原始递归算术

沈百英 著

华东化工学院出版社

原始递归算术

沈百英 著

华东化工学院出版社

责任编辑　郑汶玉

责任校对　黄黎峥

原始递归算术

Yuanshi Digui Suanshu

沈百英　著

华东化工学院出版社出版

（上海市梅陇路130号）

新华书店上海发行所发行

常熟第四印刷厂印刷

开本787×1092 1/16　印张 28　字数 681千字

1990年10月第 1 版　1990年10月第 1 次印刷

印数1-1,000 册

ISBN 7-5628-0081-2 /O·11　定价：5.55元

内 容 提 要

原始递归算术是一种无逻辑的方程演算，更是一种严格的有穷主义理论，它所研究的对象和所使用的研究方法都具有可构造性。

原始递归算术也是初等数论中可构造的部分。由于它的概念简单，论证严密，且推理的根据都详细列出，因此它的研究使初等数论的研究出现了一个可构造性的新面貌。

本书系统地介绍了递归算术的基本概念。内容包括原始递归算术的各种公理系统、初等算术算子、原始递归式的加强、配对函数的一般构成、数论函数的逆函数、各种高等规则、可构造发展的初等算术、算术的不完备性定理等。全书以递归思想贯穿始终。

本书总结并发展了国内外学者的，特别是作者近几年来的研究成果，其中很多重要结果系第一次公开发表。

本书可作为高等院校计算机科学系与数学系高年级学生及研究生的教材或教学参考书。

序　言

原始递归算术首先由挪威数学家Skolem, Th.(1923年)提出，并由 Hilbert, D. 和Bernays, P(1934年)加以发展。1941年，Curry, H. B. 和Goodstein, R. L. 独立地提出了不用逻辑形式建立原始递归算术，明确地把原始递归算术描述为一个无逻辑的方程演算。他们各自提出了原始递归算术的形式系统。所不同的是，前者使用通常的归纳规则，而后者使用原始递归式定义函数的唯一性规则。1954年，Goodstein[32]第一次明确地建立了第一类原始递归算术 R 与第二类原始递归算术 R_1（以及 R_1 的一个变种）。Goodstein 的工作系统地总结在文献[33]中。1957年，Church, A. 建立了一个二元原始递归算术，第一次使用配对函数考虑参数的化归，特别对存在性公理作了化归。但 Church 的这一系统使用了命题逻辑。1962年，Rose, H. E. 建立了一个三元原始递归算术，全部不用逻辑。

在国内，莫绍揆教授是第一个研究递归算术的。作者在莫绍揆先生的指导下从1962年开始对递归算术进行研究。首先他们对Goodstein在1954年建立的系统作了本质上的改进，建立了第一类的原始递归算术系统 B_1V_2 以及几个变种（见文献[19]）和几个第二类的原始递归算术系统（见文献[20]）。1963年，又对初等系统的完备性与判定性问题作了研究[17,18]。同时作者对基础系统作了化归（见文献[1]）（后来得知，1962年Rose, H. E. 也独立地得到了一个第一类原始递归算术，即上面提到的三元原始递归算术，此系统对基础系统也相应地作简化。1963年 Goodstein[34]也独立地对初等系统的判定问题作了一些研究），并在1963年西安召开的第一次全国数理逻辑专业学术会议上作了介绍。1962—1965的三年研究生阶段，作者又对原始递归算术作了多方面的发展。详见文献[3]。但由于历史原因，三年的成果都未被发表。从1980年开始，按照 Gladstone, M. D. 对原始递归式的进一步化归的结果[28,29]，作者对原始递归算术的公理系统以及其他的问题重新又进行了研究，详见文献[4—12]中。

在20多年的生活波动中，作者一直想把其对原始递归算术所作的研究工作做一个总结，但这一愿望一直未能实现。两年前，作者对原始递归算术的公理系统的研究又有了新的突破，从根本上改进了第一类，第二类原始递归算术[7,8]。因此，本书是作者20多年研究工作的总结，不仅有作者在研究生阶段所得到的成果，还有很多是在这次总结工作中重新得到的。有些内容虽在近几年中已有发表，但在本书中又得到了根本的改进。

