

知识工程与

知识发现

杨炳儒 主编



冶金工业出版社

知识工程与知识发现

杨炳儒 主编

北 京
冶 金 工 业 出 版 社
2000

内 容 提 要

本书是国内涉及知识发现领域的第一本专著。全书分为两大部分:第一部分是知识工程,分4篇(共26章),主要内容包括知识工程的逻辑基础,知识表示与推理机制,机器学习与知识获取,实用智能系统等基本内容与新进展;第二部分是知识发现(共8章),在综述与基本内容介绍的基础上,着重讲述了我们围绕国家自然科学基金重点项目所做的扩展性研究成果。

本书可供计算机、自动化、电子信息工程、测控技术与仪器、管理科学与工程、系统科学与工程、认知科学、科学哲学与方法论等专业的研究生、部分高年级学生、教学与科研人员、相关的工程技术人员阅读或参考。

图书在版编目(CIP)数据

知识工程与知识发现/杨炳儒主编. —北京:冶金工业出版社, 2000. 12

ISBN 7-5024-2681-7

I . 知… II . 杨… III . 知识工程-概论 IV . TP182

中国版本图书馆 CIP 数据核字(2000)第 49500 号

出版人 卿启云(北京沙滩嵩祝院北巷 39 号, 邮编 100009)

责任编辑 纵晓阳 谭学余 美术编辑 李 心 责任校对 刘 倩 责任印制 牛晓波

北京源海印刷厂印刷;冶金工业出版社发行;各地新华书店经销

2000 年 12 月第 1 版, 2000 年 12 月第 1 次印刷

787mm×1092mm 1/16; 44.5 印张; 1077 千字; 700 页; 1—2800 册

85.00 元

冶金工业出版社发行部 电话:(010)64044283 传真:(010)64027893

冶金书店 地址:北京东四西大街 46 号(100711) 电话:(010)65289081

(本社图书如有印装质量问题,本社发行部负责退换)

虽然我同意,科学知识只是日常知识或常识知识的发展,但是我坚决认为那些把自己限制在分析日常或常识知识或者这种知识在日常语言中的表述的人们,必定完全看不见认识论的最重要、最令人激动的问题。

——Karl R. Popper

前 言

一份最近的 Gartner 报告列举了 5 项将在今后 3~5 年内对工业产生重要影响的关键技术,其中 KDD 和人工智能排名第一,同时这份报告将并行计算机体系结构和 KDD 的研究列入今后 5 年内应该投资的 10 个新技术领域。这些领域与技术是当代计算机科学家与其他科学技术人员所面临的极富吸引力的当代科学发展的前沿,是最富挑战性的布满生长点的“多学科交叉的无人区”。

Acton 曾说过:“科学家最必需的东西莫过于科学的历史和发现的逻辑……”。当我们演绎地铺展与构建人工智能的主体与核心内容——知识工程和新崛起的知识发现的体系结构的时候,我们力求沿着其历史发展的轨迹较为完备与系统地去概括与综合其基本内容;利用结构主义的合理内核去剖析每一问题的基本结构;注重现实发展中最新颖的素材;合乎逻辑地将创新性研究成果集成;在理论与应用的结合点上阐明规律与技法。这些就是我们撰写此书所遵循的基本原则。

就知识发现系统的研究而言,由于我们正在承担着国家自然科学基金重点项目(No. 69835001)等缘故,我们汇集与学习了 300 多篇相关文献;并在充分占有资料的基础上开展了较长时间的研究与开发工作(包括与国外的合作),积累了点滴成果,我们将一些公开发表的内容纳入本书中。由于国内有关知识发现方面的专著不多,故希望本书的出版能起到抛砖引玉之作用。

本书是集体劳动的产物,我指导的博士生与部分硕士生参加了编写工作。其中,黄绍君与王许书参加了第一篇的编写工作;綦艳霞与张德政参加了第二篇的编写工作;谢永红与杨天梁等参加了第三篇的编写工作;王枏与唐菁参加了第四篇的编写工作;孙海洪与申江涛参加了第五篇的编写工作。

本书考虑到知识的继承性与积累性,取材众多,涉及到国内外一些专家学者,此处一并声明并致谢意。

由于本书论题具有科学前沿的性质且在不断发展中,加之作者水平的限制,故决定了本书所具有的探索性与尝试性,也不可避免地会出现一些缺点错误,诚请读者指正,以期改进与完善。

