如何增长，人类如何探索未知，这是科学哲学永恒的主题。远在希腊时代，柏拉图就提出了这样的诘难：

“你如何去探求未知的东西呢？如果你不知道它，你怎么知道你需要寻求什么呢？你又如何知道你找到的正是你所寻求的呢？”

当我们试图赋予机器以自动获取知识的能力时，也会碰到同样的疑难：机器怎样知道它该学什么呢？Burks在其“机率、因果和推理”一书中指出，对于科学发现来说，回答这一诘难，只需把知道问题和知道其解区别开来。一次探索，始于了解一个问题，终于找到它的解法。对于机器发现来说，这一回答同样适用。为此，需要有一种形式语言，使机器能够理解提出的问题，并且能够寻求和判别解答，也就是说需要建立机器发现逻辑。

早期科学哲学中的发现逻辑是企图给人们提供一种机械的规则簿。希望这样的规则簿能指导人们，从经验事实的特称陈述，正确地归纳出全称陈述。培根的归纳法就是这种思想的代表。尽管培根的方法是原始的。但是，他着眼于使逻辑成为科学发现的手段，这一思想还是十分可贵的。

18世纪，休谟对归纳法发难，提出了著名的休谟诘难。他指出，在先验方面，归纳的有效性也不能从经验证明。任何做这种证明的企图都将陷入循环论证，因而不可能给归纳法提供理性依据。休谟的诘难，终于使科学哲学放弃了培根的归纳法（可惜的是，同时也放弃了培根的重要思想）。后来的逻辑实证主义，不再把归纳逻辑看成发现的手段，而是把它看成是为已有的发现提供证实的手段。Carnap的概率逻辑是这种思想的代表。他的工作的目的是为科学假说提供或然性的证据。这样，现代科学哲学就把注意力转向如何为科学发现作辩护，或者说，转向了科学和伪科学的分界标准问题。至于说到发现本身，那就只好推给三个“I”，即直觉（Intuition）、冥思（Incubation）和顿悟（Illumination）。

这一转变是很令计算机科学家失望的。我们无法在机器上实现直觉、冥想和顿悟。为了使计算机具有学习和发现的能力，我们仍然期待着某种机械的规则簿。

在实证批判主义之后发展起来的理性批判主义，使计算机科学家看到了希望的曙光。理性批判主义的奠基人Popper指出，一切无限全称陈述形式的命题，不但逻辑上是不可证的，也是不可见的，因而概率归纳逻辑同样不能避免循环论证。

Popper和Lakatos在详细分析科学史的基础上，提出和发展了一种基于证伪的科学发现方法论。他们的依据是，无限全称陈述虽然不能被证实，却是可以被证伪的。他们把科学发现看成一个动态过程，认为科学的猜想（假说）是在证明、反驳、证明分析的交替过程中逐步形成和改进的。虽然理性批判主义仍然把他们的主要注意力放在分界问题上，

但由于他们强调了逻辑的作用，所以就有可能派生出一种有助于机器发现的逻辑模型。

当然，为此我们首先需要有一个形式演算系统，以便把他们的方法论形式化。如果说，科学史分析和玄思是科学哲学家手中的魔杖的话，那么，符号演算就是计算机科学家手中的魔杖。

§ 1.4 Lakatos的证伪方法论

在他的名著“证明与反驳——数学发现的逻辑”一书中，Lakatos分析了数学史上的两个案例——欧拉猜想和哥西猜想。从而揭示了反驳、证明、证明分析在发现过程中的作用。证明诱发反驳，反驳诱发证明分析，证明分析改进猜想及其证明，这种辩证统一的过程构成了发现的过程。这里我们以哥西猜想为例，对Lakatos的方法论作一简单说明。

原始猜想是：如果连续的级数函数是收敛的，则其极限函数也是收敛的，也就是说

如果

(1) 对任意 n ，函数 $f_n(X)$ 是连续的，且

(2) $\sum f_n(X)$ 是收敛的

则 $\sum f_n(X)$ 是连续的。

1821年，Cauchy给出了这个假说的一个证明。1822年，Fourier提出了他有名的级数

$$\sin(X) - \sin(2X)/2 + \sin(3X)/3 \dots \quad (3)$$

1826年，Albel注意到Fourier级数可以当作Cauchy定理的一个反例。

由于(3)肯定是一个收敛函数，根据Cauchy的 $\varepsilon \cdot \delta$ 的连续性定义，(3)的每个项都是连续的，但是它的极限函数

$$\delta(X) = \begin{cases} \pi/4 & \text{if } |X| < \pi/2 \\ 0 & \text{if } X = \pm \pi/2 \\ \pi/4 & \text{if } \pi/2 < |X| < \pi \end{cases}$$

是不连续的。

所以它是个Cauchy定理的全局反例（在他的书中，Lakatos称反驳一个逻辑推理序列的反例为全局反例，反驳前提的反例为部分反例）。Albel建议在Cauchy定理中用“幂级数”来代替“级数”的概念，这是一个典型的“怪物排除法”。直到1847年Seidel才发现了Cauchy的证明的错误引理，他使用的方法类似于Lakatos的反驳与证明分析思想，所以他抽象出绝对收敛的更深刻的概念。

这个故事是Lakatos方法学的典型示例。Lakatos把它的思想归纳为下面五条启发式则规：

1. 如果你有一个猜想，则试着证明和反驳它，找出全局反例和部分反例。

2. 如果你有一个全局反例，则放弃该猜想，通过证明分析找出反驳的引理，把它作为前提加到原始猜想中。

3. 如果你有一个部分反例，则检查它是否也是一个全局反例。如果它是，你就应用启发式规则2。

4. 如果你有一个部分反例且不是全局反例，则用没有反驳的引理代替反驳的引理。

5. 如果你有任何种类的反例，则使用演绎推理得出一个更深的定理，对该定理来说，这些例子不再是反驳。

我们看到，Lakatos方法的核心是证明分析，所以为了把Lakatos方法形式化，就需要建立比一阶证明理论高一个层次的形式系统。关于这个问题，我们将在第六章作一个初步的探讨。