目前在国内，介绍递归算术方面的专著还未见出版，而在国外也仅有Goodstein, R. L. 写的《递归数论》一书[33]（1957年）。因此本书难免有缺点，甚至错误，为此敬请读者不吝指正。

本书的出版得到华东化工学院领导、华东化工学院出版社领导及编辑同志的关心与支持，作者深表感谢。

沈百英

1988年10月于上海

目　录

第0章 绪 论

§0.1 递归算术的本质

1902年罗素(B. Russell)发现了集合论中的悖论，使得人们真正相信集合论本身含有矛盾。由于整个数学都是以集合论为基础，于是人们一方面使用公理的方法来重整集合论，另一方面要求对每个数学理论必须作相容性(或不矛盾性)的证明。除了对一些理论作相对相容性的证明外，还必须对另一些理论作直接的相容性证明。为此希尔伯特(D. Hilbert)提出一个规划(通常称为希尔伯特规划)，用作直接证明一个数学理论的相容性。但问题是使用什么工具去证明一个数学理论的相容性？用什么来保证所使用的工具本身是没有矛盾的呢？为了保证所使用的证明工具本身是相容的，至少要求使用最可靠的、最有效的并且尽可能简单的工具。于是希尔伯特提出使用有穷主义方法。但什么方法才是有穷主义方法？“这个不明确的概念的最可能的解释是，它们大约相当于原始递归算术的一个弱的扩张”①。由此可见，原始递归算术是一个严格有穷主义的理论(比希尔伯特提出的有穷主义更严格)。事实上，递归算术除要求所处理的对象具有可构造性外，还要求所用的证明方法也是可构造性的方法。因此在递归算术中，可构造性这个要求是贯彻始终的，而递归算术成为一门严格可构造算术或严格可构造理论。

最简单的可构造性对象无疑是自然数，递归算术的处理对象就是自然数。可以说，递归算术是关于自然数的理论。在逻辑中，全称量词、存在量词是一个非可构造性的对象，因此递归算术必然不使用这些量词。由于不使用全称量词，就必须要大量使用自由变元。因为对全称公式

$$\forall y P(y)$$

就简单地去掉全称量词，而引入自由变元 x，从而转为处理

$$P(x)。$$

对存在公式

$$\exists x P(x)$$

就要把使得 $P(x)$ 成立的 x 具体找出来，也即用一个函数常项表达出来。于是在递归算术中要大量地使用函数。对一个函数的定义，必须也是可构造性的。显式定义无疑是允许的，但只靠显式定义是不够的。一个强有力的可构造的定义方式是递归定义或归纳定义的方法，因此递归算术需要大量使用归纳定义。现在对每个(一阶)量词公式的处理可换以一个无量词公式的处理。例如，公式

$$\forall x \exists y P(x, y)$$

① 王浩，数理逻辑通俗讲话，科学出版社，1981年，P.151.

换以公式

$$P(x,\ f(x))。$$

除此以外，递归算术对任意谓词P也加以简化处理，只把“相等性”作为基本，而其他的关于自然数的关系都借助函数和等词(即“=”)化归为相等性的关系(即以等式的形式出现)。更进一步，递归算术对逻辑中命题联结词的处理也归结为关于相应的特征函数的处理。

于是递归算术中所讨论的对象是：个体为自然数，以及由自然数经函数构成的项。关系为相等性。因此所处理的是等式，函数的引入是使用显式定义和归纳定义。

递归算术中所使用的证明方法也是可构造性的。最简单的可构造性证明方法是代入和替换。由于对象是等式，因此代入是对一个(已证)等式中的变元处处代以某个项(自然数或函数的填式)。替换是在一个(已证)等式中把某项(的一处出现)替换以与此项相等的另一项。正如除使用显式引入函数以外还要使用更重要的归纳方法引入函数一样，除使用代入和替换这些可构造性证明方法外，还要使用一种更重要的证明方法，即递归定义函数的唯一性证明方法。详细地说：如果两个函数满足相同的递归定义，则它们是相等的，即用相同的递归定义方法所确定的函数是唯一的。这与通常的数学归纳论证方法相类似，但由于我们处理的对象是等式，因此使用递归定义函数的唯一性方法更为方便。另外，使用这种方法还可与引入函数的递归定义方法相对应起来，统一为使用递归定义函数的存在性和唯一性两个方面。