杨炳儒

2000 年 9 月

目 录

第一篇 知识工程的逻辑基础	(1)
第 1 章 形式系统概论	(3)
1.1 形式系统的产生与发展	(3)
1.2 形式系统的定义	(4)
1.3 形式系统的元语言	(6)
1.4 形式系统的元理论	(6)
第 2 章 命题演算	(10)
2.1 命题演算形式系统(FSPC)	(10)
2.2 FSPC 的讨论	(13)
第 3 章 一阶谓词演算	(16)
3.1 一阶谓词演算形式系统(FSFC)	(16)
3.2 FSFC 语构的讨论	(18)
3.3 FSFC 语义的讨论	(22)
3.4 语构与语义关系的讨论	(24)
3.5 FSFC 的形式表达能力	(28)
3.6 多型变元一阶谓词演算系统	(30)
第 4 章 二阶谓词逻辑	(33)
4.1 二阶谓词演算形式系统	(33)
4.2 二阶语义及其与二阶谓词演算系统的关系	(36)
4.3 知识表示的格林方法和科瓦尔斯基方法	(38)
第 5 章 模态逻辑	(41)
5.1 模态逻辑的正规系统及其语义	(41)
5.2 模态逻辑系统 NSKD、NSKT、NSKB、NSK4、NSK5 及其它	(46)
5.3 模态谓词演算	(49)
5.4 模态逻辑的几种解释(知道逻辑与信念逻辑)	(50)
第 6 章 时序逻辑	(56)
6.1 MPTL 的语言	(56)
6.2 MPTL 的语义	(57)
6.3 时序逻辑系统 MPTL	(59)
第 7 章 动态逻辑	(68)
7.1 命题动态逻辑	(68)
7.2 一阶动态逻辑	(70)
7.3 一阶动态逻辑的描述能力	(73)

第 8 章 多值逻辑	(76)
8.1 三值逻辑	(77)
8.2 无穷值逻辑	(80)
8.3 完备性定理	(83)
8.4 谱理论简介	(86)
8.5 多值逻辑的应用	(89)
第 9 章 模糊逻辑	(94)
9.1 “点值”模糊逻辑	(94)
9.2 语言真值模糊逻辑	(96)
9.3 区间值模糊逻辑	(98)
9.4 算子模糊逻辑	(99)
9.5 模糊逻辑的应用	(102)
第 10 章 非单调逻辑	(106)
10.1 非单调逻辑的产生	(106)
10.2 缺席推理逻辑	(109)
10.3 非单调逻辑	(111)
10.4 限定理论	(113)
第 11 章 归纳逻辑	(116)
11.1 经典归纳逻辑简介	(116)
11.2 现代归纳逻辑简介	(121)
11.3 归纳逻辑与人工智能	(132)
第二篇 知识表示与推理机制	(135)
第 12 章 知识表示方法	(137)
12.1 产生式表示法	(138)
12.2 语义网络表示法	(141)
12.3 框架表示法	(146)
12.4 谓词表示法	(149)
12.5 面向对象表示法	(153)
12.6 基于范例表示法	(157)
12.7 基于 Rough Set 表示法	(159)
12.8 基于语言场表示法	(161)
12.9 基于知识体表示法	(164)
12.10 其它表示方法	(170)
第 13 章 推理机制	(173)
13.1 不确定性推理	(173)
13.2 非单调推理	(210)
13.3 定性推理	(223)
13.4 其它推理	(239)

