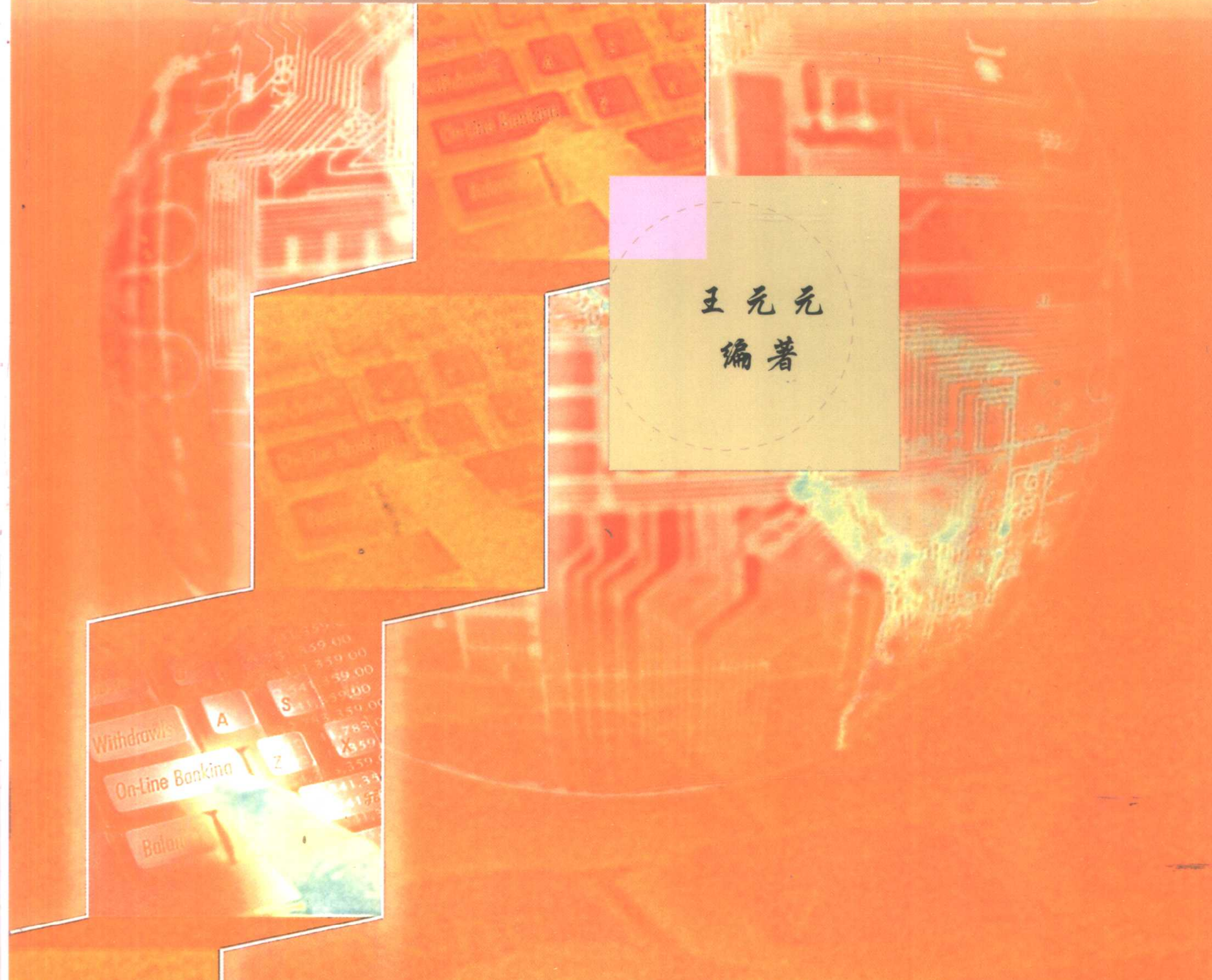
计算机科学中的现代逻辑学

王元元
编著



科学出版社

计算机科学中的现代逻辑学

王元元　编著

科学出版社
2001

内 容 简 介

1989年科学出版社出版了《计算机科学中的逻辑学》，该书于1992年获国家优秀教材奖，1997年被国家教委列为“九五”规划重点发展教材。根据教材建设的需要，借本次重印之机，对原书的内容进行了增补，其主要内容包括：形式化和形式系统概论，命题演算形式系统，各种形式的一阶谓词演算形式系统，直觉主义的一阶谓词演算形式系统，各种形式的模态逻辑系统，时态逻辑系统和动态逻辑系统，多值逻辑系统和模糊逻辑系统，非单调逻辑推理系统，λ-演算和组合逻辑，以及这些经典的和非经典的逻辑系统在计算机科学技术中的应用。

