可计算函数讲义

〔苏联〕B. A. 烏斯宾斯基著

上海科学技术出版社

可計算函数讲义

〔苏联〕B. A. 烏斯宾斯基 著
毕源章 譯 管紀文 校

上海科学技术出版社

內 容 提 要

本书系原作者根据他在莫斯科大学数学力学系讲授"递归函数"課的讲稿加以整理而成．用集合論观点闡述了可計算函数的理論基础及其某些应用．全书計分十四节．§§ 1～3 是閱讀本书的预备知識，§§ 4～9 是部分递归函数和递归可枚举集合的理論基础，§ 10 反映了可計算函数論、可枚举集合論与描述集合論之間的深刻的类似性，§ 11 叙述了編号概念和运算，§§ 11～14 是可計算函数理論在实数理論、定义理論、計算机理論上的一些应用．

本书可作为高等院校数学专业中"递归函数"課程的教学参考书，也可供有关工作人員参考．

ЛЕКЦИИ О ВЫЧИСЛИМЫХ ФУНКЦИЯХ

В. А. Успенский

Физматгиз · 1960

可 計 算 函 数 讲 义

毕源章 譯 管紀文 校

上海科学技术出版社出版（上海瑞金二路 450 号）

上海市书刊出版业营业許可証出 093 号

商务印书館上海厂印刷 新华书店上海发行所发行

开本 850×1156 1/32 印張 13 4/32 排版字数 324,000

1966 年 3 月第 1 版 1966 年 3 月第 1 次印刷

印数 1—1,800

統一书号 13119·686 定价（科六）2.00 元

記 号 表

I. 俄文字母記号

II. 謂詞、函数与算子記号

目 录

§1. 引　言

可計算函数是这样一种函数，这种函数是可以找到計算它的值的算法的. 当然，这句話有很多不淸楚的地方. 在整个的这一节里，将对这些不淸楚的地方加以解釋. 暂时，我們只指出：这句話把可計算函数的概念归結为两个基本概念，即：函数的概念和算法的概念.

想必讀者已然知曉函数的概念. 我們回想起，在談到函数时，通常是指某一規則，按照这一規則可以使某些对象（所謂**变元的值**）对应于另一些对象（所謂**函数的值**）. 这里，对于对应規則的特性沒有附加任何限制. 这种規則可以是任意的，其中，它也可以是这样的規則，按照这一規則，我們根本不可能由变元的值实际地求出相应的函数值. 例如，我們引进这样的函数 f：

$$f(n)=\begin{cases}1, & \text{如果在数 } \pi \text{ 的十进制展开式中有相继的 } n \text{ 个零;}\\ 0, & \text{否則.}\end{cases}$$

我們认为这样刻划出来的函数 f 是給定的，虽然（据作者所了解的）現代知識水平还沒有給出对任何 n 計算出这个函数的值的方法. 可能这种方法是存在的，只不过我們还沒有找到它；也可能根本就不存在这样的方法（我們預先指出，对于可計算函数来說，这类方法总是存在的，这就从一般的函数中分出了可計算的函数）.

无疑地，讀者已多次接触过算法的概念（虽然可能还沒有注意到这一点）. 数学中出現的算法的例子幷不少于函数的例子. 但是，“算法”这个詞本身，看来不象“函数”这个詞那样常見和“术語化”（似乎，“算法”一詞，除了在算法論中，也許就只有在“欧几里得