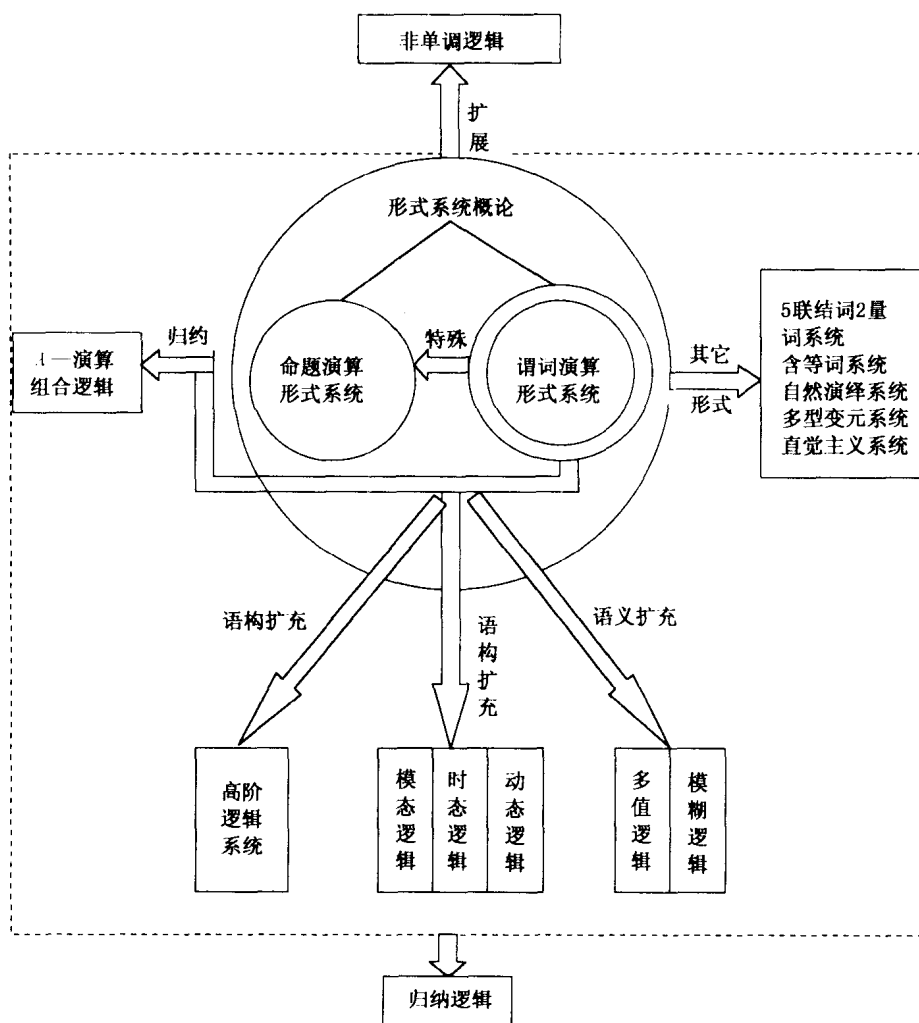
第 14 章 搜索策略	(248)
14.1 盲目搜索.....	(248)
14.2 启发式搜索.....	(253)
第三篇 机器学习与知识获取.....	(259)
第 15 章 概述	(261)
15.1 知识来源和关于学习的理论.....	(261)
15.2 机器学习.....	(262)
15.3 知识获取的主要手段.....	(264)
15.4 知识获取工具.....	(266)
第 16 章 通过例子学习	(277)
16.1 学习单个概念.....	(277)
16.2 学习多个概念.....	(286)
16.3 学习执行多步任务.....	(290)
第 17 章 通过类比学习	(299)
17.1 类比学习与推理系统.....	(299)
17.2 转换类比.....	(305)
17.3 派生类比系统.....	(309)
17.4 设计和再设计问题.....	(311)
17.5 重演中的问题.....	(312)
17.6 POPART 等简介	(317)
第 18 章 基于解释的学习	(338)
18.1 基于解释的抽象.....	(338)
18.2 基于解释的学习.....	(341)
第 19 章 通过观察学习	(350)
19.1 引言.....	(350)
19.2 概念内聚.....	(351)
19.3 所涉及到的术语和基本操作.....	(352)
19.4 聚类质量的标准.....	(356)
19.5 总结.....	(357)
第 20 章 问题求解中的机器学习方法	(358)
20.1 一个符号积分问题求解系统(FISLS)	(358)
20.2 一个类比求解方法的理论模型	(365)
20.3 基于案例的不定积分求解系统.....	(370)
20.4 反应块及其在类比推理中的应用.....	(375)
20.5 通过联想的学习方法及其应用.....	(380)
20.6 LEAP:一个数字电路设计的学徒学习系统	(384)
20.7 布尔电路设计操作的学习.....	(389)
第 21 章 知识获取	(394)

21.1	基于人工神经网络的知识获取方法·····	(394)
21.2	专家知识的归纳获取·····	(399)
第四篇	实用智能系统 ·····	(403)
第22章	专家系统 ·····	(405)
22.1	专家系统的概念·····	(405)
22.2	专家系统的基本结构·····	(411)
22.3	知识的获取·····	(412)
22.4	知识的管理·····	(416)
22.5	专家系统的设计原则·····	(421)
第23章	决策支持系统 ·····	(423)
23.1	基本概念·····	(423)
23.2	决策支持系统的基本结构·····	(431)
23.3	决策支持系统与管理信息系统的关系·····	(439)
23.4	决策支持系统的分类·····	(440)
23.5	决策支持应用实例·····	(447)
第24章	机器人规划 ·····	(451)
24.1	机器人规划的作用和任务·····	(452)
24.2	积木世界的机器人规划·····	(453)
24.3	使用目标栈的规划系统·····	(456)
24.4	STRIPS 规划系统 ·····	(461)
24.5	一个通用规划求解系统的简单介绍·····	(463)
24.6	具有学习能力的 PULP- I 规划系统 ·····	(467)
24.7	NOAH 规划系统 ·····	(468)
24.8	基于专家系统的机器人规划系统·····	(472)
24.9	基于知识的机器人规划系统·····	(475)
24.10	总结 ·····	(478)
第25章	新型计算机系统——神经计算机系统 ·····	(480)
25.1	神经计算机系统的背景及研究现状·····	(481)
25.2	神经网络的拓扑结构分析·····	(483)
25.3	神经网络模型·····	(486)
25.4	神经计算机系统的体系结构的考虑·····	(490)
25.5	Transputer 与多处理器系统 ·····	(493)
25.6	神经计算机系统的 3 种实现方式·····	(496)
25.7	神经网络计算机的一种分形实现·····	(502)
25.8	总结·····	(506)
第26章	其它新型计算机系统 ·····	(508)
26.1	数据库计算机系统·····	(508)
26.2	面向智能信息处理的多机系统及多媒体计算机系统·····	(513)