本书可用作计算机专业、数学专业的本科生、硕士研究生的数理逻辑、计算理论等课程的教材或教学参考书。

图书在版编目(CIP)数据

计算机科学中的现代逻辑学/王元元编著.-北京:科学出版社,2001
ISBN 7-03-001067-1

Ⅰ.计… Ⅱ.王… Ⅲ.①数理逻辑-基本知识 ②逻辑运算 Ⅳ.TP 301.6

中国版本图书馆CIP数据核字(2001)第042030号

科学出版社 出版
北京东黄城根北街16号
邮政编码:100717
http://www.sciencep.com
双青印刷厂 印刷
科学出版社发行 各地新华书店经销
*
1989年9月第 一 版 开本：787×1092 1/16
2001年9月第二次印刷 印张：17 1/4
印数：3 201—7 200 字数：405 000

定价：28.00元

(如有印装质量问题,我社负责调换〈环伟〉)

前　言

1989 年科学出版社出版了《计算机科学中的逻辑学》,本次重印,更名为《计算机科学中的现代逻辑学》。书名中的“现代逻辑”一词指数理逻辑(mathematical logic),或符号逻辑(symbolic logic)。对“现代逻辑”的理解通常有两种,一种是狭义的理解,它指“数理逻辑基础”,即所谓两个演算(命题演算与一阶谓词演算)以及其它一些相关的逻辑系统;另一种是广义的理解,它指数学的一个分支,包括数理逻辑基础及建于其上的“公理化集合论”(axiomatical set theory)、“证明论”(prof theory)、“递归论”(recursion theory,也称可计算性理论)、“模型论”(model theory)。我们取第一种理解,但内涵更广泛些。我们的讨论将涉及两个演算及许多在计算机科学中有直接应用的符号逻辑系统,包括所谓经典的与非经典的两个演算的扩张。

近年来,现代逻辑学在计算机科学中的重要地位越来越被计算机专家所认识。著名的计算机软件设计大师戴克斯特拉(E. W. Dijkstra)曾经这样说:“我现在年纪大了,搞了这么多年软件,错误不知犯了多少,现在觉悟了。我想,假如我早年在数理逻辑上好好下点功夫的话,我就不会犯这么多的错误。不少东西逻辑学家早就说了,可我不知道。要是我能年轻 20 岁的话,就要回去学逻辑。”[1]我国著名数理逻辑学家莫绍揆教授甚至说得更加直截了当:“事实上,它们(程序设计)或者就是数理逻辑,或者是用计算机语言书写的数理逻辑,或者是数理逻辑在计算机上的应用。”[2]因此,在计算机科学界加强现代逻辑学知识的普及与应用,实在是十分紧迫和重要的。

但是,许多从事计算机科学基础研究和教学的同仁都有这样一种感觉,目前相当缺乏介绍上述领域知识的适当书籍。关于现代逻辑学的书籍或者过于简单(常被看作离散数学的一个部分[3]);或者过于专业化,其深度恐难以接受,其广度却又难以满足计算机界学者的要求,因为它们通常不涉及非经典的逻辑系统,而这些系统在计算机科学中有着广泛的应用。

本书的写作和修订正是受了上述这种感受的驱使,希望它能弥补一点计算机科学基础研究和教学方面的缺憾,为读者提供一点方便,掌握或多少了解一点现代计算机技术飞速发展的理论背景的一个重要方面。