综上所述，递归算术实质上是一种无逻辑的方程演算。它是一种严格的有穷主义理论，所研究的对象和所使用的研究方法都是可构造性的，而递归的思想是递归算术的精华。首先作为研究对象的自然数本身是递归生成的，因为每个自然数可看作是由零经有限次后继运算(或加“1”运算)而得；所处理的函数主要是使用递归的方法生成的，而这种函数在某个变目(或某组变目)处的值都可由初始值(即在零处的值)经有限次的已知运算逐步而得。另外所使用的证明方法基本上是一种归纳的方法或递归的方法：先证明在初始情况下某等式成立，然后假定当前情况的等式成立去证明后继情况的等式亦成立。

本书仅涉及原始递归算术或最后可化为原始递归算术处理的那些更复杂的递归算术。至于不能化归为原始递归算术的多重递归算术乃至一般递归算术目前还未有深入的研究。尽管如此，它得出的成果已几乎包括了整个初等数论。

对递归算术的研究至少有下列三方面的意义。

首先，对递归函数的研究是使用直观上的逻辑，而且对所假定的公理和规则都未曾详细地列出，以致我们不能明白在什么公理系统下获得所讨论的结果。最明显的一点是：通常总把存在公理当作自明的，不必明白提出的。例如，要把由递归式

$$\begin{cases} \varphi(u,0)=f(u) \\ \varphi(u,\ x+1)=g(u,\ x,\ \varphi(u,\ x)) \end{cases} \qquad (甲)$$

定义的二元函数φ改用复迭式定义，我们总是先令

$$\psi(u,\ x)=J(J(u,\ x),\ \varphi(u,\ x))。$$

其中$J(u,\ x)$以及下面的Kx，Lx是任意一组配对函数组，即满足下列两个条件：

$$KJ(u,\ x)=u,$$

$$LJ(u,\ x)=x$$

的三个函数。进而推出

$$\psi(u,\ 0)=J(J(u,\ 0),\ \varphi(u,\ 0))=J(J(u,\ 0),\ f(u))\equiv g_1(u),$$

$$\psi(u,\ x+1)=J(J(u,\ x+1),\ \varphi(u,\ x+1))=J(J(u,\ x+1),\ g(u,\ x,\ \varphi(u,\ x)))$$
$$=J(J(K^2\psi(u,\ x),\ LK\psi(u,\ x)+1),\ B(K^2\psi(u,\ x),\ LK\psi(u,\ x),\ L\psi(u,\ x)))$$
$$\equiv g_2(\psi(u,\ x))。$$

从而说：先由复迭式如下定义二元函数ψ：

$$\begin{cases}\psi(u,\ 0)=g_1(u)\\ \psi(u,\ x+1)=g_2(\psi(u,\ x))\end{cases}\qquad(乙)$$

然后得

$$\varphi(u,\ x)=L(\psi(u,\ x))。$$

严格说来，上面的结果只是说，假定递归式(甲)能够定义φ，那么φ也可改用复迭式定义。但这并不保证：如果复迭式(乙)可以定义ψ，则$L(\psi(u,\ x))$必然满足(甲)。因此若只知由复迭式可以引入函数，由上述证明并不能保证由原始递归式也可以引入函数。

明白地把日常推理过程中所使用的公理和规则分析出来，这是(原始)递归算术的第一个任务。

其次，通常所使用的直观上的逻辑是一种很强的逻辑，它不但使用命题演算一阶谓词演算(或狭义谓词演算)，有时还使用高阶谓词演算，甚至还使用元数学的分析方法。前面已提到，递归算术首先排除量词的使用，从而只限于命题演算。再进一步则不使用命题演算，纯粹由本身的公理及规则(这些公理和规则是纯数学的，不含任何逻辑的成分)来推演。这样自从发展递归算术后，人们可以说不必依靠任何逻辑来发展数学，从而逻辑主义者所认为的逻辑先于数学，应该把数学化归到逻辑等说法便值得怀疑了，故独立发展数学便是它的第二任务。