26.3 总结	(519)
第五篇 知识发现系统	(521)
第 27 章 知识发现系统的研究综述	(523)
27.1 KDD 的产生简介	(523)
27.2 KDD 的描述性定义	(524)
27.3 KDD 预处理	(526)
27.4 KDD 的应用	(529)
27.5 有关 KDD 软件的简介	(531)
27.6 KDD 研究和应用所面临的挑战	(534)
第 28 章 知识发现系统的机理研究	(536)
28.1 双库协同机制研究(I)	(536)
28.2 双库协同机制研究(II)	(549)
第 29 章 知识发现系统的模型结构研究	(564)
29.1 KDD* 系统	(564)
29.2 KD(D & K) 系统	(570)
29.3 ESKD 系统	(580)
第 30 章 数据发掘方法	(584)
30.1 概述	(584)
30.2 KDD 的可视化发掘技术	(586)
30.3 基于证据理论的发掘方法	(588)
30.4 基于概念树提升的发掘方法	(595)
30.5 基于神经网络的发掘方法	(598)
30.6 基于因果关系定性推理的发掘方法	(605)
30.7 基于统计的发掘方法	(609)
30.8 基于遗传算法的发掘方法	(616)
30.9 基于粗糙集的发掘方法	(629)
30.10 其它	(639)
第 31 章 知识评价方法及知识的组织、管理和维护	(663)
31.1 知识评价方法	(663)
31.2 可信度计算	(668)
31.3 知识的组织、管理和维护	(671)
第 32 章 KDD 原型及应用	(679)
32.1 KDD 原型系统	(679)
32.2 实例	(683)
参考文献	(686)

第一篇 知识工程的逻辑基础

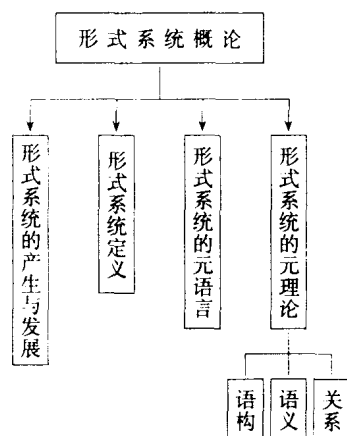
本篇的结构图如下所示：



第 1 章 形式系统概论

形式化实质上是一个算法,即一个可机械地实现的过程,用于将概念、断言、事实、规则、推演乃至整个被描述系统表述得严密、精确而又无须任何专门的知识即可被毫无歧义地感知。形式系统是理论系统或实际系统形式化的产物,在这种系统中所进行的推演均可被机械地测试,以确定是否是正确的。形式化(formalization)是现代逻辑学的基本特征,形式系统(formal system)是现代逻辑学的重要工具。形式系统和形式化方法是现代逻辑学的一个组成部分。

对于计算机科学技术的发展,形式化方法和形式系统始终有着巨大的影响。正是对计算机这一概念的形式化研究,导致了第一个计算模型——图灵机(Turing machine)的诞生,它被公认为现代计算机的祖先;一阶谓词演算形式系统为知识的形式表示及定理的机器证明铺平了道路;证实这种影响的一个有力证据是,被推举为第五代计算机程序设计语言的 PROLOG,就是一个典型的符号逻辑形式系统。本章概括介绍在知识工程中有着重要作用的逻辑系统赖以构建的理论基础——形式系统。本章的结构安排如下:



1.1 形式系统的产生与发展

公元前 5 世纪,古希腊的一些数学家发现等腰直角三角形的斜边与直角边不可约这一事实后,便把注意力从数学计算转移到数学推理上来,开始时只从一些最简单的概念出发,承认一些再显然不过的事实,使用极少数逻辑规则进行数学推演,在此基础上产生了第一个数学公理化系统——欧氏几何的公理系统。在欧氏几何中,原始概念正是现实世界中最简单的空间形态的刻画,公理和逻辑推理规则也是人们的直觉所意识到的正确的客观事实和思维规律的总结,从点、直线、平面等不加定义的原始概念出发,定义出另外一些更加复杂的概念,系统推演所得的定理继承了它们的客观性和正确性,因此像欧氏几何那样的公理系统

常被称为具体公理系统。

然而由于非欧几何的出现,人们感到具体公理系统过于受直觉的局限。似乎与直觉相悖的罗巴契夫斯基平行公理与欧氏几何公理并不冲突,罗巴契夫斯基几何与欧氏几何一样有着雄辩的正确性,也有令人信服的物理解释和实际应用。在 19 世纪末 20 世纪初,人们开始研究抽象公理系统。在抽象公理系统中,原始概念没有任何直觉意义,甚至没有任何预先设定的意义。