全书以一阶谓词演算及其在计算机科学中的应用为核心来展开。第一章作为引论介绍形式化方法、形式系统概念,使读者对现代逻辑学的研究思想有一个概貌,从而了解人与机器交流的基本途径(这一章的阅读也可以放在读完二、三两章之后再进行)。第二章作为准备照例先介绍命题演算系统。当然,它也是形式化方法的一个简明的释例。第三、四、五、六章全面介绍一阶谓词演算的种种形式系统,介绍它们在程序语义描述、知识表示、逻辑程序设计等方面的应用。其余各章(除最后一章)讨论一阶谓词演算的扩张系统。这些扩张系统通常分为两类。一类是从语构的角度扩张。例如,引入谓词变元、函数变元和高阶量词的二阶谓词演算系统,将在第七章介绍;引入模态词的模态逻辑正规系统、时态逻辑系统、动态逻辑系统,分别在第八、九章讨论;近期时髦的非单调逻辑系统专用一章

(第十一章)加以综述。另一类扩张则是从系统语义的角度进行的,大多以扩张“真值”(真、假)概念为基本出发点,第十章专题介绍相关的多值逻辑系统。第十二章“λ-演算与组合逻辑”是一个例外,它很难列入谓词演算的扩张,倒应该算作谓词演算的归约,或列为递归论的一个主题。但是,由于它在计算机科学中的重要意义[作为可计算性(computability)、计算复杂性(computational complexity)以及函数式程序设计(functional programming)语言的基础],我们决意收入本书。

由于本书在许多地方使用了集合论的概念、术语和记号,因此要求阅读本书的读者具有集合论基础知识。如果读者已预修“离散数学”[3]教程,那么阅读起来会更容易些。除第一章和第十一章外,各章都备有一些习题供读者练习,其难度均不很大。

莫绍揆教授对本书的编写给予了热情的支持与帮助,并对一些章节提出了宝贵的修改意见。南京大学吕义忠教授也审阅过全部初稿。汤律庆先生为全书的录入和校正作出了辛勤的劳动。作者谨向他们表示由衷的谢意。

由于作者水平所限,疏漏之处实恐难免,尚希各界同仁、学者不吝赐教。

作　者

2001年8月

序　言

电子数字计算机与数理逻辑具有非常密切的关系。正是在数理逻辑中，把人类的推理过程分解成一些非常简单原始的、非常机械的动作，才使得用机器代替人类推理的设想有了实现的可能。随着电子技术的发展，出现了电子数字计算机。在使用电子计算机时，必须先进行程序设计，把整个推理、计算的过程，丝毫不漏地考虑到，统统编入程序，而机器则依此而运行；如稍有错误，将立即得到毫无意义的错误结果。可见，必须有足够的数理逻辑训练，熟悉推理过程的全部细节，才能从事程序设计。此外，程序设计是一个很细致而又很麻烦的工作，如何设计正确的程序，如何防止在计算过程中出现错误，如何很快地发现这种错误而及时加以改正，都是程序设计理论（软件理论）中的非常根本而又非常重要的内容。大家都认为，这些内容都与数理逻辑息息相关。事实上，它们或者就是数理逻辑，或者是用计算机语言书写的数理逻辑，或者是数理逻辑在计算机上的应用。戴克斯特拉所说的一段话（见本书前言所引），可以说代表了软件理论家的肺腑之言，也道出了数理逻辑与计算机的密切关系。

通常把计算机中用到的数理逻辑（以及别的一些数学理论）放在离散数学中，但其中写得极少，很难满足计算机工作者的需要。《计算机科学中的逻辑学》一书，不但详细地叙述了数理逻辑的基本内容，而且详细地介绍了针对计算机的应用而发展出来的各种逻辑（通常叫做非经典逻辑）——模态逻辑、时态逻辑、动态逻辑、模糊逻辑、非单调逻辑等等，其中虽然有些是最新的发展，它们在计算机上的作用及理论本身还未有定论，但是读者知道这么多的新内容后，对计算机科学肯定会有进一步的了解，而有助于对计算机科学技术的研究。