算法”一語中才經常出現)．“算法”这一术語意味着什么？或者，人們通常所說的数学中的“算法”究竟是什么意思？“一个計算过程从可变的初始資料导出所求的結果，在数学中通常把确定这种計算过程的准确指令理解为‘算法’”(A. A. 馬尔科夫[1954]，中譯本“算法論”，上册第i頁)．当然，这还不是这个詞在通常数学意义下的定义，它只不过是一种描述，对算法的一般概念未必能給出精确的数学定义．大概不能把这一概念归結为更簡单的概念了❶．看来，还是宜于把它当作一个不可定义的概念(这种情况决定了算法概念在一定程度上的模糊性和直观性)．算法概念是直接从經驗中抽象出来的，因此只能通过例子来理解它．求两个正整数的最大公因子的著名的欧几里得算法，就是算法的經典的例子．这个例子絕不是最簡单的，它之所以能如此著名，只是因为这里明显地使用了“算法”这个詞．例如，用竪綫段做数的加法的算法，以及在小学里讲的算术运算的其他算法，則簡单得多(我們指出，在中世紀的欧洲，正是把現代的小学算术，即十进制及其演算方法，叫做算法)．“会加”或“会乘”就意味着知道某些算法．这些簡单的例子就已表明，在数学里(无論在理論上，或在应用上)，每一步都会遇到算法，而且学会解决“一般形式”的問題，实质上总是意味着掌握了某种算法．我們介紹讀者参看 Б. A. 特拉赫坚勃罗特的通俗著作[1957](新版本[1960])，那里有一些更有趣的和离奇的算法的例子．

对于每个算法，都存在着某个可能的初始資料(对象)的总体——把这个算法应用到这些对象上是有意义的．例如，对于欧几里得算法来說，所有正整数偶的总体就是这样的总体．如果应用算法到任一对象的过程能以得出結果而告終，就称这算法**可应用于**这个对象．算法完全未必可应用于相应(可能的初始資料的)

❶ 讀者对这种情况不必感到过分不安，因为，实质上，一切初始的数学概念(如“集合”，“对应”，“自然数”等等)都有类似的情形。

总体中的任何对象．并且，一般説来，当我們把算法应用到所述总体中任一对象上时，事先并不知道能否得到結果，也就是說，事先我們并不知道这算法是否可应用于所取的对象(这里可以指出，甚至能够构造出这样的算法，对于它来説，根本不存在按它的任意的可能初始資料而辨别它是否可应用于此对象的算法)．这样，对于每一个算法，在这个算法的一切可能的原始資料的集合中又分出了算法的**可应用域**．

每个算法都給出一个函数，这函数定义在算法的可应用域上，且使这域中的每个元素都对应于应用該算法于此元素上的結果．任何函数，如果它可以在上述意义下由某个算法所給出，則称这函数是**可計算的**．这样，我們就精确化了本节头一句話中关于把可計算函数概念归結为算法概念的説法．为了最終完成可計算函数这一概念的精确化，显然只要把算法概念予以精确化就是了．

算法概念已經有了一系列的精确化❶．其中每种精确化实质上都描述了某个具体的算法类，这算法类有着下述意义下的充分的完全性：任意一般算法都可以用此具体类中导致同样結果的算法来代替(假定被代替的算法的初始資料和結果都包括在为該精确化所容許的初始資料和結果之中)．当然，这种“完全性”并不能用数学方法証明，它对于每种精确化来説乃是一种自然科学的事实或假設．最后，一切已知的关于算法概念的精确化，在某种合理的意义下又都是互相等价的(这种等价性已被确切地定义和証明了)．

一切已有的关于“算法”概念的精确化，都是从下述的一般概念出发的(或者是从很容易归結为这些概念的事实出发的)．

算法过程(即应用算法于某个对象的过程)分成若干各別的、

❶ 关于算法与可計算函数概念的一些基本的精确化的簡要綜述，可参看 A. H. 柯尔莫哥罗夫和 B. A. 烏斯宾斯基的論文[1958]中的 § 1.

相当初等的步驟．每一步都是用这过程的一个状态去代替另一个状态(而初始資料則作为初始状态)．从某个状态到直接的下一步状态的过渡，是根据所謂的直接加工規則进行的，这些規則应当是相当初等的．某些状态被判明是結尾状态(根据相当初等的結尾規則)，而最終的結果即从中选出(也是根据相当初等的規則)．当把算法应用于某个对象时，算法过程可能有三种情形：1° 每一状态都为下一状态所替代，因而过程永不終止；2° 到某一步遇到了一个这样的状态，对于这个状态，无論直接加工規則或是結尾規則都不适用，因而过程无結果終止；3° 到某一步遇到了結尾状态，因而过程有結果地終止，并相应地得到了最后的結果．因之，算法只可应用于那些使算法过程按第三种情形进行的对象．

