高等学校教材

可计算性引论

王元元

东南大学出版社

高等学校教材

可计算性引论

王元元

东南大学出版社

内 容 提 要

本书为可计算性理论的导论，是计算能行性理论的一门课程。

全书分五个部分：第一部分阐明抽象算法族可计算概念；第二部分介绍两种计算模型——Turing机及理想化的简明程序设计语言；第三部分讨论传统的递归论，并使之扩展到字函数上；第四部分介绍形式语言论的基础知识；第五部分讨论判定问题。

本书可用作工科高校计算机专业高年级学生和研究生的教材，也可供从事计算机科学的工程技术人员参考之用。

可 计 算 性 引 论

王元元

东南大学出版社出版

南京四牌楼2号

江苏省新华书店发行 江苏新华印刷厂印刷

开本787×1092毫米 开本1/32 印张10.25 字数231千字

1990年5月第1版 1990年5月第1次印刷

印数：1—2000册

ISBN 7—81023—103—X

TP·12 定价：2.10元

责任编辑 冉榴红

责任校对 陈 跃

出 版 说 明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作。由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用。

为了使工科电子类专业教材能更好地适应“三个面向”的需要，贯彻“努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材”的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986～1990年的“七五”（第三轮）教材编审出版规划。列入规划的教材、实验教材、教学参考书等近400种选题。这批教材的评选推荐和编写工作由各编委会直接组织进行。

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会（小组）评选择优产生出来的。广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力。

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位，广大教师和同学积极

提出批评建议，共同为不断提高工科电子类专业教材的质量而努力。

电子工业部教材办公室

前　言

本教材系按电子工业部的工科电子类专业教材1986～1990年编审出版规划，由计算机与自动控制教材编审委员会计算机教材编审小组征稿，推荐出版。责任编委陈火旺教授。

本教材由南京通信工程学院担任主编，郑州大学苏锦祥教授担任主审。

本课程的参考学时数为50学时，其主要内容介绍能行可计算和递归可计算概念，从理想化计算模型(计算所使用的时间和空间不受任何限制)的计算能力、固有局限性研究与可计算函数的数学特征研究这两个方面出发，来揭示计算的本质和计算的能行性。本书分五个部分：第一部分(第一章)介绍抽象算法族(抽象计算机)可计算概念；第二部分(第二至第三章)介绍两种计算模型及两种可计算概念的形式规定(这两种计算模型是简明的程序设计语言及一种理想化的计算装置——Turing机)；第三部分(第四至第六章)讨论可计算函数的数学特征，介绍传统的递归论的基本内容；第四部分(第七章)介绍形式语言论的基础知识、形式语言的生成文法和接受器；第五部分(第八章)讨论判定问题，亦即可计算概念的应用。

全书除要求读者了解“集合、关系、谓词”等离散数学基本概念外，不再要求读者预修其它专门课程。各节内容后设有一定数量的习题，大多属基本练习题。

本教材编写过程中，承陶志诚同志对书稿进行了审阅，

此外，方世昌同志也详细阅读了书稿，他们都为本书提出许多宝贵的意见，这里表示诚挚的感谢。由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正。

编　者

1987.9

目　录

第一章 抽象算法族与算法族可计算性

“计算”的概念总是和“函数”、“算法”的概念联系在一起，因为从直观上看，所谓计算就是对函数的自变量实施一组人为的操作——算法，求得因变量值的过程；当且仅当对 f 而言上述算法存在，函数 f 才是可计算的。因此，要讨论函数的可计算性，只要讨论这些算法的集合(算法族)的特性就可以了，即用它们可计算怎样的函数，不可计算怎样的函数。

如果把算法想象为一个用高级语言编写的程序，那么很容易归纳出算法族应具有的一些基本计算功能。但是，为了避免一开始便陷入具体算法的细节，使讨论更具有一般性，本章要引入抽象算法族的概念，将通常程序设计语言的计算功能强加于它，然后讨论它的计算特性。规定抽象算法族的可计算性概念，其目的是为今后讨论具体的、可形式定义的算法族——Ω语言程序族、Turing机族作好准备，为今后讨论Ω-可计算性及Turing可计算性，进而为形式定义能行可计算性铺平道路。