相信《计算机科学中的逻辑学》一书将对我国计算机科学理论做出有益的贡献。值本书出版之际，特志数语以作介绍。

莫绍揆

1988 年 3 月

目　录

第一章　绪　　论

我们知道,**逻辑学**(logic)是研究人类思维规律的科学,而现代逻辑学则是用数学(符号化、公理化、形式化)的方法来研究这些规律。何谓"思维"? 何谓"符号化、公理化、形式化"? 这正是绪论欲论者。

1.1　思维:感知的概念化和理性化

就现时人类对大脑机理的研究水平而言,尚无力从机理的角度回答"什么是思维"的问题,我们只想从功能分析出发,对思维的过程作一番揣摩与探讨。

思维实体(entity)处于一个客观世界,称为该实体的环境(environment),通过对环境的感知(perception)形成概念(concepts)。这些概念以自然语言(包括文字、图像、声音等)为载体,在思维实体中记忆、交流,从而又成为这些思维实体的环境的一部分。通过对概念外延(extension)的拓广和对概念内涵(intension)的修正,完成思维的最基础的功能——**概念化**(conceptualization)。这一过程将物理对象抽象为思维对象(语言化了的概念),包括对象本身的表示、对象性质的表示、对象间关系的表示等。例如,我们的祖先通过对日月星辰的长期观察,有了"日、月、天、地"等概念,有了"天圆地方"的概念,有了"日出东方、日落西方"的概念;后又随这些概念的修正,逐步获得正确的认识。

在概念化的基础之上,思维进入更加高级的层次——**理性化**(rationalization)思维,即对概念的思维:**判断**(judgments)与**推理**(reasoning)。判断包括:概念对个体(individuals)的适用性判断(特称判断、全称判断及其否定),个体对多个概念同时满足或选择地满足的判断(合取判断或析取判断),概念对概念的蕴涵的判断(条件判断)等等。推理可说是对概念、判断的思维,即由已知的判断根据一定的准则导出另一些判断的过程。这些准则是思维主体对自身思维属性感知并概念化的产物。它们中包括逻辑学家所称的"三段论"(由大前提、小前提得出结论)、假言推理(由条件判断"A 蕴涵 B"和判断 A 得出判断 B)等等。例如,以全称判断"所有的人都是会死的"为大前提,以特称判断"苏格拉底(Sokrates)是人"为小前提,可用三段论导出判断"苏格拉底是会死的"。又如,由条件判断"如果苏格拉底是柏拉图(Platon)的老师,那么柏拉图是苏格拉底的学生"和判断"苏格拉底是柏拉图的老师",导出判断"柏拉图是苏格拉底的学生",是一种假言推理。

因此,大体可以这样说:思维是感知的概念化和理性化。逻辑学,特别是现代逻辑学的宗旨,便是用符号化、公理化、形式化的方法来研究这种概念化、理性化过程的规律与本质。

1.2　现代逻辑学求助数学——符号化

数学的符号化是人所共知的。所谓符号化即是用"一种只作整体认读的记号

(sighs)”——符号(symbols)表示量、数及数量关系。

思维的概念化过程离开语言显然是难以完成的,语言是一种符号体系,语言化是符号化的初级阶段,但若要对思维作深入的讨论和研究,这种初级的符号化是不够充分的,现代逻辑除求助数学对思维过程作符号化的探讨之外,别无它路。我们知道,数字0,1,2,3,……是由人类的基数、序数概念符号化而来,但只是在有了“字母表示数”、“符号表示数的运算、关系”之后才有代数理论,才有人们对数的概念的深刻认识。完全相同,现代逻辑学对思维的研究,需要更加彻底的符号化过程。我们也用字母、符号表示思维的物理对象、概念对象、判断对象等。例如,对图1.1中的情景,用table表示图中的桌子,a,b,c,d,e表示桌上的积木块;用

On(d,table)	表示“积木d在桌子上”
On(b,c)	表示“积木b在积木c上”
$\neg$ On(d,c)	表示“积木d不在积木c上”
On(a,b)$\wedge$On(b,c)	表示“积木a在b上而且积木b在c上”
wang=owner(table)	表示“王先生是桌子的主人”