但是,即使在抽象公理系统中,仍然保留着直觉的成分,因为在这类系统的定理推演中,所使用的逻辑规则仍然是非形式的。这些规则之所以被运用,是由于它们在直觉上被认为是正确的,对它们的理解及运用方式都取决于人的直觉,而不是取决于规则的形式。如果将这最后的直觉成分也剔除,我们就得到所谓形式系统。在形式系统中,用于推演的逻辑规则也是形式地表述的,称为推理规则(inference rules)。同公理一样,推理规则的使用方式及使用结果,都完全取决于这些规则的形式,与逻辑直觉及人们对这些规则的意义所持的认识没有任何关系。

形式系统常用符号语言来表达。这一过程一旦完成,系统便和一切实际意义、甚至逻辑都毫不相干,留下的只是这种语言中的符号(symbols)——一种只作整体认读的记号(signs)、符号串(symbol strings)以及符号串的重写规则;对象、概念、运算等用符号和符号串来表示,公理只是特定的符号串,而推理规则正是符号串的重写规则,系统内的推演也只是对给定符号串的一系列重写而已。这里,决定一切的是符号、符号串及重写规则的形式。公理的识别、系统内的推演都可以依据公理及推理规则的形式机械地完成,不需要比认读和改写符号及符号串更多的本领和知识,甚至不需要逻辑。不难想象,具有这种特性的形式系统还可以用于逻辑学的形式化,而不必担心产生“循环”。当然,也只有具有这种特性的形式系统,才是毫无“悟性”的机器可以接受的。

从具体公理系统、抽象公理系统到形式系统,是一个不断发展的过程,这就跟距离空间一样,从 Banach 空间、Hilbert 空间到内积空间,都是由于现实世界的需要,而不是凭空捏造出来的。形式系统的提出,正是希望用符号语言来一般性地表达现实世界。正如列宁在《哲学笔记》中写道的:“当思维从具体的东西上升到抽象的东西时,它不是离开——如果它是正确的——真理,而是接近真理。物质的抽象、自然规律的抽象,价值的抽象及其它等等,一句话,那一切科学的(正确的、郑重的、不是荒唐的)抽象,都更深刻、更正确、更完全地反映着自然。从生动的直观到抽象的思维,并从抽象的思维到实践,这就是认识真理,认识客观实在的辩证的途径。”(列宁:《哲学笔记》155 页,人民出版社,1957 年版)。

1.2 形式系统的定义

从公理系统到形式系统的发展演变过程中,我们已经看到形式系统的轮廓。扼要地说,用符号语言表达的形式系统是一个符号体系,它由两大部分组成。第一部分是符号体系的组成部分,包括语言所使用的符号及各类符号串的形成规则。这一部分也称为形式系统的语言部分。第二部分是符号体系的推演部分,称为理论部分,包括称为公理的符号串集合,称为推理规则的符号串重写规则集合以及由它们重写生成的被称为定理(theorems)的那些符号串集合。更严谨一些,我们可以如下定义形式系统。

定义 1.1.1: 一个形式系统 FS 由以下 5 个部分组成:

- (1) 非空集合 Σ , 称为 FS 的符号表(alphabet), 其元素称为符号;
- (2) Σ 上全体字的集合 Σ^* 的一个子集 TERM, 其元素称为 FS 的项(terms), TERM 有子集 VARIABLE, 其元素称为变元(variables);
- (3) Σ^* 的一个子集 FORMULA, 其元素称为 FS 的公式(formulas)。FORMULA 有子集 ATOM, 其元素称为原子公式(atomic formulas);
- 公式集与项集不交, 即 $\text{TERM} \cap \text{FORMULA} = \Phi$, 而项和公式常统称为表达式(expressions);
- (4) FORMULA 的一个子集 AXIOM, 其元素称为 FS 的公理;
- (5) FORMULA 上的 n 元关系的集合 RULE, 即

$$\text{RULE} = \{r \mid \exists n (n \text{ 是正整数} \wedge n \geq 2 \wedge r \subseteq \text{FORMULA}^n)\}$$

其元素称为 FS 的推理规则。FORMULAⁿ 表示笛卡儿积:

$$\underbrace{\text{FORMULA} \times \text{FORMULA} \times \cdots \times \text{FORMULA}}_{n \uparrow}$$

因此, FS 可用五元矢 $\langle \Sigma, \text{TERM}, \text{FORMULA}, \text{AXIOM}, \text{RULE} \rangle$ 来表示。其中 $\Sigma(\Sigma^*)$, TERM, FORMULA 是 FS 的组成部分, 而 AXIOM, RULE 构成 FS 的演绎部分, 由它们出发可造就整个系统的全部定理, 即 FS 的理论, 可进行给定前提下的演绎。关于 FS 的定理、证明、理论及演绎的定义如下:

