STML 大学生数学图书馆 4
STUDENT MATHEMATICAL LIBRARY

可计算函数

A. Shen, N. K. Vereshchagin 著
陈光还 译

```
program selfprint;
var a:array[1..100]of string;i:integer;
begin
a[1]:='program selfprint;';
a[2]:='var a:array[1..100]of string;i:integer;';
a[3]:='begin';
a[4]:='for i:=1 to 3 do writeln(a[i]);';
      'for i:=1 to 11 do begin';
         write(chr(97),chr(91),i);';
  ]:='    write(chr(93),chr(58),chr(61));';
  ]:='    writeln(chr(39),a[i],chr(39),chr(59));';
  ]:='end;';
  0]:='for i:=4 to 11 do writeln(a[i]);';
  1]:='end.';
   i:=1 to 3 do writeln(a[i]);
   i:=1 to 11 do begin
 rite(chr(97),chr(91),i);
 rite(chr(93),chr(58),chr(61));
```

高等教育出版社
HIGHER EDUCATION PRESS

STML 大学生数学图书馆 4
STUDENT MATHEMATICAL LIBRARY

可计算函数

□ A. Shen, N. K. Vereshchagin 著
□ 陈光还 译

KE JISUAN HANSHU

高等教育出版社·北京
HIGHER EDUCATION PRESS BEIJING

International Press

Originally published in English in the title:
A. Shen and N. K. Vereshchagin
Computable Functions

图书在版编目（CIP）数据

可计算函数 /（俄罗斯）沈，（俄罗斯）韦列夏金著；陈光还译．—北京：高等教育出版社，2014.1

书名原文：Computable functions

ISBN 978-7-04-038692-9

Ⅰ．①可… Ⅱ．①沈… ②韦… ③陈… Ⅲ．①可测函数－教材 Ⅳ．① O174.1

中国版本图书馆 CIP 数据核字（2013）第 271506 号

策划编辑 李 鹏　　责任编辑 李 鹏　　封面设计 赵 阳　　版式设计 余 杨
责任校对 刘娟娟　　责任印制 朱学忠

出版发行	高等教育出版社	咨询电话	400-810-0598
社　　址	北京市西城区德外大街4号	网　　址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印　　刷	高教社（天津）印务有限公司	网上订购	http://www.landraco.com
开　　本	889 mm × 1194 mm 1/32		http://www.landraco.com.cn
印　　张	5.375	版　　次	2014 年 1 月第 1 版
字　　数	140 千字	印　　次	2014 年 1 月第 1 次印刷
购书热线	010-58581118	定　　价	35.00 元

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换

物 料 号　38692-00

《大学生数学图书馆》丛书序

改革开放以后, 国内大学逐渐与国外的大学增加交流. 无论到国外留学或邀请国外学者到中国访问的学者每年都有增长, 这对中国的科学现代化大有帮助. 但是在翻译外国文献方面的工作尚不能算多. 基本上所有中国的教科书都还是由本国教授撰写, 有些已经比较陈旧, 追不上时代了. 很多国家, 例如俄罗斯、日本等, 都大量翻译外文书本来增长本国国民的阅读内容, 对数学的研究都大有裨益. 高等教育出版社和美国国际出版社在征求海内外众多专家学者的意见的基础上, 组织了《大学生数学图书馆》丛书, 这套丛书选取海内外知名数学家编写的数学专题读物, 每本书内容精练, 涵盖了相关主题的所有重要内容.

我们希望这套翻译书能够使我们的大学生从更多的角度来看数学, 丰富他们的知识. 本丛书得到了作者本人及海外出版公司的诸多帮助, 我们谨此鸣谢.

丘成桐 (Shing-Tung Yau)

2013 年 6 月

引言

本书是在作者于国立莫斯科大学力学数学系给学生上课的讲义的基础上写成的. (作者的另一本书《集合论基础》已经出版, 是这套丛书的第 17 卷[1].)

可计算函数理论出现于 20 世纪 30 年代, 那时还没有 (现代意义上的) 计算机. 第一台计算机是 20 世纪 40 年代设计出的, 设计者之一是英国数学家 Alan Turing, 他正是可计算函数理论的创立者之一. 1936 年 Turing 描述了后来被称为 *Turing* 机的抽象计算机, 而通用 Turing 机的存在性证明导致了把程序存放到计算机存储器中这一想法的产生.