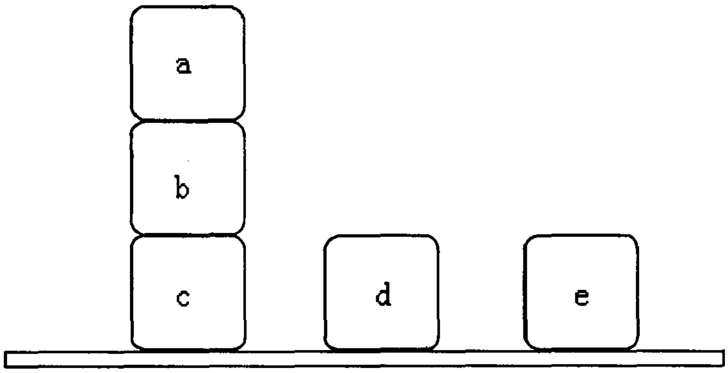


图1.1

表1.1将数学中的符号化与现代逻辑学中的符号化通过例子进行对照,读者自会发现,后者甚至更为彻底,这是现代逻辑还必须公理化和形式化的需要。(读者不必为表中生疏的符号而苦恼,不久我们便要系统地学习它们。)

表1.1

	数学	现代逻辑学
概念	x,y,z (数) $\sqrt{x}$ (函数) x≤y (关系)	x, y, z (任意对象) owner(x) (函数:x的主人) On(x,y) (关系:x在y上)
判断	如果a≤b,则a+c≤b+c	On(a,b)→$\neg$ On(b,a) (如果a在b上,则b不在a上)
推理	由“如果a≤b,则a+c≤b+c” 以及“a≤b”可推得“a+c≤b+c”	{On(a,b)→$\neg$ On(b,a),On(a,b)} $\vdash\neg$ On(b,a) (由On(a,b)→$\neg$ On (b,a)和On(a,b) 可推得$\neg$ On(b,a))

1.3 现代逻辑学追随数学——公理化

我们从数学的**公理化**(axiomization)说起。

由于等腰直角三角形的斜边与直角边不可通约这一事实被发现,公元前五世纪,古希腊的一些数学家便把注意力从数学计算转移到数学推理上来。他们感到,直觉不是绝对可靠的,接受未经证明的事实是危险的,正如他们的前人错误地接受"一切数可以表示为整数之比"所表明的那样。于是,他们开始只从一些最简单的概念出发,只承认一些再显然不过的事实,只使用极少数逻辑规则进行推演。这种小心翼翼的做法,酝酿着一个正确的思想——公理化思想。不久,第一个数学公理化系统在欧几里德(Euclid)的《几何原本》中出现了。从此,**公理系统**(axiomatic systems)成为数学研究的有力工具。

欧氏几何(Euclidean geometry)的公理系统,从点、直线、平面等不加定义的原始概念(primitive concepts)出发,定义另外一些更为复杂的概念,例如平行线、三角形、平行四边形等等。它接受一些所谓自明的事实作为公理(axioms)不予证明,例如"两点确定一条直线"。它运用很少几条逻辑推理规则,例如"假言推理",推导出平面几何学的诸多定理,这些推理规则也被认为是毋庸置疑的。

在欧氏几何中,原始概念正是现实世界中空间形态基础成分的概念化,公理和逻辑推理规则则是对空间形态最基本属性以及人类思维规律概念化、理性化的结果,因而系统推演所得的定理继承它们的客观性和正确性。欧氏几何公理系统中的所有概念都有鲜明的直观背景,其公理、定理也都有强烈的客观意义。像欧氏几何这样的公理系统,常被称为**具体公理系统**(concrete axiomatic systems)。

创始于数学全盛时期(18 世纪~19 世纪)的现代逻辑学不可能不受到公理化思想的影响,人们开始探求思维的规律,并努力使之公理化。始于亚里士多德(Aristotle)的逻辑学被符号化、公理化,逐步演化为现代逻辑学。例如众所周知的思维法则"一个条件命题等价于它的逆否命题","全称判断蕴涵特称判断"可以表示为如下的**公理模式**(Axiom schema):

$$(A\to B)\leftrightarrow(\neg B\to\neg A)$$
$$\forall xA(x)\to A(t)$$

其中$\leftrightarrow$表示"等价",$\forall xA(x)$表示"一切对象皆满足性质 A",而 $A(t)$表示"对象 t 满足性质 A"。

