可计算性与不可解性

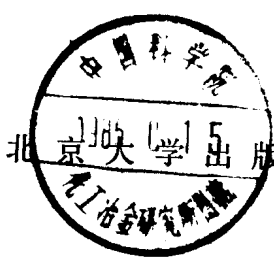
[美] M. 戴维斯 著

北京大学出版社

可计算性与不可解性

〔美〕M.戴维斯 著

沈 泓 等译 吴允曾 校



北京大学出版社

内 容 简 介

本书是一本数学系和计算机系研究生教材。第一至五章为可计算性理论的基本内容，可作为数学系和计算机科学系大学生教材或参考书。第六至八章为这一理论在代数、数论及逻辑中的应用。第九至十一章为可计算性理论的专题。

本书也可供有关研究工作者参考。

MARTIN DAVIS
Computability and Unsolvability
McGRAW-HILL, 1958

可 计 算 性 与 不 可 解 性

北 京 大 学 出 版 社 出 版
（北京大学校内）
新华书店北京发行所发行
北 京 大 学 印 刷 厂 印 刷
787×1092毫米 32开本 9.25印张 200千字
1984年8月第一版 1984年8月第一次印刷
印数：1—32,000册

统一书号：13209·89 定价：0.95元

译者前言

本书作者M.戴维斯现任美国纽约州立大学计算机科学系教授，系著名的数理逻辑学者。他曾和H.Putnam, J.Robinson, Yu.Matijasevic等三人一起解决了希尔伯特第十问题。

本书可作为数学系、计算机系研究生教材。全书共分十一章。第一章至第五章为可计算性理论的基本内容，也可作为数学系和计算机科学系大学生教材或参考书；第六章至第八章为这一理论在代数、数论及逻辑中的应用；第九章至第十一章为可计算性理论的专题。

在此译本即将付印时，我们看到了此书的第三版（1982年），第三版与前两版的不同之处是增加了附录二“希尔伯特第十问题是不可解的”，用来取代第七章。有兴趣的读者可参阅“Hilbert's Tenth Problem Is Unsolvable”(The American Mathematical Monthly, Vol.80, No.3, March 1973, pp.233—269)。

由于译者水平有限，译文中肯定有不少缺点和错误，敬请读者批评指正。

参加本书翻译的有：沈泓（引言、第一章），戴伟长（第二、十、十一章），董亦农（第三章），陈安捷（第四、九章），何自强（第五章），陈进元（第六章），郭世铭（第七章、附录），宋柔（第八章）。吴允曾校对。全部译稿由毕源章作了文字上的统一处理。

译　者

一九八三年八月

序　言

本书是可计算性和不可计算性理论的导引，这个理论通常称为递归函数论，它涉及到解决各种问题的纯机械过程的存在性。虽然这个理论是纯数学的一个分支，但是，由于它同某些哲学问题及数学计算理论有关，所以，对于非数学工作者来说也是有一定意义的。绝对不可解问题的存在和哥德尔不完全性定理，都属于具有哲学意义的可计算性理论的结果。该理论的另一个结果是通用图林机的存在，这证实了使用数字计算机的工作者的这样一种信念：可以构造一台“全能的”数字计算机，凡是能在一台可以设想的确定的数字计算机上进行程序设计的任何问题，都能在这台全能的机器上进行程序设计（当然要受时间和存贮容量的限制）。有时，还给出这一断言的加强形式：凡是做得完全精确的任何事情，都能在一台全能数字计算机上进行程序设计。然而，这样的断言却是错误的。事实上，可计算性理论的基本结果之一（即存在非递归的递归可枚举集合），可以解释成如下断言：可以在给定的计算机上编出一个程序，以致不可能在任何一台计算机（或者是那台给定计算机的复制品，或者是另一台机器）上编出一个程序，来断定给定的数据是否是那台给定计算机的输出的一部分。另一个结果（停机问题的不可解性）可以解释成：不可能构造出一个程序来确定任意给出的程序是否会陷入无限循环。

我的目的是使可计算性理论能被不同经历和不同兴趣的

人所接受，所以我已注意（特别是在前七章中）不假定读者有任何特别的数学训练。基于同一理由，对于 McGraw-Hill 图书公司关于把这本书列入该公司的信息处理和计算机丛书的建议，我很高兴。

虽然本书中沒有多少眞正的新结果，但专家们也许会在某些课题的整理和处理上找到一些新东西。特别是以图林机的概念作为展开的核心。这样做看来是可取的，因为一方面，图林机的概念具有直观性，而且图林机同实际的数字计算机之间存在着相似之处；另一方面，把对图林机的处理同哥德尔和 Kleene 的强有力的语法方法结合起来，就可以对可计算性理论的各个方面——从 Post 的正规系统到 Kleene 的层次理论——用统一的方法进行介绍。

本书的部分材料曾用于我编的伊利诺斯大学的研究生教材，也曾用于我在伊利诺斯大学控制系统实验室和贝尔电话公司实验室的一系列讲演。本书的一部分是作者在海军研究局的资助下，在普林斯顿高级研究所工作时完成的。本书可以作为数理逻辑或可计算性理论课程的教材或补充教材。

我要感谢 Donald Kreider 先生，Hilary Putnam 教授，Hartley Rogers, Jr. 教授和 Norman Shapiro 博士，他们对本书提出过许多修正和改进的意见。