下面先来讨论函数与算法族的概念。

§1.1 函数与算法

1.1.1 函数

本书基本上沿用集合论对函数概念的定义方式及有关术语、符号。由于今后常常要涉及偏函数，故将函数定义重述如下：

定义1.1 设$A_1, A_2, \cdots, A_n, B$是集合，称f是$A_1, A_2, \cdots, A_n$到B的n元**函数**(functions)，当且仅当f是笛卡儿积$A_1\times A_2\times\cdots\times A_n\times B$的一个子集，并且对一切$x_1\in A_1, \cdots, x_n\in A_n, y_1, y_2\in B$，满足$\langle x_1, \cdots, x_n, y_1\rangle\in f, \langle x_1, \cdots, x_n, y_2\rangle\in f$蕴涵$y_1=y_2$。集合$Dom(f)$, $Rang(f)$分别称为f的**定义域**和**值域**，规定为

$$\mathrm{Dom}(f)=\{\langle x_1, \cdots, x_n\rangle \mid \exists y(y\in B\wedge\langle x_1, \cdots, x_n, y\rangle\in f)\}$$

$$\mathrm{Rang}(f)=\{y \mid \exists x_1\cdots\exists x_n(x_1\in A_1\wedge\cdots\wedge x_n\in A_n\wedge\langle x_1, \cdots, x_n, y\rangle\in f)\}$$

当$\mathrm{Dom}(f)=A_1\times A_2\times\cdots\times A_n$时，称$f$为**全函数**(total functions)，否则($\mathrm{Dom}(f)\subset A_1\times A_2\times\cdots\times A_n$)称$f$为**偏函数**(partial functions)。特别地，当$A_1, A_2, \cdots, A_n$及B均为自然数集合(记为N)时，称f为**数论函数**(number-theoretic functions)。

为了叙述方便，本书承袭一些常用的函数术语及记号。$y=f(x_1, \cdots, x_n)$在这里有两个意义，一个是指$\langle x_1, \cdots, x_n, y\rangle\in f$，也可看作：$y$是函数$f$在$x_1, \cdots, x_n$处的值；另一个是指由等式$y=f(x_1, \cdots, x_n)$所规定的那个函数，例如用$y=x_1+x_2$表示由$y=x_1+x_2$所规定的二元加函数

$$\{\langle x_1,x_2,y\rangle \mid x_1,x_2,y\in N\wedge y=x_1+x_2\}$$

$y=f(x_1,\cdots,x_n)$取何种意义可由它的上下文来判别。值得注意的是，当我们书写$f(x_1,\cdots,x_n)$时，并不象通常数学教材中那样已经假定它有定义。它是否有定义要据$\langle x_1,\cdots,x_n\rangle$是否在Dom($f$)中来确定。当$\langle x_1,\cdots,x_n\rangle\in$Dom($f$)时，$f(x_1,\cdots,x_n)$有定义，用$f(x_1,\cdots,x_n)\downarrow$或$f(x_1,\cdots,x_n)=a$($a$为一确定的值)表示之；当$\langle x_1,\cdots,x_n\rangle\notin$Dom($f$)时，$f(x_1,\cdots,x_n)$无定义，用$f(x_1,\cdots,x_n)\uparrow$表示。类似地，等号的意义应作如下理解：

$A=B$指“对任意$x_1,x_2\cdots,x_n$，A，B均有定义且取相同的值或同时无定义。”

$A\neq B$指“有$x_1,x_2,\cdots,x_n$，使A，B均有定义但A，B取不同的值，或者A，B不同时有定义也不同时无定义。”

$\vec{x}$常用来缩写$x_1,\cdots,x_n$或$\langle x_1,\cdots,x_n\rangle$。

将处处无定义的n元函数称为n元**空函数**，记为$\Phi^{(n)}$，即$\Phi^{(n)}$为$A_1,\cdots,A_n$到B的函数，但是Dom($\Phi^{(n)}$)$=\phi$。