事实上,现代逻辑学的公理化也更为彻底,它将人们的推理规则也符号化和模式化,它们本质上和公理相同,但为了突出它们在形式上和应用上与公理的区别,称为推理规则模式(inference rules schema)。例如假言推理规则可以表示为如下的规则模式:

$$\frac{A\to B,A}{B}$$

1.4 现代逻辑学改造数学——形式化

由于更加彻底地符号化与公理化的现代逻辑学的萌生,也由于数学公理化进程出现

了从具体公理化到抽象公理化的趋势，使数学和现代逻辑学同时步入形式化的新时代。

非欧几何(non-Euclidean geometry)的出现，使人们感到具体公理系统过于受直觉的局限，因为他们看到，与直觉似乎相悖的罗巴契夫斯基(Lobachevsky)平行公理(过直线外一点至少可作两条直线与已知直线平行)，同欧氏几何公理(除平行公理)并不冲突，并且在由它们组成的公理系统上建立起来的罗巴契夫斯基几何(一种非欧几何)有同欧氏几何一样雄辩的逻辑正确性，也有令人信服的物理解释和实际应用。因而，19 世纪末和 20 世纪初，一些杰出的数学家、现代逻辑学家开始了**抽象公理系统**(abstract axiomatic systems)的研究。

在抽象公理系统中，原始概念的直觉意义被忽略，甚至没有任何预先设定的意义。不加证明而接受的断言——公理，也无需以任何实际意义为背景，它们无非是一些形式约定——一些符号串，约定系统一开始便要接受为定理的是哪些语句。对原始概念和公理，人们甚至可以不知所云，唯一可识别的是它们的表示形式，这也是它们唯一有意义的东西。希尔伯特(Hilbert)的《几何基础》给出了一个“几何”抽象公理系统，欧氏几何可以说是这个系统的“一种说法”、“一个释例”。在那本书里，希尔伯特称“我们必须能够不用点、线、面，而说桌子、椅子、啤酒杯”，以此强调，几何的空间形态的意义在这里已经消失，它们不再是公理系统的依托；概念的涵义全部由公理的形式来赋予，系统内的定理完全被公理的形式和允许使用的逻辑推理规则所决定，几何直觉在此毫无立足之地。例如，可以用(两个集合)L 和 P 表示两个原始概念(对应于直线和点的概念)，用(L 到 P 的一个二元关系)R 表示另一个原始概念(相应于某直线过某点的概念)。当然，这里的术语“集合”、“关系”等的引入只是为了叙述的可读性，并非抽象公理系统中必需的东西。现在，公理“两点确定一条直线”可表述为无任何直观可言的抽象符号串：

$$\forall x \forall y(x \in P \wedge y \in P \to \exists!\, l(l \in L \wedge R(l,x) \wedge R(l,y))$$

可读作：对 P 中任意 x，y，有且仅有 L 中 l，使 l 与 x，y 同时有 R 关系。

应当承认，抽象公理系统的提出往往是有客观背景的，常常是因为现实世界的某些对象及其性质需精确地刻画、深入地探究。但是，抽象公理系统一旦建成，它便应当是超脱客观背景的，它可刻画的对象已不限于原来考虑的那些对象，而是与它们有着(公理所规定的)共同结构的相当广泛的一类对象，因而对它们性质的讨论也必定深刻得多。因此，对一个抽象公理系统，一般会有多种解释(释例)。例如，布尔代数抽象公理系统，可以解释为有关命题真值的命题代数，有关电路设计研究的开关代数，也可以解释为讨论集合的集合代数。在这些释例中，抽象符号 x 分别被看作“命题 x”、“电路 x”、“集合 x”等；0 和 1 分别被用于表示“假和真”、“闭和开”、“空集和全集”。

所谓**形式化**(formalization)，就是彻头彻尾的“符号化 + 抽象公理化”。因此，现代逻辑学在形式化数学的同时，完成了自身的形式化。综上所述，现代逻辑学形式系统如下组成：

