



## 第2章 从头开始：自然数

在本书中，我们将回顾曾在高中以及初等微积分课程上学过的那些知识，但是我们会按照尽可能严格的要求来进行。为此，我们将不得不从最基础的内容开始学习。事实上，我们要追溯到数的概念以及数有哪些性质。不可否认，你已经与数字打交道十余年了，并且了解如何利用袋鼠法则化简含有数字的表达式。但是，现在我们将转向一个更加基础的问题，那就是：为什么代数法则总是起作用？例如，对于任意三个数  $a$ 、 $b$ 、 $c$ ，为什么  $a(b+c)$  等于  $ab+ac$  总是成立？这种运算法则并不是随意给出的，它能够从数系更为原始也更为基本的性质中推导出来。这将教会你一种新的技能——利用较为简单的性质证明复杂的性质。你会发现，虽然某个命题可能是“显然的”，但是要证明它却好像并不是那么容易。本书中将给出大量这样的练习，另外在这个过程当中，你会被引导着思考为什么一个显然成立的命题的确是显然的。这里你将掌握一个技能是使用数学归纳法。数学归纳法是很多数学领域中都会用到的一种基本的证明工具。

因此，在这本书的前几章中，我们将让你重新认识一下实分析中用到的各种数系。随着复杂程度的不断提高，这些数系分别是自然数系  $\mathbf{N}$ 、整数系  $\mathbf{Z}$ 、有理数系  $\mathbf{Q}$  以及实数系  $\mathbf{R}$ 。（还存在其他的数系，如复数系  $\mathbf{C}$ ，但在 15.6 节之前，对它们不作研究。）自然数系  $\{0, 1, 2, \dots\}$  是所有数系中最原始的一个，但是自然数并用来构造整数，然后整数又被用来构造有理数。更进一步，有理数被用来构造实数，而实数被用来构造附属。于是，要想从最初的内容开始，我们必须考察自然数。我们将考虑如下问题：人们如何真正地定义自然数？（这与如何使用自然数是完全不同的两个问题，显然，使用自然数是你非常擅长的事情。这就好比知道如何使用一台计算机与知道如何组装一台计算机是完全不同的两码事。）

对于这个问题的回答要比看起来难得多。根本的问题在于，你使用自然数已经很久了，以至于自然数在你的数学思维中根深蒂固，你会无意识地对自然数做出各种隐含的假设（例如， $a+b$  总是等于  $b+a$ ）。另外，像第一次见到自然数系那样去考察它是非常困难的。因此，下面我们将让你完成一个难度相当大的任务：尝试把你所了解的所有关于自然数的知识暂时放在一边，同时忘记如何计数、求和、求乘积以及如何使用代数法则等。我们将尝试逐个引入这些概念并在学习过程中明确地标识出我们的假设都有哪些。此外，在真正地证明出那些更加“高级”的技巧（比如代数法则）之前，我们都不能使用这些技巧。这看起来可能是一个令人烦恼的限制，特别是当我们需要花费大量时间去证明那些“显然”的命题时，但这种把已知事实暂时放在一边的做法对于避免循环论证（例如，用高等的知识去证明较初等的理论，然后又用该初等理论去证明之前的高等知识）是非常必要的。同时，这种训练也是夯实数学基础知识的一种非常好的方式。更进一步来说，当我们的课程进展到更加高深的概念，如实数、函数、序列和级数、微分以及积分等时，这里对证明和抽象思维能

力的训练将对你有很大的帮助。总之，此处所阐述的结果看起来好像是无关紧要的，但是目前对于我们来说，过程比结果更重要。（一旦构造出合适的数系，我们就可以重新使用代数法则等，而不需要在每次使用它们之前都进行重新推导。）

我们也要忘掉所了解的十进制。虽然十进制是我们处理数时所采用的一种极为便捷的方法，但对于什么是数这个问题而言，十进制并不是最基本的内容。（例如，人们可以使用八进制或者二进制、甚至罗马数字系统来代替十进制，而且这些进制所得到的数的集合是完全相同的。）另外，尝试去彻底解释什么是十进制并非你想象中那么自然。为什么 00423 与 423 表示的是同一个数，而 32400 与 324 所表示的却不是同一个数？为什么 123.4444... 是一个实数，但 ...444.321 却不是实数？当数相加或者相乘的时候，我们为什么要进位？为什么说 0.999... 与 1 表示的是同一个数？最小正实数是多少？它是不是 0.00...001？因此，为了暂不涉及这些问题，我们不会给出关于十进制相关知识的任何假设。然而，对数的描述我们将继续沿用数系的数字符号，如 1, 2, 3 等，而不去使用其他诸如 I, II, III 或者 0++, (0++)++, ((0++)++)++（参见下文）等没必要的人工记号。为了保证完整性，在附录 B 中我们会对十进制进行回顾。