定义 1.1.2: 公式 A 称为 FS 的定理, 记为 $\vdash_{\text{FS}} A$ 如果存在一个有穷的公式序列

$$A_1, A_2, A_3, \dots, A_l (= A) \quad (1.1)$$

使得 $A_i (i=1, 2, \dots, l)$ 或者是 FS 的公理, 或者是由 $A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}} (j_1, j_2, \dots, j_{n-1} < i)$ 使用推理规则 r 得出的, 即存在 n 元关系(推理规则) $r \in \text{RULE}$, 有 $\langle A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}, A_i \rangle \in r$ 。此时称公式序列(1.1)为该定理 A 的证明, 并说 A 是 FS 中可证的。全体定理的集合称为 FS 的理论, 记为 $\text{Th}(\text{FS})$, $\text{Th}(\text{FS}) = \{A \mid A \in \text{FORMULA}, \vdash_{\text{FS}} A\}$ 。

注意, 公理显然为定理, 因为它们自己即为自己的证明。

定义 1.1.3: 设 Γ 为一公式集合, 称公式 A 是 Γ 推出的, 或称 A 是 Γ 的演绎结果, 记为 $\Gamma \vdash_{\text{FS}} A$, 如果存在有穷公式序列

$$A_1, A_2, A_3, \dots, A_l (= A) \quad (1.2)$$

使得 $A_i (i=1, 2, \dots, l)$ 或者是公理, 或者是 Γ 中的成员, 或者是由 $A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}} (j_1, j_2, \dots, j_{n-1} < i)$ 使用推理规则 r 得出的, 则公式序列(1.2)称为公式 A 的(由 Γ 出发的)演绎。 Γ 常称为演绎的前提。

显然, 对任意公式集 Γ , 公理、定理和 Γ 中的成员均为 Γ 的演绎结果。用 $\text{Th}(\text{FS} \cup \Gamma)$ 或简单地用 $\text{Th}(\Gamma)$ 表示 Γ 的全体演绎结果的集合, 即 $\text{Th}(\Gamma) = \{A \mid A \in \text{FORMULA}, \vdash_{\text{FS}} A\}$ 。此外, 如果公式 A 是 $\Gamma = \Phi$ 的演绎结果, 那么 A 是 FS 的定理。

对形式系统定义的进一步说明:

(1) 形式系统的项可看作是研究对象的抽象, 公式则是对这些对象的性质及关于这些性质讨论的抽象。形式系统的语言也称对象语言。项集和公式集通常用一些组成规则来确定, 这些规则常被称为该对象语言的词法。

(2) $\langle A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}, A_i \rangle \in r$ 表示利用规则 r 可从 $A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}$ 重写 A_i 。这里的 r 实际上是一个规则模式, 它的每一个成员都是一条具体的重写规则, 它们取完全相

同的格式。为直观和记忆的便利,可将规则表示如下:

$$\frac{A_{j_1}, A_{j_2}, \dots, A_{j_{n-1}}}{A_i}$$

(3) TERM, FORMULA, AXIOM, RULE 均可能是空集。

(4) 形式系统各组成部分及其成员的称谓并非系统的语言成分,它们是人们为了通信交流、讨论研究的方便而赋予形式系统的,它们属于另一层次的语言——元语言。

(5) 上述定义是一般性的,多数情况下要对形式系统作些限制,以便形式系统的推演可机械地实现。本篇将采用的限制是:

① Σ 为可数集(有穷集或可数无穷集),从而使形式系统语言的各组成部分均为可数集。

② TERM, FORMULA, AXIOM, RULE 及 RULE 的每一个成员 r 都是递归集,即存在一个算法,据此可判定任一表达式是否是项、公式、公理,可判定对 FORMULAⁿ ($n \geq 2$) 中任一 n 元矢 $\langle A_1, A_2, \dots, A_n \rangle$ 是否有 RULE 的一个成员 r , 使 $\langle A_1, A_2, \dots, A_n \rangle \in r$, 从而对任意公式序列 $B_1, B_2, \dots, B_l$, 可判定它是否是 B_l 的一个证明;是否是由 Γ 出发的 B_l 的一个演绎(当 Γ 为一递归集时)。

1.3 形式系统的元语言

形式系统本身所使用的语言(它的语言部分所规定的语言)叫做对象语言,它是一种形式语言。对形式系统的研究,在某种意义上就是对这一语言的研究。对象语言中的符号一方面是客观对象的抽象,另一方面它们作为符号客体又是被研究的对象——元理论中使用元语言进行讨论的对象。元语言是对形式系统及其语言进行研究时通讯、交流所使用的语言。通常使用的元语言是一种借助形式符号的自然语言。

元语言中的成分包括:

(1) 对形式系统各组成成分的称谓,例如术语:项、公式、公理等;

(2) 对系统分析讨论时所使用的逻辑术语,例如“当且仅当”、“如果……那么”、“所有”、“存在”等等;

(3) 描述形式系统有关性质的元语言术语,例如一致性、完备性、可判定性等等。

元语言中使用大量的形式符号,大体分为以下三类。

第一类:元语言中继续使用对象语言中的符号,但它是在不同意义上使用这些符号。对象语言是使用这些符号的客观实体,而元语言使用这些符号时,只是把它们看作它们自身的名来使用。