(1) 用于将概念符号化的符号语言，通常为一形式语言(formal languages)，包括一符号表 Σ 及语言的文法，可生成表示对象的语言成分项(terms)，表示概念、判断的公式(formulas)。

(2) 表示思维规律的逻辑学公理模式和推理规则模式(抽象公理系统)，及其依据它们推演可得到的全部定理组成的理论体系。

建基于现代逻辑学可构成形式化的数学系统或其它理论系统，它们与现代逻辑学系统不同的只是

(1) 表示对象更为广泛的形式语言。

(2) 抽象公理系统中还包括对象理论(例如数论)的公理——非逻辑学公理(例如数论的皮亚诺(Peano)五公理)。

因此可以这样说:形式化是现代逻辑学的基本特性，**形式系统**(formal systems)是现代逻辑学的重要工具，借助于形式化过程和对形式系统的研讨完成对思维规律或其它对象理论的研究。那么，人们是如何对形式系统进行研究的呢?

显然，首先是对系统内定理推演的研究。哪些是系统内的定理? 如何更快地导出这些定理? 定理之间有怎样的本质联系? 由于推理规则本质上是一种符号串的重写规则(rewrite rules)，系统内的推演也只是对给定符号串的一系列重写而已，从而决定一切的是符号、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。因此，这类研究被看作是对形式系统的语构(syntax)的研究。

我们曾经提到，抽象公理系统、形式系统并不一定针对某一特定的问题范畴，但可以对它作出种种解释——赋予它一定的个体域，即研究对象的集合;赋予它一定的结构，即用个体域中的个体、个体上的运算、个体间的关系去解释系统中的抽象符号。我们说这一过程赋予形式系统一个语义结构，或简称**语义**(semantic)。在给定语义结构中可以讨论形式系统中项所对应的个体，公式所对应判断具有的真值(真，假)。对语义的规定及对形式系统在给定语义下的讨论，便是所谓对形式系统的语义的研究。

由于语义结构通常是抽象出形式系统的那个问题范畴(例如抽象出现代逻辑形式系统的问题范畴是“人类思维”)的数学描述，因此一个好的形式系统中的定理，应当都是在所有相关语义中的真命题;反之，所有这些真命题所对应的形式表示，应当都是形式系统的定理。诸如此类的讨论，可视为对形式系统语构与语义关系的研究。

我们对现代逻辑学的介绍包括:①现代逻辑形式系统。②对现代逻辑形式系统作以上三个方面的研究。正因为如此，两个层次的问题需要加以注意。

我们将使用两种语言。一种是现代逻辑形式系统本身所使用的那个形式语言，有人称之为**对象语言**(object language)，另一种是介绍和研讨这个形式系统时使用的语言，为通常所用的数学语言，常称为**元语言**(meta language)。后者也用大量符号，包括①沿用形式系统的符号，②表示形式系统中同一类符号的符号，称为**语法变元**(syntactic variables)，③为表达元语言概念引入的新符号。

呈现于读者面前的有两个理论。一个是现代逻辑形式系统内的公理、推理规则及其由它们导出的定理所构成的那个逻辑学理论;另一个是对这个形式系统进行研究所得的关于这个系统的性质的定理(称为**元定理**(meta theorems))所组成的理论，称为**元理论**(meta theory)。

1.5 现代逻辑学与计算机科学

本书前言中所引戴克斯特拉教授与莫绍揆教授之言，已深刻提示现代逻辑学与计算

机科学之密切关联。通过以上讨论读者也不难看到，两者在以下三个根本方面上又是何等地一致。

现代逻辑学和计算机科学具有完全相同的宗旨：扩展人类大脑的功能，帮助人脑正确、高效地思维。它们是一对天生的同盟军，分别作战在基础理论和实用技术两条战线。

现代逻辑学试图找出构成人类思维或计算的最基础的机制，例如推理中的“代换”、“匹配”、“分离”，计算中的“运算”、“迭代”、“递归”。而计算机程序设计则是要把问题的求解归结于程序设计语言的几条基本语句，甚至归结于一些极其简单的机器操作指令。