## 2.1 皮亚诺公理

现在我们根据皮亚诺公理给出定义自然数的一种标准方法，其中皮亚诺公理是由朱塞佩·皮亚诺（1858—1932）首次提出的。但这并非定义自然数的唯一方法。比如另一种方法是通过讨论有限集合的势给出的。比方说，给定一个含有五个元素的集合，我们可以定义 5 来表示这个集合中含有的元素个数。在 3.6 节中，我们将对这种定义自然数的方法进行讨论。但现在我们继续讨论根据皮亚诺公理定义自然数的方法。

如何定义什么是自然数呢？我们可以通俗地说，

定义 2.1.1  $\neg$  非正式的  $\neg$  自然数 是集合

$$\mathbf{N} := \{0, 1, 2, 3, 4, \dots\}$$

的元素。其中，集合  $\mathbf{N}$  是由从 0 开始，无休止地往前进行计数所得到的所有元素构成的集合。我们称  $\mathbf{N}$  为 自然数集。

注 2.1.2 在有些教材中，自然数被定义为从 1 开始，而非从 0 开始，但这仅仅是一种符号约定而已。在这本书中，我们定义集合  $\{1, 2, 3, \dots\}$  为正整数集并记作  $\mathbf{Z}^+$ ，而不是自然数集。自然数有时也被称为 完整数（whole number）。

从某种意义上来说，上述定义给出了什么是自然数这个问题的答案：自然数是集合<sup>[1]</sup>  $\mathbf{N}$  中的元素。然而这个定义并非真的那么令人满意，因为它带来了一个新的问题： $\mathbf{N}$  是什么？“从 0 开始，无休止地进行计数”看起来好像是对  $\mathbf{N}$  的一个足够直观的定义，但是这种叙述带来很多尚未回答的问题，因此它并不能被人们完全接受。例如：如何确定我们能够无休止地进行计数而不会出现循环回到 0 的情况？你应该如何进行诸如加法、乘法以及指数运算这样的运算？

首先，我们可以回答第二个问题：我们能够利用较为简单的运算来定义复杂的运算。指数运算只不过是重复地进行乘法运算： $5^3$  就是把 3 个 5 乘在一起。乘法运算只不过是重复地进行加法运

算： $5 \times 3$  就是把 3 个 5 加在一起。（由于减法和除法运算不是完全适用于自然数的运算，因此这里不对它们做论述；只有当我们学到整数和有理数的时候，才会分别对减法和除法进行讨论。）那么加法有时如何定义的呢？它不过是重复地往前进行计数或不断增长的运算。如果你把 5 加上 3，那么你所做的就是让 5 增长了三次。另外，增长看起来是一个基本运算，它无法简化为更加简单的运算；事实上，增长是人们在学习数时首先接触到的运算，它甚至出现在学习加法运算之前。

于是，为了定义自然数，我们将使用如下两个基本概念：数 0 和增量运算。为了与现代计算机语言相一致，我们用  $n++$  来表示  $n$  的增量或紧跟在  $n$  之后的数，例如， $3++ = 4$ ， $(3++)++ = 5$  等。这与  $n++$  在计算机语言中的用法稍有不同。例如，在 C 语言中  $n++$  实际是把  $n$  重新赋值为紧跟在  $n$  之后的那个数。但是在数学中，我们在任何情况下对一个变量仅定义一次，这是因为如果对一个变量定义多次经常会发生混淆的状况；很多命题可能对某变量原有的赋值为真，但是对于该变量新的赋值而言就会变成假的，反之亦然。

因此，这看起来貌似我们要说  $\mathbf{N}$  是由 0 以及所有能通过增量运算由 0 得到的数构成的： $\mathbf{N}$  应该由如下这些对象构成：

$$0, 0++, (0++)++, ((0++)++)++, \dots$$

如果我们把自然数用上述对象来表述，那么将得到如下关于 0 和增量运算  $++$  的公理。

**公理 2.1** 0 是一个自然数。

**公理 2.2** 如果  $n$  是一个自然数，那么  $n++$  也是一个自然数。

于是，举例来说，我们利用一次公理 2.1 和两次公理 2.2，能够推得  $(0++)++$  是一个自然数。显然这个记号将变得越来越不灵便，因此我们共同约定采用更熟悉的记号来表示这些数。

**定义 2.1.3** 我们定义 1 为数  $0++$ ，2 为数  $(0++)++$ ，3 为数  $((0++)++)++$ ，等等。┐换言之， $1 := 0++$ ， $2 := 1++$ ， $3 := 2++$ ，等等。在本书中，我用“ $x := y$ ”来表示命题“令  $x$  的值等于  $y$ ”。┐