再次，递归算术的公理十分简单，比初等几何的公理系统远为优越。就公理的训练来说，递归算术远远优于初等几何，而与纯逻辑不相上下(上面已提到，递归算术实质上是一种方程演算，即是一种带等词的函数演算，它涉及的唯一的谓词是常谓词"="。与带等词的一阶谓词演算比较，递归算术除不用谓词字母外，还不用量词)。因此对递归算术进行研究实在是一件非常重要而有意义的事。

§0.2 形式系统的刻划

递归算术是作为一个形式系统而建立的。要建立一个形式系统，首先要列出相应的形式语言，它由两部分组成：(1)一些基本符号，其中有逻辑符号与非逻辑符号(或数学符号)。(2)由基本符号构成该系统所研究的对象所使用的组成规则(或文法)。由基本符号按组成规则所构成的该系统的研究对象称为合式公式。如果把基本符号的一个有限序列称为符号串的话，合式公式就是一个特殊的符号串。一些不同的形式系统可以有相同的形式语言，但由于组成规则不同，因此得到不同的形式系统。也即同一个形式语言的不同合式公式集合构成不同的形式系统，从而构成不同的理论。

列出了一个形式系统的形式语言后，就要作语义和语法的研究。语义的研究就是要对基本符号给以解释，对逻辑符号要给以逻辑上的解释，对数学符号要给以数学上的解释。然后给出在某个解释中(或代数结构中)一个合式公式为真和为假的概念，以及一个合式公

式永真的概念。一切在某个解释(或代数结构)中为真的合式公式(简称真公式)组成某个具体的理论。语法的研究，就是要对某个解释中的一切真公式作出一个公理系统的刻划。这种刻划也由两部分组成：①列出一个特殊的合式公式集，其中每个合式公式称为一个公理(或公理模式)。这些公理中所含的基本符号都是认为不加定义的原始符号。可以说这些原始符号被这些公理隐定义了，也即它们的含义就是由这些公理本身刻划了。

②列出一些推理规则或合式公式的变形规则，它们规定了由哪些合式公式可以推出什么样的合式公式，或对哪些合式公式作变形后可得到新的合式公式。如果一个合式公式的有限序列中的每个合式公式或者是一个公理，或者是由这个序列的前面若干个合式公式根据推理规则而得，那么称此有限序列为一个证明。一个证明的最后一个合式公式称为定理(相应的有限序列也称为此定理的证明)。

对于一个形式系统的形式语言的语法研究，一般有几种要求，首先要求希望所建立的公理系统是可靠的，也即此公理系统是就相应的语义研究中所有真公式而作的。具体地说，是希望相应公理系统中的定理是语义研究中的真公式，也即对于每个合式公式 α，如果 α 是一个定理，那么 α 是一个真公式。要达到此目的，只需要求在语法研究中，所给出的公理是相应语义研究中的真公式，另外要求在语法研究中所给出的推理规则是保真的，也即由真公式推出真公式。

其次，要求希望所建立的公理系统是相容的，或无矛盾的。不要在相应的公理系统中得到两个互相矛盾的定理，也即不要推出两个公式 α、β，使得在相应的语义研究中一个为真公式，一个为假公式。如果在一个形式系统中有"否定"的概念，那么当相应公理系统满足可靠性时，只要在语义中规定一个真公式的否定是假公式，则相应的公理系统必是相容的。另外，一个公理系统(的公理之间)最好是独立的。也即在公理中没有多余的公理，少了哪个公理都不行。不过一个公理系统的公理之间不相互独立亦未尚不可。系统的独立性要求相当于"锦上添花"。最后对一个公理系统的完备性也是必须要研究的。如果在公理系统中能把所有语义中的真公式都推出，也即若 α 为真公式，则 α 是相应公理系统的定理，就称此系统是完备的。当然一个公理系统如果既完备，又可靠，那么对相应形式系统的语法与语义的刻划就可得到了完美的统一。但遗憾的是，大部分的形式系统的语法刻划都是不完备的。现在已经证明，只要在一个形式系统中能把算术的基本概念，如自然数、加法、乘法等表达出来，那么对此形式系统就不可能有一个完备的语法刻划，也即不可能构造出一个完备的公理系统。本书中将要研究的原始递归算术就是一个不能作完备的公理刻划的形式系统。