此外，现代逻辑学的形式化方法又和计算机科学不谋而合。计算机系统本身，它的硬件、软件都是一种形式系统，它们的结构都可以形式地描述；程序设计语言更是不折不扣的形式语言系统。要研究计算机、开发种种程序设计语言，没有形式化知识和形式化能力是难以取得出色的成果的。另一方面，应用计算机求解实际问题，首要的任务便是形式化。离开对问题正确的形式化描述，没有理性的机器何以理解、解答这些问题呢？人们必须用计算机懂得的形式语言告诉它“怎么做”或者“做什么”，而计算机理解这些语言的过程，又正是按照人赋予它的形式化规程（编译程序，compiler），将它们归约为自己的基本操作。

计算机科学技术人员常常会发现，一个问题的逻辑表达式，几乎就是某个程序设计语言（例如逻辑程序设计语言 Prolog）的一个子程序；而用有些语言书写的程序（例如关系数据库查询语言 SQL 程序）简直就是逻辑表达式。事实上，正是现代逻辑学对“计算”的追根寻源，导致了第一个计算的数学模型——图灵机（Turing machines）的诞生[4]，它被公认为现代数字计算机的祖先；λ-演算系统为第一个人工智能语言 LISP[5]奠定了基础；一阶谓词演算系统为计算机的知识表示及定理证明铺平了道路[6]，以其为根本的逻辑程序设计语言 Prolog，曾被不少计算机科学技术专家誉为新一代计算机的核心语言。

作为数学分支的现代逻辑学在 20 世纪末风靡世界，并不因为数学界的推崇，却完全是由于计算机科学界的青睐。目前，从基本逻辑电路的设计，到巨型机、智能机系统结构的研究；从程序设计过程到程序设计语言的研究发展；从知识工程到新一代计算机的研制，无一不需要现代逻辑学的知识、成果，无一可离开现代逻辑学家的智慧与贡献。

第二章　命题演算形式系统

在传统逻辑中，是从低级到高级，即从概念到判断再到推理来讨论人的思想规律的，这未必是一种最好的安排。事实上，当我们把推理作为研究的根本目标时，先忽略判断的细节——概念，把判断看作不可分的整体——命题来讨论，更便于对推理规律进行分析；在此基础之上，再引入概念的形式表示——谓词，把对推理的研究引入更加深入的层次，会显得格外顺理成章。本章的阐述将遵循这一次序，先讨论命题、命题演算及其形式系统。

2.1　命题演算基本概念

2.1.1　命题与联结词

命题(propositions)是指对事物作出确定判断的陈述语句，当判断合理或符合事实时，常称该命题**真**(true)，否则称该命题**假**(false)。真、假又称命题的**真值**(truth values)，为简便我们用 1 和 0 分别表示真、假这两个真值。在一些非经典的逻辑系统中，对真值概念作了扩充，这是后话。

例 2.1　考虑下列语句：

(1) 雪是白的。

(2) 2+2=5。

(3) 2 是质数又是偶数。

(4) 陈胜、吴广起义之日杭州下雨。

(5) 大于 2 的偶数均可分拆为两个质数的和(哥德巴赫猜想)。

(6) 您上哪儿去?

(7) x+y<0。

(8) 我说的这句话(例 2.1 之(8))不对。

显然(1)(3)是真命题，(2)是假命题。(4)(5)虽不知真、假，但确是命题。(6)不是陈述句；(7)含有变元 x，y，不是确定的判断；(8)则是一个病态的语句——悖论(paradox)，它对自身作了否定，因此，(6)(7)(8)不是命题。

注意，上例中(3)与其它命题不同，它实际上是由两个判断联结而成的，即“2 是质数”并且“2 是偶数”，其中“并且”是一个联结词。这一命题的真值不仅依赖于这两个组成它的命题，而且还依赖于这个联结词的意义。像这样的联结词称为**真值联结词**(truth value connectives)。通常把不含有真值联结词的命题称为**原子命题**或**原子**(atoms)，其它命题称为**复合命题**(composite propositions)。

例 2.2　下列命题都是复合命题，其中黑体字为真值联结词：

(1) 雪不是白的(**并非**“雪是白的”)。

(2) 今晚我去看朋友**或者**去看电影。