单凭这一点, 可计算性理论 (算法的一般理论) 的基本概念就值得数学家和程序设计者加以注意了. 然而, 这个理论还有着更为广泛的文化意义. 1944 年, 理论的奠基者之一、美国数学家 Emil Post 指出, 在离散数学的发展史中, 可计算性概念形式化的重要性仅次于自然数概念的形式化而排在第二位.

也许 Post 的说法现在看起来有些夸张, 他说: 近十年来, 可行性问题与不可行性问题的差异在本质上并不小于可判定性问题与不可判定性问题的差异, 而可计算复杂性理论在逻辑、数学、计算

[1]指该书英文版为美国数学会 (AMS) 的 “大学生数学图书馆” 丛书第 17 卷. 该书中译本已由高等教育出版社出版. —— 译者注

机科学上都处于核心位置, 这在哲学意义上变得越来越清晰了.

可计算复杂性已超出了本书的范围. 我们的目标限制在选择算法的一般理论中的核心概念和事例并清楚地表述出来, 避免因注重技术细节而模糊了一般概念. 阅读本书不要求特别的预备知识, 但应有一定的数学文化水平 (例如, 我们不再解释什么是实变函数).

我们期望读者能享受算法理论初步学习的这一过程. 如要进一步学习这个理论 (它是数理逻辑的核心), 读者可求助于书后的参考文献.

作者感谢他们的老师 Vladimir Andreevich Uspensky, 他的讲稿、教材和观点深深地影响了作者 (以及本书的内容).

感谢 (国立莫斯科大学力学数学系) 数理逻辑和算法理论专业的全体学生、所有听讲者和讨论班的参加者, 以及本书初稿的读者.

最后, 我们要向美国数学会和 Sergei Gelfand 致谢, 是他组织了本书的英文翻译, 我们还要感谢极其认真翻译本书的 Vladimir Dubrovsky.

作者欢迎读者对本书的错误和打字失误进行指正 (E-mail: ver@mccme.ru, shen@mccme.ru; 地址: Moscow Center for Continuous Mathematical Education, Bolshoy Vlasyevskiy Pereulok 11, Moscow, Russia, 119002).

N. K. Vereshchagin, A. Shen

目录

《大学生数学图书馆》丛书序

引言

第一章　可计算函数、可判定集与可数集 1

1. 可计算函数 1
2. 可判定集 2
3. 可数集 4
4. 可数集与可判定集 6
5. 可数性与可计算性 7

第二章　通用函数与不可判定性 11

1. 通用函数 11
2. 对角构造 13
3. 可数的不可判定集 14
4. 可数的不可分集 15
5. 单集: Post 构造 16

第三章　编号与运算 19

1. Gödel 通用函数 19

2. 可计算函数的可计算序列 22
3. Gödel 通用集 23

第四章　Gödel 编号系统的性质 27
1. 编号集 27
2. 旧函数的新编号 30
3. Gödel 编号系统的同构 33
4. 函数的可数性 35

第五章　不动点定理 39
1. 不动点与等价关系 39
2. 打印程序文本的程序 41
3. 系统的技巧: 另一个证明 44
4. 几点附注 46

第六章　m- 可约性与可数集的性质 51
1. m- 可约性 51
2. m- 完全集 53
3. m- 完全性与有效不可数性 54
4. m- 完全集的同构 57
5. 产生集 59
6. 不可分集的对 62

第七章　Oracle 计算 67
1. Oracle 机 67
2. 相对可计算性: 等价描述 69
3. 相对化 71
4. $\mathbf{0}'$- 计算 74
5. 不可比集 77
6. Friedberg-Muchnik 定理: 构造的一般方案 79
7. Friedberg-Muchnik 定理: 胜出条件 81