更精确地说，这个系统是一个自由变元的纯方程演算，或是一个函词演算。其基本符号有五类：

① 可数无穷多个数变元。

② 对每个正整数 n，可数无穷多个 n 元函数字母。

③ 一个常谓词字母"="；也可把这个等号"="看作为一个逻辑符号。

④ 一个常数 0，两个一元常函数字母 S，O，可数无穷多个 n 元常数字母 I_m^n ($m=1, 2, \cdots$)。

⑤ 辅助性符号(左、右括号，逗号)。

这个系统的合式公式都是形为 $t=s$ 的符号串(称为方程的符号串)，其中 t，s 为项。所

谓项归纳地定义为：0与某个数变元为项；

若f为某个n元函数字母，t_1…，t_n为项，则$f(t_1, \cdots, t_n)$为项。

若t为项，则$(S(t), O(t)$为项。

若$t_1, \cdots, t_n$为项，则$I_m^n(t_1, \cdots, t_n)$为项。

项仅限于此。

第 1 章　有关概念的非形式描述

§1.1 自然数，数变元，数论函数

本书所研究的基本对象是自然数。每个自然数用相应的数字表示，0 是第一个自然数，其他的自然数都是 0 经过有限次的“＋1”运算而得到。于是由 0 经一次“＋1”运算得数字 0＋1，由 0 经二次“＋1”运算得0＋1＋1，一般由 0 经 n 次“＋1”运算得数字 $0+\underbrace{1+\cdots+1}_{n\text{个}1}$。

为简单起见，把每个数字用十进制表示法加以缩写，于是得自然数的缩写 0，1，2，3，4，5，…。

任何一门严密的学科都大量使用变元。变元是用来代替某集合中的任何一个对象的，或者说变元表示某集合中的任何一个对象。与每个变元相伴随的集合称为此变元的变域。因此要刻划一个变元必须要同时给出此变元的名称与相应的变域，而变域中的任何一个对象都是相应变元的一个变值。本书所研究的数变元是以自然数集为变域的变元。

对任意一个数字 n 作定义是困难的，不得不使用“…”那样不确定符号。为了避免这种情况，就要使用数变元。例如，若使用数变元 x，则任意一个数字可定义如下：由 x 出发，进行 n 次 $x+1$ 代以 x 的运算，接着用零代以 x，即可得数字 n。

本书中涉及的函数都是处处有定义的(即全的)数论函数，也即这样的一个 n 元函数：

$$f(x_1, \cdots, x_n)\ (n=1, 2, \cdots)。$$

它的定义域是全体由自然数组成的 n-矢的集合，即一切形为 $\langle a_1, \cdots, a_n\rangle$ 的集合，其中各 a_i 为自然数；而它的值域是全体自然数。关于函数符号位置的书写，对于一元函数而言，用前置法，例 $f(x)$ 或 fx。对于二元函数而言，用中置法，例如，加法函数是 $x+y$。对于三元以上(包括三元)的函数或一般形式的 n 元函数，仍用前置法，例如，$f(x_1, \cdots, x_n)$ 或 $fx_1\cdots x_n$。实际上是把函数的命名式作为一个函数的表示。因此粗略地说，函数是由(数)变元组成的式子，或者说一个函数是借助于变元定义的在数上的运算。我们也可把函数命名式 $f(x_1, \cdots, x_n)$ 作如下解释：此函数的名称是 f，依次有 n 个空位，其中第 i 个空位 ($i=1, \cdots, n$) 的名称是 x_i。当每个空位处分别填以数 $a_1, \cdots, a_n$ 后，得函数的填式或函数值 $f(a_1, \cdots, a_n)$。