第二类:在元语言中要使用语法变元来表示对象语言中的一类符号或一类表达式。这些语法变元的使用,使公理、规则的交代简单明了,一个模式便交代了可数无穷多的公理或规则。

第三类:元语言也采用一些自身所需的符号来代替一些需反复使用的术语。例如 $\vdash_{FS}, \vdash_{FSPC}, \models, \models$, 他们的否定可表示为 $\nvdash_{FS}, \nvdash_{FSPC}, \not\models, \not\models$ 。 $\nvdash_{FSPC} A$ 表示 A 不是 FSPC 的定理, A 不可证; $\Gamma \not\models A$ 表示 A 不是 Γ 的演绎结果。

1.4 形式系统的元理论

元理论是研究形式系统时所取得的理论成果的总体。但我们可把元理论理解得更广泛

一些,它包括研究形式系统使用的理论工具以及研究形式系统的理论成果。前者主要指一些基本的数学原理和逻辑概念,后者将在后面再作说明。

使用数学工具和逻辑概念来研究关于数学或逻辑的形式系统,似乎有点循环,但其实不然。我们在元理论中使用的数学或逻辑只是其最基本的、直觉上毫无疑义的那一部分,但使用它们所研究的则是一个符号系统,它反映了数学或逻辑学的更为深刻的部分,乃至有关其整体的重要特性。特别是人们还要求元理论的概念是容易理解的,其推理是依直觉可进行的,要求元理论中所使用的工具和方法是简单的,即限于形式主义者所说的有穷性工具及方法,它们只接受直觉上可能的客体及可行的过程,拒绝把任何无穷集当作完备的整体,要求存在性证明是一个构造性证明。

元理论中关于形式系统的研究成果可分为三部分。一是关于形式系统语构(syntax)的研究。在这类研究中,形式系统只是一个没有具体意义的符号体系,被研究的是符号串的推演(重写)规律。二是关于形式系统的语义(semantics)的研究。在这类研究中,元理论赋予形式符号一定的意义,研究形式系统在被作出各种解释时的性质,特别是系统中公式的真值性质。三是关于形式系统语构与语义关系的研究,特别关注形式系统的整体特性,对系统作出评价。

1.4.1 元理论关于系统语构的研究

元理论关于系统语构的研究也称语法研究,是指把形式系统当作符号体系而做的研究。语构研究简单地把符号看作是不同种类的可识别的客体,而对其它的客体只注意它们由符号而组成的方式(结构),公理不过是满足特定模式的符号串,推理过程被看作为符号串依重写规则进行变换的过程。总之,语构研究中元理论关心的是形式系统的语言构成规律,并从这个角度来看系统的性质,例如系统的演绎特性(演绎定理、演绎等价性等)、系统的整体结构特性(独立性、一致性、可判定性等)。

定义 1.1.4:形式系统 FS 称为独立的,如果它的每一公理都是独立的,即若把任一公理 A 从 FS 中删去后,所得系统 FS' 不满足 $\vdash_{FS'} A$ 。

定义 1.1.5:形式系统 FS 称为一致的,或相容的,如果不存在 FS 的公式 A ,使得 $\vdash_{FS} A$ 与 $\vdash_{FS} \neg A$ 同时成立。

定义 1.1.6:形式系统 FS 称为完全的,如果对 FS 的任一公式 A ,或者是 $\vdash_{FS} A$,或者是 $\vdash_{FS} \neg A$ 。

定义 1.1.7:形式系统 FS 称为可判定的,如果存在一个算法对 FS 的任一公式 A ,可以确定 $\vdash_{FS} A$ 是否成立,否则称 FS 是不可判定的。如果上述算法虽不一定存在,却有一个过程,可对该系统的定理作出肯定的判断,但对非定理的公式过程未必终止,因而未必能作出判断。这时称 FS 为半可判定的。

很显然,FS 可判定当且仅当 $Th(FS)$ 为递归集;FS 半可判定当且仅当 $Th(FS)$ 为递归枚举集。

定义 1.1.8:形式系统 $FS1 = \langle \Sigma1, TERM1, FORMULA1, AXIOM1, RULE1 \rangle$ 称为形式系统 $FS2 = \langle \Sigma2, TERM2, FORMULA2, AXIOM2, RULE2 \rangle$ 的扩充,如果 $\Sigma2 \subseteq \Sigma1$, $TERM2 \subseteq TERM1$, $FORMULA2 \subseteq FORMULA1$, $RULE2 \subseteq RULE1$, $AXIOM2 \subseteq Th(FS1)$ 。

定义中不用 $AXIOM2 \subseteq AXIOM1$ 而用 $AXIOM2 \subseteq Th(FS1)$,是因为我们希望定义更为广泛一些,因而可能有 $FS2$ 的扩充 $FS1$, $AXIOM2 \not\subseteq AXIOM1$,但由于 $RULE1$ 足够地大,仍