8. Friedberg-Muchnik 定理: 优先方法 82

第八章 算术分层 85

1. 类 Σ_n 和 Π_n 85
2. Σ_n 和 Π_n 中的通用集 88
3. 跳跃运算 89
4. 分层中集的分类 94

第九章 Turing 机 97

1. 简单的可计算模型: 需要它们做什么? 97
2. Turing 机: 定义 98
3. Turing 机: 讨论 99
4. 字问题 102
5. Turing 机的模拟 103
6. Thue 系统 106
7. 半群、生成元和关系 108

第十章 可计算函数的算术化 111

1. 有限个变量的程序 111
2. Turing 机和程序 113
3. 可计算函数是可算术化的 115
4. Tarski 定理和 Gödel 定理 118
5. Tarski 定理和 Gödel 定理的直接证明 120
6. 算术分层和量词交换数 121

第十一章 递归函数 125

1. 原始递归函数 125
2. 原始递归函数的例 126
3. 原始递归集 127
4. 递归的其他形式 129
5. Turing 机和原始递归函数 132

6. 部分递归函数 133
7. Oracle 可计算性 136
8. 生长率的估计、Ackermann 函数 138

参考文献 143

人名表 145

索引 147

第一章

可计算函数、可判定集与可数集

1. 可计算函数

设 f 是一个有着自然数自变量和自然数函数值的函数, 如果存在一个计算 f 的算法, 即存在如下的算法 A

- 如果 $f(n)$ 对于确定的自然数 n 是可定义的, 那么输入 n, 算法 A 停止并打印 $f(n)$;
- 如果 $f(n)$ 是不可定义的, 那么输入 n, 算法 A 不停止,

则 f 就称为可计算的.

这个定义有几点附注.

(1) 这个可计算性概念的定义是针对部分函数的 (它的定义域是自然数集的子集). 例如, 空函数 (在任何地方都没有定义, 即这个函数的定义域为空集) 是可计算的: 考虑算法 A, 无论输入什么它都永不停止.

(2) 这个定义可修改为 "如果 $f(n)$ 是不可定义的, 那么或者算法 A 不停止, 或者停止但不打印任何东西". 实际上, 这并没有任何实质性的改变 (代替停止而没有任何输出, 算法可以进入无限循环).

(3) 为避免误解, 我们把自然数理解为非负整数 (而不是通常那样只看作正整数). 显然, 我们的可计算函数的定义不依赖于自然数特定的表达形式 (例如, 用二进制形式表示的算法很容易改写成十

进制形式, 等等). 除了自然数外, 算法还可以输入、输出二进制串 (用字母表 $\{0,1\}$ 写成的 “字”), 自然数对, 串的有限序列, 以及一般的任意 “构造对象”. 因此, 我们能够相似地定义具有两个自然数变量、值为有理数的可计算函数.

注意, 实变量和实数值的函数的情形要难些, 函数 $f:\mathbb{R}\to\mathbb{R}$ 的可计算性需要特别的定义, 而定义可以用多种方式给出. 我们不准备详细讨论这类函数的可计算性了. 我们只要注意到, 例如, 正弦是可计算的 (在一个适当的可计算性定义的意义下), 而符号函数 $\mathrm{sign}(x)$, 对于 $x<0, x=0$ 和 $x>0$ 它分别等于 $-1, 0$ 和 1, 则是不可计算的.

同样, 对于变量为 0 和 1 的无穷序列的函数的可计算性也需要特别定义, 等等.

(4) 几十年前, 算法的概念必须作出详细的解释. 现在, 由于 “计算机文化” 的普及, 没人愿意听这种解释了. 可以把算法看作你喜欢的程序语言的程序, 只要假定存储器容量无限, 而所用的整数 (数组下标、指针, 等等) 也没有限制. 我们知道, 可计算性概念是不依赖于特定语言的: 比如在端口, C++ 的程序转为 C 程序也许是一个冗长的任务, 但理论上通常都可以做到.

然而要注意, 不要把一个非算法误认为是算法, 下面就是一个错例.

我们来 “证明” 任何有自然数变量和值的可计算函数 f 都能扩展为可计算的全 (意思是定义在整个自然数集 $\mathbb{N}$ 上的) 函数 $g:\mathbb{N}\to\mathbb{N}$. 事实上, 如果 f 由算法 A 计算, 那么下面这个算法 B 就可以计算由 f 扩展而得的全函数 g: “如果 A 在 n 处停止, 那么 B 与 A 结果相同; 如果 A 在 n 处不停止, 那么 B 返回 0 值”. (自变量错在哪里?)