于是，举例来说，我们有：

**命题 2.1.4** 3 是一个自然数。

**证明：** 根据公理 2.1 可知，0 是一个自然数。根据公理 2.2， $0++ = 1$  是一个自然数。再次利用公理 2.2 就可以得到， $2++ = 3$  是一个自然数。□

目前看来我们似乎已经对自然数进行了充分的描述，但是还没有彻底弄清楚  $\mathbf{N}$  的特性。

**例 2.1.5** 考察一个由数 0、1、2、3 所构成的数系。在这个数系中，增量运算将从 3 绕回到 0。用更加准确的语言来描述就是， $0++$  等于 1， $1++$  等于 2， $2++$  等于 3，但是  $3++$  等于 0。

(通过给出 4 的定义，可以得到  $3++$  也等于 4)。当人们尝试在计算机中存储一个自然数的时候，这种“绕回”的状况在现实生活中也会发生：如果我们从 0 开始反复运行增量运算，最终计算机内存将溢出并且数将归为 0（尽管导致这种状况发生所需运行的增量运算的次数可能会非常庞大，比如只有在运行 65 536 次增量运算后，一个整数的双字节表示才会绕回到 0）。我们注意到这种类型的数系遵守公理 2.1 和公理 2.2，尽管这显然与我们凭直觉所感知的自然数应该具备的特性并不一致。

为了防止出现这种“绕回状况”，我们将引入另一个公理。

**公理 2.3** 0 不紧跟在任何自然数之后。换言之，对任意一个自然数  $n$ ， $n++ \neq 0$  均成立。

现在我们可以证明某些绕回状况不会发生，比如利用下述命题，我们可以阻止例 2.1.5 中绕回状况的发生。

**命题 2.1.6** 4 不等于 0。

不要笑！因为我们是按照这样的方式来定义 4 的——4 是把 0 增长、增长、增长、在增长之后得到的，所以数字 4 和 0 不是同一个数这个先验命题（a priori）并不一定为真，尽管该命题为真是“显然的”。（a priori 是 beforehand 的拉丁语，指人们在开始进行证明或论述之前，就已经知道或假定其为真的命题。它的反义词是 a posteriori，表示人们通过证明或论述后，才能确定其为真的命题。）在例 2.1.5 中的情况，4 实际上是等于 0 的，并且在自然数的标准双字节计算机表示中，以 65 536 为例，它是等于 0 的（根据我们对 65 536 的定义，65 536 表示把 0 运行 65 536 次增量运算之后得到的结果）。

**证明：** 由定义知， $4 = 3++$ 。根据公理 2.1 和公理 2.2 可知 3 是一个自然数，因此根据公理 2.3 可得， $3++ \neq 0$ ，即  $4 \neq 0$ 。□

然而，即便现在有了新的公理，我们的数系仍然可能表现出其他病态特性。

**例 2.1.7** 考虑由 0 `1 `2 `3 `4 这五个数构成的数系。在这个数系中，增量运算在数 4 处碰到了“天花板”。更准确地说，令  $0++ = 1$ ， $1++ = 2$ ， $2++ = 3$ ， $3++ = 4$ ，但令  $4++ = 4$ （换言之，有  $5 = 4$ ，从而  $6 = 4$ ， $7 = 4$ ，等等）。这与公理 2.1、公理 2.2 以及公理 2.3 并不矛盾。有类似问题的另一个数系是这样的：它的增量运算同样存在绕回状况，但并不是绕回到 0，例如，令  $4++ = 1$ （那么可以得到  $5 = 1$ ，进而得到  $6 = 2$ ，等等）。

有许多方法可以阻止上述情况的发生，但是最简单的方法之一是假定下面这个公理成立。

**公理 2.4** 对于不同的自然数而言，紧跟在它们之后的数字也一定是不同的。也就是说，如果  $n$  和  $m$  都是自然数，并且  $n \neq m$ ，那么  $n++ \neq m++$ 。等价说法为<sup>[2]</sup>，如果  $n++ = m++$ ，那么一定有  $n = m$ 。

举例来说，我们得到：



命题 2.1.8  $6 \neq 2$ 。

证明：为了推出矛盾，不妨假设  $6 = 2$  成立。那么有  $5++ = 1++$ ，故根据公理 2.4 可得， $5 = 1$ ，进而有  $4++ = 0++$ 。再次利用公理 2.4，我们有  $4 = 0$ ，这显然与之前命题 2.1.6 相矛盾。

正如我们从该命题中看到的那样，看起来好像所有的自然数都是两两不相等的。但是现在仍然存在这样一个问题：虽然根据公理（特别是公理 2.1 和 公理 2.2）我们能够确信  $0, 1, 2, 3, \dots$  是  $\mathbf{N}$  中的不同元素，但是仍旧存在一个问题，那就是可能还有另外的不是上述形式的“流氓”元素存在于我们的数系当中。