能使 $\text{AXIOM2} \subseteq \text{Th}(\text{FS1})$ 。

定义 1.1.9: 称 FS 的公式集 Γ 是一致的, 如果 FS 是一致的, 并且它的扩充 $\langle \Sigma, \text{TERM}, \text{FORMULA}, \text{AXIOM} \cup \Gamma \rangle$ 是一致的。

今后把这种仅增加公理的 FS 的扩充简记为 $\text{FS} \cup \Gamma$, 将 $\text{Th}(\text{FS} \cup \Gamma)$ 简记为 $\text{Th}(\Gamma)$ 。

不难明白, $\text{Th}(\Gamma) = \{A \mid \vdash_{\text{FS}} A\}$, 而 Γ 为一致的当且仅当不存在 FS 的公式 A , 使得 $\Gamma \vdash_{\text{FS}} A, \Gamma \vdash_{\text{FS}} \neg A$ 同时成立。

定义 1.1.10: FS 的公式 A, B 称为演绎等价的, 如果 $A \vdash_{\text{FS}} B$ 且 $B \vdash_{\text{FS}} A$, 记为 $A \vdash_{\text{FS}} \neg B$ 。

1.4.2 元理论关于系统语义的研究

虽然抽象公理系统、形式系统并不针对某一特定的客体范畴、特定的性质关系, 但可以对它作出种种解释——赋予它一定的论域, 即研究对象的集合; 赋予它一定的结构, 即用论域中的客体、客体上的运算、客体间的关系去解释系统中的抽象符号。我们把上述赋予形式系统的论域及解释称为形式系统的语义结构, 结构连同该结构下公式所取真值的规定, 称为系统的一种语义。语义的规定方式不止一种, 上述语义规定称为指称语义, 通用于逻辑形式系统。

定义 1.1.11: 形式系统 FS 的结构 $\mathfrak{U}$

(1) 当 FS 的 $\text{TERM} \neq \Phi$ 时, 由下列集合 U 及规则组 I 所组成 (即 $\mathfrak{U} = \langle U, I \rangle$):

① 非空集合 U , 称为论域, 也称个体域。

② 一组规则 I , 称为解释, 据此规定如何对项 (TERM 的成员) 指称 U 中个体的性质 (或 U 的子集)、关系 (或 U^n 的子集)。

(2) 当 FS 的 $\text{TERM} = \Phi$ 时, 结构 $\mathfrak{U}$ 为一映射 v

$$v: \text{ATOMIC} \rightarrow \text{VALUE}$$

v 常称为赋值映射, 这里 VALUE 为一确定的集合, 至少含有成员 0, 1, 它们表示真值假和真, VALUE 仍称为真值集合, 但由于它被允许包含 0, 1 以外的成员, 因而可以是更广泛的真值集合。

定义 1.1.12: 若形式系统 FS 的 $\text{VARIABLE} \neq \Phi$, 那么下列映射称为指派:

$$s: \text{VARIABLE} \rightarrow U$$

当给定结构 $\mathfrak{U}$ 后, 对每一指派 s 可作出其扩展 $\bar{s}: \text{TERM} \rightarrow U$, 使

$$\bar{s}(t) = \begin{cases} s(t) & \text{当 } t \text{ 为变元} \\ s \text{ 指派 } t \text{ 中变元后由解释 } I \text{ 确定} & \text{当 } t \text{ 非变元} \end{cases}$$

定义 1.1.13: 赋值是指一组给公式赋真值的规则, 据此规则可对每一结构 $\mathfrak{U}$ 指派 s 确定一映射 $v: \text{ATOMIC}_{\mathfrak{U}, s} \rightarrow \text{VALUE}$ (当 $\text{TERM} = \Phi$ 时, 该映射即结构 $\mathfrak{U}$ 所确定的赋值映射), 并据此规则扩展 v 为赋值函数 $\bar{v}: \text{FORMULA}_{\mathfrak{U}, s} \rightarrow \text{VALUE}$, 使

$$\bar{v}(A^{\mathfrak{U}}[s]) = v(A^{\mathfrak{U}}[s]) \quad \text{当 } A \in \text{ATOMIC 时}$$

注意: (1) 结构 $\mathfrak{U}$ 是任意给出的, 而对任一结构 $\mathfrak{U}$, 指派 s 又可任意规定, 但赋值作为一组规则是不依赖于 $\mathfrak{U}, s$ 的, 尽管由它确定的赋值映射 v 可能依赖于 $\mathfrak{U}, s$ 。因此, 每一组 $\mathfrak{U}, s$ 连同这组赋值规则所确定的赋值映射 v 及赋值函数 $\bar{v}$ 构成了 FS 的一个语义。(2) 集合 VALUE 通常是一类语义结构所公用的。限定 $\text{VALUE} = \{0, 1\}$ 的语义结构称为二值结构, 限定二值结构为其语义的逻辑系统常称为二值逻辑系统。