并非数学中的任何对象都可作为算法的初始資料、結果或中間資料．例如，把某个无穷集合当作初始資料而談論关于可应用于这样的集合的算法是沒有意义的．能够参与算法过程的，只是那些所謂**构造性对象**——自然数和有理数、以自然数或有理数为系数的多項式、元素为自然数或有理数的矩陣、某个字母表[1]中的字等等．概略地說，构造性对象就是这样的对象，它能够被完整地构造出来而整个地呈現在我們面前．在我們作这种了解的时候，广泛地应用着所謂**潜在的可实現性的抽象**．根据 A. A. 馬尔科夫的說法，这种抽象是“从我們构造可能性的实在界限中抽象出来的，这种构造的可能性是由我們生活在空間和时間中受到的限制所提供的”(A. A. 馬尔科夫[1954]，中譯本“算法論”，上册第11頁)．这一抽象使我們能够討論随便多么大的自然数，随便多么长的字，一般地，随便多么大的和随意多么复杂的(但是有限的!)构造性对

[1] 任何有限符号組都称为**字母表**，出現在字母表中的符号称为該字母表的**字母**，由某字母表的字母一个挨一个排列起来的有限序列称为該字母表中的**字**；更詳細的討論可參看 A. A. 馬尔科夫[1954]的第一章．

象．对构造性对象的概念，未必能給出形式的定义．与算法概念相仿，看来宜于把这一概念也当作原始概念．

既然只有构造性对象才能够作为算法的可能的初始資料和結果，所以也只有构造性对象才能作为可計算函数的变元和值．现在，我們可把第3頁上所表述的可計算函数的定义改述如下：設已有两个构造性对象的集合：X 和 Y．其一切变元值皆属于 X 而其一切函数值皆属于 Y 的函数 f 称作是**可計算的**，如果存在着一个算法，它的可应用域与函数 f 的定义域相同，且对于此域中的任何 x，应用該算法于 x 的結果与 $f(x)$ 相等．应当着重指出，我們完全不要求可計算函数是(在整个 X 上)处处有定义的．此外，我們也不要求自己会区别变元的哪些值属于函数的定义域，而哪些值則不属于．我們只要求存在着計算这函数的值的算法，当此算法应用于变元之属于函数定义域之值时，經过若干步(一般説来，我們預先并不知道这个步数，而且对步数也未加任何限制)之后，它将計算出函数的值．而如果函数在变元的某个值上无定义，則对于被应用到变元的这个值上的計算函数值的算法，除了要求它不导致明知在这种情况下是錯誤的結果而外，不再提出任何要求．可能在这种情形下应用算法的过程将停止，但并不給出結果．这时我們就知道函数在变元的这个值上无定义．但，也可能算法将无限地工作下去，因而我們就无法知道，是算法工作的步数还不足以計算出所求之函数值呢，还是函数在变元的这个值上无定义？

在可計算函数概念的基础上，也定义了一系列重要的概念，首先是可判定集合和可枚举集合的概念．

一集合(被包含在某个构造性对象的总体中的)称为**可判定的**(关于这个构造性对象的总体)，如果存在一个算法，它能辨别出此总体中的任意元素是否属于这个集合．换言之，一集合是可判定的充要条件是：它的特征函数(卽定义在此总体上的函数，这个函

数对属于該集合的对象取值为1，对其他对象則取值为0）是可計算的.

一集合称为**可枚举的**，如果它是某个定义在整个自然数列上的可計算函数的值的集合（因此，可枚举集合的元素必然是构造性对象）．既然可枚举集合是定义在自然数列上的可計算函数 f 的值的集合，所以函数 f 就能够依次地、一个一个地（可能有重复）得到此集合的元素，或者說生出此集合的元素．这样，每个可枚举集合在下述意义下都是能行可生成的，卽，对于这个集合来說，存在着一个生出它的元素的能行（卽遵从一些确切的且为大家所了解的規則的）过程．当然，通过依次地計算定义在自然数列上的可計算函数之値来生出元素的过程，是能行生成过程的极其特殊的形式．还可以想出更为一般和更为复杂的过程（例如，从給定的前提中做出推論的形式推理就是能行生成过程）．因此，有可能以为能行可生成集合的概念要比可枚举集合的概念更为广泛些．然而，当把“能行生成过程”的概念予以合理的精确化之后就可以看出，任何非空能行可生成集合都是可枚举的．但空集显然是能行可生成的（可以指出一个不产生任何元素的过程），所以，为了不破坏能行可生成与可枚举这两个概念的同义性，我們把空集也算作可枚举集合．