我们强调，无论什么时候也不把函数f看作是给定自变量求因变量值的一个算法，即使f确可由某个计算过程来定义，例如$f=\{\langle x_1,x_2,y\rangle \mid y=\alpha(x_1,x_2)\}$，其中$\alpha$为一计算两数乘积的程序。我们宁可把$f$看作由$y=x_1\cdot x_2$定义的二元乘函数，而用另一个记号$T_f$表示计算这个函数的某个算法，这里$T_f$可以是上述程序$\alpha$。

例1.1 设f为$\{0,1,2\}$，$\{0,1,2\}$到$\{0,1,2\}$的一个函数

$$f=\{\langle 0,0,0\rangle,\langle 0,1,1\rangle,\langle 1,0,1\rangle,\langle 1,1,1\rangle\}$$

那么Dom(f)$=\{\langle 0,0\rangle,\langle 0,1\rangle,\langle 1,0\rangle,\langle 1,1\rangle\}$，Rang($f$)$=\{0,1\}$。$f$为一偏函数，$f(0,2)\uparrow$，$f(1,2)\uparrow$，$f(2,0)\uparrow$，$f(2,1)\uparrow$，$f(2,2)$

↑。读者容易作出一个程序，它象计算布尔和那样计算f，但把“2”看作“害群之马”，当有自变量取值2时，该程序一律“出错”。该程序为计算f的一个算法，它与f是两回事。

由于今后要大量地运用函数的复合，因而在此重新给出有关的定义。

定义1.2 设g_i为$A_1, \cdots, A_n$到$B_i(i=1,2,\cdots,m)$的m个n元函数，而f为$B_1, B_2, \cdots, B_m$到C的m元函数，称h为f与$g_1, \cdots, g_m$的**复合函数**(composite functions)，如果对任意$\langle x_1, \cdots, x_n\rangle \in A_1 \times \cdots \times A_n$, $z \in C$:

$\langle x_1, \cdots, x_n, z\rangle \in h$ 当且仅当 $\exists y_1 \cdots \exists y_m(\langle y_1, \cdots, y_m\rangle \in B_1 \times \cdots \times B_m \wedge \langle x_1, \cdots, x_n, y_1\rangle \in g_1 \wedge \cdots \wedge \langle x_1, \cdots, x_n, y_m\rangle \in g_m \wedge \langle y_1, \cdots, y_m, z\rangle \in f)$, h常记为$f(g_1, \cdots, g_m)$, 而$h(x_1, \cdots, x_n) = f(g_1, \cdots, g_m)(x_1, \cdots, x_n)$。

显然，当$f, g_1, \cdots, g_m$均为全函数时，h也是全函数；而当$f, g_1, \cdots, g_m$不都是全函数时，那么使得某个$g_i(x_1, \cdots, x_n)(1 \leq i \leq m)$无定义，或使$g_i(x_1, \cdots, x_n)(i=1, 2, \cdots, m)$均有定义，但$f(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n))$无定义的$x_1, \cdots, x_n$也必使$h(x_1, \cdots, x_n)$无定义，因此$h(x_1, \cdots, x_n) = f(g_1(x_1, \cdots, x_n), \cdots, g_m(x_1, \cdots, x_n))$。

例1.2 设f为例1.1中函数f, g_1, g_2均为$\{0, 1, 2\}$到$\{0, 1, 2\}$的三元函数

$$g_1 = \{\langle 0,0,0,2\rangle, \langle 1,1,1,0\rangle, \langle 2,2,2,0\rangle\},$$

$$g_2 = \{\langle 1,1,1,2\rangle, \langle 0,0,0,1\rangle, \langle 3,3,3,1\rangle\}$$

那么复合函数$h = f(g_1, g_2)$为一三元空函数，因为在$\langle x_1, x_2, x_3\rangle$不是$\langle 0,0,0\rangle, \langle 1,1,1\rangle$时，因$g_1(x_1, x_2, x_3)$或$g_2(x_1, x_2, x_3)$