2. 可判定集

自然数集 X 称为可判定的, 如果存在一个算法, 对于任意给定

的自然数 n 都能确定它是否属于集 X. 这个算法对任意 n 都应停止并给出两个答案 "yes" 或 "no" 之一 (或者 1/0, TRUE/FALSE, 等等).

换句话说, X 是可判定的, 如果它的特征函数

$$\chi(n) = (\text{如果 } n \in X \text{ 则为 } 1, \text{ 否则为 } 0)$$

是可计算的.

显然, 可判定集的交、并、差集也是可判定的. 任何有限集是可判定的.

自然数对集、有理数集等集合的可判定性可以类似地定义.

问题 1. 证明小于 e (自然对数的底数) 的所有有理数组成的集合是可判定的.

问题 2. 证明非空的自然数集是可判定的当且仅当它是自变量和函数值均为自然数的全非减可计算函数的值域.

下面这点很微妙: 集合的可判定性可以非构造性地证明, 而不需要明确地描述判定算法. 一个传统的例子是: 所有 n 的集, 数 π 中至少含有 n 个连续的 9. 这个集是可判定的, 因为它或者包含所有自然数 (因而是可判定的), 或者由小于某数的所有自然数组成 (所有有限集均可判定). 这样, 我们就证明了在任何情形下这个集都是可判定的. 然而, 我们并没有精确地证明, 对于给定的 n, 算法能够确定数 π 中是否至少包含着 n 个连续的 9.

问题 3. 在上述论证中, 我们用到了数 π 的任何性质吗? 如果把 "至少 n 个 9" 换成 "恰有 n 个 9 (旁边都不是 9)" 会有什么变化吗?

不可判定集是否存在? 回答显然是肯定的. 因为有可数多个算法 (因此就有可数多个 $\mathbb{N}$ 的可判定子集), 然而, $\mathbb{N}$ 的所有子集的集合却是不可数的. 后面我们将给出具体的例子.

3. 可数集

一个自然数集称为可数的, 如果它能被一个确定的算法计数, 即存在一个算法, 打印 (按任意顺序和任意延时) 而且只打印这个集的所有元素.

这样的算法是没有输入的; 打印完几个数字后可能会突然进行冗长的计算, 经过一定时间间隔再打印下一个数字, 或者不再打印任何东西了 (这意味着这个集是有限的).

可数集有许多等价定义. 以下是其中的几个:

(1) 一个集是可数的, 如果它是一个可计算函数的定义域.

(2) 一个集是可数的, 如果它是一个可计算函数的值域.

(3) 一个集是可数的, 如果它的 (有时会说成) 半特征函数是可计算的. 半特征函数定义如下: 对 X 的元素它等于 0, 在 X 之外无定义.

我们来证明上述定义等价; 证明中, (0) 表示最初的那个定义.

(0) $\Rightarrow$ (1), (3) 假设 X 由算法 A 计数, 那么集 X 的半特征函数是可计算的. 事实上它可由如下的算法计算:

> 输入数 n, 一步步执行算法 A 直到数 n 被打印, 立即输出 0, 且算法停止.

(1) $\Rightarrow$ (0) 令 X 为 (可计算) 函数 f 的定义域, f 由算法 B 计算, 则 X 可由下述的算法 A 计数:

> 顺序输入逐渐增大的数 $0, 1, 2, \ldots$, 一步步并行执行算法 B (首先, 输入 0 和 1, B 执行一步; 每输入 $0, 1, 2$ 就执行两步; 每输入 $0, 1, 2, 3$ 就执行三步; 等等). 一旦检查到算法 B 终止, 立即打印出自变量.

定义 (3) 显然蕴涵定义 (1), 它们与初始定义等价即已成立.

我们来证明 (2) $\Rightarrow$ (1), 即要找到对可计算函数 f 的值域计数的算法. 为此, 我们只需把上述的算法 A 改为打印由 B 返回的结果, 而不是 B 终止时的自变量.