所有使 $f(a)=b$ 的序偶 $\langle a, b\rangle$ 的总体，称为函数 $y=f(x)$ 的**图形**．根据直观的考察，可以提出如下的断言（这一断言的模拟形式将在§6定理3中予以証明）：一函数为可計算函数的充要条件是它的图形为可枚举集合．由此，我們就可以通过可枚举集合的概念来定义可計算函数的概念：把具有可枚举图形的函数叫做可計算函数．既然可枚举集合的概念可以独立于算法概念而通过精确化的能行生成过程的概念来定义，于是我們就得到了一种新的使可計算函数概念精确化的可能性（这种可能性不依賴于算法概念的任何精确化）．事实上，也正是沿着这个途徑，曾經获得了可計

算函数概念的几种最早的精确化❶.

可計算函数概念的精确化,还有第三条可能的途徑,这也就是我們在本"讲义"中所选擇的途徑.

我們指出，首先可以只限于考察其变元和值都是自然数的函数. 因为对于任何一个自然地产生的构造性对象的总体(例如以自然数为元素的一切矩陣的全体，或者是給定的字母表中的一切字的全体,等等),都可指出这总体的一个一一編号的算法,也就是这样的算法,它使每个自然数都对应这总体中的某个对象,而且使不同的数对应不同的对象(由这样的算法的存在又可以推出逆算法的存在.逆算法对总体中的对象給出它的号数,即对应于它的自然数). 这时，把这些对象变成那些对象的可計算函数，就对应于把这些对象的号数变成那些对象的号数的可計算函数，而且从后一函数很容易找到前一函数(根据同样的道理,为使可枚举集合与可判定集合的概念精确化,只要考察自然数的集合就够了). 在本书中,我們只限于考察其变元和值都是自然数的函数.

下面就来說明一下这条使可計算函数概念精确化的第三条途徑. 我們将(不依賴于算法和能行生成过程的概念)写出某些函数类和某些集合类,然后說明它們就是可計算函数、可枚举集合和可判定集合概念的精确化. 亦即,我們将定义出所謂部分遞归函数、遞归可枚举集合和一般遞归集合，并且把它們分別与以自然数为变元和值的可計算函数、自然数的可枚举集合和自然数的可判定

❶ 最早的几种精确化为下列作者所提出：对于以自然数为变元和值的处处有定义的函数的情形——A. 邱吉[1936]. 更早一些时候，K. 哥德尔[1934]定义了**一般遞归函数**,他把这种函数定义作这样的处处有定义的函数,即它的图形的元素可被某个特殊形式的过程所生成. A. 邱吉在他的文章[1936]中考察了另一种形式的生成过程，并断定(同时指出， S. C. 克林也得到了这个結果)由这一过程所定义的处处有定义的函数(他称之为 **λ 可定义**的函数)与一般遞归函数一致，而且也建議把以自然数为变元和值的处处有定义的可計算函数的概念与一般遞归(或 λ 可定义)函数的概念等同起来(后来"λ 可定义的"这一术語不仅被应用于处处有定义的函数).

集合等同起来. 一方面，这种等同是直观上显然的(即，直观上显然：任何部分递归函数都是可計算的，任何递归可枚举集合都是可枚举的，任何一般递归集合都是可判定的)，另一方面，它是为經驗所証实的自然科学假設.

这样，我們就可以在双重意义下来使用"可計算函数"这一术語:第一,用来表示在这种或那种精确化的基础上所产生的某个精确的概念;第二,用来表示在直观观念的基础上所产生的多少有些含糊的概念. 如果我們所指的是第二种意义下的可計算函数，而我們又想把这一点强調出来，这时我們将說是**在直观意义下的可計算函数**,或者簡短地說成**直观可計算函数**.