M. 戴维斯

专 用 符 号 表

这里所注明的是符号被首次引入处的页码

目　录

第四章　作用于自身的图林机

第五章　不可解的判定问题

第二部分　一般理论的应用

第六章　组合问题

第七章　刁番图方程

第八章　数理逻辑

第三部分 一般理论的进一步发展

第九章 KLEENE层次

第十章 可计算泛函

第十一章 不可判定问题的分类

引　言

1. 谈谈判定问题　当代数学的基本任务是确定关于数学对象(如整数、实数、连续函数等)的各种命题是否正确。然而，我们主要关心的是另一类数学任务，其中之一是在数学发展早期就被认为有极大的重要性，而且至今还在产生着具有重大数学意义的问题，即解决各种问题的算法或能行的计算过程的存在性问题。我们所指的是一组指令，这组指令给出一个机械过程，通过这个机械过程可以得到某类问题中任何一个问题的答案。要求这样的指令在执行过程中不带“创造性的”思维。原则上总可造出一个机器来实现这样一组指令，或编出一个程序让给定的大型数字计算机来实现这组指令。

作为一个例子，考虑以普通的十进制给出的两个正整数的求和问题。所用的算法能在任何一本算术书中找到。根据直观，我们立即就能承认这个过程纯粹是机械的，因为计算员能只依靠直接的基本指令来实现它。而且也确实有能准确地执行这些指令的加法机。

线性习蕃图方程理论给出一个更为复杂的例子。令 a,b,c 为给定的正整数、负整数或零。问是否存在整数 x,y，它们满足

$$ax+by=c。\qquad(1)$$

如果 $a=2,b=-1,c=1$，容易看出 $x=1,y=1$ 就是所希望的一组解。如果 $a=2$，$b=-6$，$c=1$，容易看出不存在作为式

(1)的解的整数 x,y。(因为这时，式(1)左边总是偶数，而右边为1，无疑是奇数。)那末有沒有这样一种算法，使我们对于给定的 a,b,c 的值，能判定是否存在满足式(1)的整数 x,y？初等数论的一个基本结果断言：式(1)有整数解当且仅当作为 a,b 二者的因子的最大正整数也是 c 的因子。不难看出，这个准则本身实际上提供了所期望的算法。

上述问题的直接推广可这样作，用任意次方的任意多个变量的多项式来代替两个变量的线性多项式。这样，问题就成了对于给定的多项式方程

$$\sum_{i_1,\cdots,i_k=0}^{n} a_{i_1,\cdots,i_k} x_1^{i_1}\cdots x_k^{i_k}=0$$

(这里所有的 $a_{i_1,\cdots,i_k}$ 都是整数)判定是否有满足它的整数 $x_1,\cdots,x_k$。然而，一直沒有找到算法来解决这个问题。而且，我们将会看到(参看第七章)，对这个问题作适当的进一步的推广，就不只是沒有找到算法，而是根本不存在算法。

有一些问题询问关于判定整个一类语句的眞假的算法的存在性，这种问题称为判定问题，它不同于只涉及单个命题的眞假的通常数学问题。一个判定问题的肯定解决是指给出了解决它的算法。否定解决是指证明了不存在解决这个问题的算法，或者说证明了这个问题是不可解的。判定问题的肯定解决经常出现于古典数学，为了确认这种解决是正确的，只要核实一下所谓的算法确实是一个算法就够了，通常认为这是不成问题的而且也一直停留在直观的意义上。然而，为了得到一个数学判定问题的不可解性，这样做就不够了。有必要对“算法”一词给出一个确切的数学定义。这件事在第一章里做。本节的其余部分专用来说明，在直观的意义上，上述关于验证一个所谓的算法确实是一个算法的问题并不象

想象中那样简单，事实上它本身就是不可解的。

考虑单变量函数 $f(x)$，它定义在正整数上(即 $x=1,2,3$ 等)，且其值也是正整数。$x^2, 2^x, \pi$ 的十进展开式中的第 x 位等等，都是这样的函数。如果有一个确定的算法使我们对任给的 x 的值都能算出对应的函数值来，我们就说这种函数 $f(x)$ 是能行可计算的。设这样的一个算法能用英语表示成为一组指令。又设所有这些指令组按所包含的字母个数进行排列：首先，(如果有的话)是由一个字母组成的，然后是两个字母的，等等。字母个数相同的指令组不止一个时，按字母表象字典中所用的办法来作排列。这样就有了第一个指令组，第二个指令组，第三个等等。对于正整数 i，有这种排列中的第 i 个指令组 E_i 与之对应，这个 E_i 告诉我们怎样计算某函数的值。以这种方法与 E_i 对应的函数称为 $f_i(x)$。

现在，令

$$g(x)=f_x(x)+1。 \qquad (2)$$

那么，$g(x)$ 是一个地地道道的函数。对于给定的数 x，它的值可这样求得，找出第 x 个指令组 E_x，然后把它作用于作为自变量的数 x 上。最后把结果增加 1。我们有：

Ⅰ. 不存在使 $g(x)=f_i(x)$ 的 i。

证 设对于某整数 i_0，有 $g(x)=f_{i_0}(x)$。则由式(2)，对 x 的一切值有

$$f_{i_0}(x)=f_x(x)+1。$$

特别地，这个等式应对 $x=i_0$ 成立，因而得出

$$f_{i_0}(i_0)=f_{i_0}(i_0)+1,$$

但这是个矛盾。

由 E_i 的挑选办法知，函数 $f_i(x)$ 应包括所有能行可计算函数。这就给出：