§1.2　函数的定义

定义一个函数有两种方法。一种方法是用公理确定，对一些最基本的函数才用公理确定它们的性质。实质上是把这些函数作为原始概念，因此本质上并未对它们加以定义。另一种方法是使用一些算子由若干已知函数去确定一个新的函数，这才是真正对一个函数作定

义。本书所用的定义新函数的算子有两种：一种称为迭置或代入，另一种称为递归式。

最典型的迭置是如下的标准迭置：已知一个 n 元函数 $f(x_1, \cdots, x_n)$，n 个 m 元函数 $g_i(y_1, \cdots, y_m)(i=1, \cdots, n)$，得到一个 m 元的新函数 $h(x_1, \cdots, x_m)$，使得 h 在任意数 $a_1, \cdots, a_m$ 处的值由

$$h(a_1, \cdots, a_m)=f(g_1(a_1, \cdots, a_m), \cdots, g_n(a_1, \cdots, a_m))$$

求得。这时称函数 h 是由函数 f, $g_1, \cdots, g_n$ 使用标准迭置而定义。这种类型的定义也称为显式定义。

除了上述的标准迭置外，还有一种简单的迭置：

已知一个 K 元函数

$$f(y_1, \cdots, y_k)。$$

$x_1, \cdots, x_n$ 为不同的变元。设 $z_i(i=1, \cdots, k)$ 为各 x_i 之一，则可得新函数

$$h(x_1, \cdots, x_n)=f(z_1, \cdots, z_k)。$$

这种简单的迭置包括添加虚变元(即把少变元的函数看作多变元的函数)，例如

$$h(x_1, x_2, x_3)=f(x_1, x_2),$$

包括交换变元，例如

$$h(x_1, x_2)=f(x_2, x_1),$$

以及等同变元，例如

$$h(x_1)=f(x_1, x_1)。$$

这种定义也可称为 h 由 f 显式定义。

一般来说，显式定义是由一个方程表示，方程的右边是一个项，其中出现的函数是已知函数，而方程的左边仅是单个新函数(也为项)。这个方程称为左边新函数的定义方程。另外出现在方程右边的每个变元必定也出现在左边的新函数中，但左边还可以出现新的变元(称为虚变元)。

关于使用递归式定义新函数的方法一般由两个等式组成，这两个等式称为相应新函数的定义方程，其中第一个等式确定(借助于已知函数)新函数在某个值位处取零的值。第二个等式指出(借助于已知函数)如何由新函数已取得的值而得到还未取得的值。

最基本的递归式是原始递归式，它的两个定义方程形式如下：

$$h(u_1, \cdots, u_n, 0)=f(u_1, \cdots, u_n), \qquad (1)$$

$$h(u_1, \cdots, u_n, x+1)=g(u_1, \cdots, u_n, x, h(u_1, \cdots, u_n, x))。 \qquad (2)$$

其中，n 元函数 f 与 $(n+2)$ 元函数 g 是已知函数，而 $(n+1)$ 元函数 $h(u_1, \cdots, u_n, x)$ 是新函数，或被定义的函数，其中变元 $u_1, \cdots, u_n$ 称为 h 的参数，变元 x 称为 h 的递归变元。式(1)给出了 h 在递归变元 x 取零时的值。式(2)给出了一个由 h 在递归变元 x 取任意值 a 时的值计算 h 在递归变元 x 取值 $a+1$ 时的值的过程。于是由 h 的这两个定义方程可以计算 h 在递归变元 x 取任意值 a 时的值。易见，计算 h 在 $\langle a_1, \cdots, a_n, a\rangle$ 处的值 $h(a_1, \cdots, a_n, a)$ 的过程，与在§1.1中指出的由 x 出发不断使用两个代入运算以便得到数字 a 的过程完全相对应。

除原始递归式外，还有一些递归式，粗看起来比原始递归式复杂，但实质上仍是原始递归式，即可化归为原始递归式。这将在以后再详细介绍。