§2. 集合与函数理論方面的一些預备知識

本节将引进一些关于集合和函数的补充知識. 虽然这些知識是很初等的,但以后常常要用到. 如果不熟练地掌握本节的材料,则在讀下面章节时将会发生困难.

1. 集　　合

在研究任一集合时,常常不仅只与它的单个元素打交道,而且也要和它的元素的有序偶打交道. 同时,序偶的第一項也容許与第二項相同. 比方,

$$\langle\sqrt{2},\ \sqrt{2}\rangle,\quad \langle 10,\ \frac{11}{17}\rangle,\quad \langle\frac{11}{17},\ 10\rangle$$

是由实数組成的不同的序偶的例子.

与此类似地,在研究中也常常要考虑一已知集合之元素的有序三元組,有序四元組,……,有序 n 元組,……(和前面一样,也容許某些項相同). 比方,

$$\langle 8,8,8,8,8\rangle,\quad \langle 1,2,3,4,5\rangle,$$
$$\langle 6,6,10,1,10\rangle,\quad \langle 10,1,10,6,6\rangle$$

是由自然数序列的元素所組成的不同的有序五元組的例子.

有序偶,有序三元組,一般說来,有序 n 元組(对于任何自然数 n)的概念,未必能用更簡单的概念来定义. 推广这些概念,我們就得到已知集合之元素的有序組的概念. 在不同的数学領域中,任一集合之元素的有序組的名称是不同的:在組合分析中称为重复排列,在代数中称为矢量,在概率論中称为有退还抽样下的样本,而在抽象集合論中則称为串. 我們将使用最后这个术語. 由

集合M的元素所构成的串，簡称为**M上的串**．由元素$x_1, x_2, \cdots, x_s$，并以此为序所构成的M上的串記作

$$\langle x_1, x_2, \cdots, x_s\rangle,$$

并且說，x_i是該串的第i个**坐标**，或**分量**．

串$\langle x_1, \cdots, x_s\rangle$的坐标个数$s$称为它的**长度**．与长度为2、3、4等的串一样，我們也談长度为1的串$\langle x\rangle$和**空串**$\langle\ \ \rangle$．空串将記作$\wedge$❶，并认为其长度是0．长度为2的串，即有序偶，常簡称为序偶，长度为3的串，即有序三元組，簡称为三元組，等等．

由集合M可构成集合M的幂：长度为2的(M上的)串的集合M^2，长度为3的串的集合M^3，一般地說，长度为s的串的集合M^s．我們也談集合M的一次幂集合M^1，这时我們是指长度为1的串的集合M^1而言；我們也談集合M的零次幂集合M^0，这是指由一个元素(即：空串$\wedge$)所构成的集合，即是：$M^0=\{\wedge\}$．

M上的一切可能的串的集合記作M^∞．按照定义，

$$M^\infty=\bigcup_{s\in N}M^s=M^0\cup M^1\cup M^2\cup M^3\cup\cdots.$$

这里N表示集合0, 1, 2, 3, ⋯．我們在全书中都始終使用这一記法．今后，集合$N=\{0, 1, 2, 3, \cdots\}$将起着基本的作用．按照可計算函数論中通常的习惯，我們称集合N的元素为**自然数**(而集合N本身則称为**自然数序列**)，因此这里把0也算作自然数．今后我們往往省写“自然”这两个字(在所有不致引起混淆的地方，凡是說到数就是指集合N的元素)．

設有两个集合M_1和M_2．所謂集合M_1和M_2的**外积**$[M_1, M_2]$，是指由一切可能的序偶$\langle x, y\rangle$所构成的集合，其中$x\in M_1$，$y\in M_2$．类似地，可以定义三个、四个或者更多个因子的外积．例如，$[M_1, M_2, M_3]$是由所有可能的三元組$\langle x, y, z\rangle$构成的集合，

---

❶ 这样，我們采用了通常用来表示空集的符号来表示空串，在本书中用这同一个符号旣表示空串又表示空集．