最后再来证明 (1) ⇒ (2), 即要说明任何一个可数集 X 都是一个可计算函数的值域. 我们已经知道 X 是一个可计算函数的定义域, 如果这个函数是由算法 A 计算的, 那么 X 就是函数 b 的值域, 当输入 x 时 A 终止, 函数 b 取值 x, 此外 b 无定义. 我们写出 b 的定义如下:

$$b(x)=\begin{cases}x, & \text{当 } A \text{ 终止于 } x \text{ 时},\\ \text{无定义}, & \text{其他情形}.\end{cases}$$

计算这个函数的算法和 A 完全一样, 只是它不由算法 A 产生结果, 而是复制了输入的数据.

这里还有一个可数集的等价定义: 自然数的集 X 是可数的, 如果 X 或者为空集, 或者是一个全可计算函数的值域 (换句话说, 它的元素可以排成一个可计算序列).

事实上, 假设非空可数集 X 可由算法 A 计数, 令 x_0 是 X 的任意元素, 考虑下面的全函数 a: 如果第 n 步返回数 t, 则 $a(n)=t$; 如果什么也不返回, 则 $a(n)=x_0$ (假定在任何给定的步骤只出现一个数, 否则计算将被分成更小的步骤).

应当指出这个推理是非构造的. 就是说, 给定了算法 A 但我们并不需要知道被计数的集是空集还是非空集.

定理 1. 可数集的交和并是可数的.

证明. 如果 X 和 Y 由算法 A 和 B 计数, 那么它们的并可由并行执行 A 和 B 的算法来计数, 打印出由 A 和 B 打印出的东西. 交的情形要难一些: 由 A 和 B 生成的结果要存储起来再进行比较; 把公共的结果打印出来. ■

问题 4. 应用其他的可数性等价定义之一来证明定理 1.

我们将看到, 可数集的补集可能不可数.

问题 5. 有时会考虑 "非判定性算法" (这种矛盾修辞是常用的), 这类算法包含着类似这样的指令

$$\mathtt{n} := \text{任意自然数}$$

(指令 "n := 0 或 1" 已足够了, 因为任何数都可以一位一位地构成). 非判定性算法随着它对 "任意数" 的选择, 可能对同样的输入有不同的计算途径. 试证: 一个可数集可以等价地定义为出现在一个非判定算法 (有固定的输入) 输出中的数集.

问题 6. 证明如果集 $A \subset \mathbb{N}$ 和 $B \subset \mathbb{N}$ 是可数的, 那么它们的笛卡儿积 $A \times B \subset \mathbb{N} \times \mathbb{N}$ 也是可数的.

4. 可数集与可判定集

定理 2. 任何自然数的可判定集都是可数的. 如果集 A 和它的补集 $\mathbb{N}\backslash A$ 是可数的, 那么 A 是可判定的.

证明. 如果有一个算法可以检验一个数是否属于集 A, 那么集 A 和它的补集就是可数的. 我们可对数 $0, 1, 2, \ldots$ 中的每一个进行检验, 属于 A 就打印这个数 (或者打印不属于 A 的数).

反之, 假设我们有一个算法对 A 计数, 另有一个算法对 A 的补集计数. 给定一个数 n, 现在来确定它是否属于 A, 为此, 只要运行这两个算法, 直到其中之一打印出 n (我们知道其中之一最后一定会打印 n). 检查是哪个算法打印这个数, 就可以确定 n 属于 A 或者不属于 A. ■

这个事实被称为 *Post* 定理.

它说明可判定集是有可数补集的可数集. 此外, 可数集还能以可判定性语言来定义.

定理 3. 自然数的集 P 是可数的, 当且仅当 P 是可判定的自然数对的集合 Q 的射影. (关于数对集的射影, 指的是数对中第一分量的集合: 即 $x \in P \Leftrightarrow \exists y(\langle x, y\rangle \in Q)$.)

证明. 任何可数的数对集的射影是可数的 (对数对计数并抽出第一个元素), 这样, 可判定集的射影更是可数的了. ■

反之, 由算法 A 计数的可数集 P 是由所有数对 $\langle x, n\rangle$ 组成的