例 2.1.9（非正式的）假设我们数系  $\mathbf{N}$  是由如下所示的整数和半整数共同构成的：

$$N := \{0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, \dots\}$$

（这个例子之所以被标注为“非正式的”是因为我们在这里使用了实数，但是目前我们不应该使用实数。）我们能够验证该集合仍旧满足公理 2.1~公理 2.4。

我们希望得到这样一个公理，它告诉我们  $\mathbf{N}$  中所有的元素都可以通过对 0 进行增量运算得到，从而能够从  $\mathbf{N}$  中排除像 0.5 这样的元素。但是若不利用我们正尝试去定义的自然数，想要把我们所说的“通过  $\dots$  得到”这句话量化地解释清楚是困难的。幸运的是，存在一种巧妙的方案试着去处理此事。

**公理 2.5（数学归纳法原理）** 令  $P(n)$  表示自然数  $n$  的任意一个性质，如果  $P(0)$  为真且  $P(n)$  为真时一定有  $P(n++)$  也为真，那么对于任意自然数  $n$ ， $P(n)$  一定为真。

注 2.1.10 此处的“性质”到底指的是什么，我们还不清楚。但是  $P(n)$  有如下一些例子：“ $n$  是偶数”“ $n$  等于 3”“ $n$  是方程  $(n+1)^2 = n^2 + 2n + 1$  的解”，等等。当然我们还没有定义其中的许多概念。但是当我们给出定义后，公理 2.5 将能够应用到这些性质中。[一个逻辑上的备注：由于这个公理不仅涉及变量，同时也涉及性质，所以它在本质上与其他四个公理是不同的。事实上，公理 2.5 严格来说应该被称为公理模式而非公理——它是一个模板而不是一个独立的公理，在这个模板上可以构造出很多个（无穷多个）公理。进一步讨论其中的区别将远远超出本书的范围，而且这种讨论会涉及逻辑学的相关内容。]

隐藏在这个公理背后的通俗的直观说明如下。假设  $P(n)$  满足如下条件： $P(0)$  为真，且  $P(n)$  为真时， $P(n++)$  一定为真。那么由于  $P(0)$  为真，所以  $P(0++) = P(1)$  为真。又因为  $P(1)$  为真，所以  $P(1++) = P(2)$  也为真。通过不断重复这个过程，我们可以得到  $P(0), P(1), P(2), P(3), \dots$  全为真。然而根据上述推导过程，以  $P(0.5)$  为例，我们永远得不出  $P(0.5)$  为真。因此，对于包含如 0.5 这样“非必要”元素的数系，公理 2.5 将失效。（实际上，我们能够给出关于这个事实的“证明”。把公理 2.5 应用到具有如下性质的  $P(n)$  中：“ $n$  不是一个半整数”，即  $n$  不是一个整数加上 0.5。于是可以推出  $P(0)$  为真，且当  $P(n)$  为真时  $P(n++)$  一定为真。那么公理 2.5 说明了对任意自然数  $n$ ， $P(n)$  均为真。换言之，所有自然数都不是半整数。特别的 0.5 不可能是自然数。由于我们尚未定义诸如“整数”“半整数”以及“0.5”这类概念，因此这个“证明”并非真正意义上的证明。但是关于归纳法原理是如何将所有不是“真”自然数的数排除在  $\mathbf{N}$  之外的，该证明应该给了你一些启示。）

归纳法原理为我们提供了一种证明某性质  $P(n)$  对任意自然数  $n$  均为真的方法。因此在本书剩余部分中，我们将看到许多具有如下形式的证明过程。

**命题 2.1.11** 对任意的自然数  $n$ ，某性质  $P(n)$  恒为真。

**证明：** 我们利用归纳法来证明。首先证明最基本的  $n = 0$  时的情况，结论成立，也就是证明  $P(0)$  为真。（此处插入关于  $P(0)$  为真的证明。）现在归纳性地假设  $n$  是一个自然数，同时  $P(n)$  已经证明为真。接下来我们证明  $P(n++)$  也为真。（在已知  $P(n)$  为真的前提下，插入关于  $P(n++)$  为真的证明。）到这里整个归纳过程就结束了，因而对所有的自然数  $n$ ， $P(n)$  恒为真。  $\square$

当然，我们没有必要严格地按照上述证明过程中所采用的模板、措辞以及语句顺序来展开证明，但是利用归纳法做出的证明一般都与上面这种形式相同。后面我们还将遇到一些其他的归纳法变体，例如，逆向归纳法（参见习题 2.2.6）、强归纳法（参见习题 2.2.14）和超限归纳法（参见引理 8.5.15）。

公理 2.1~公理 2.5 被称为关于自然数的皮亚诺公理。这几个公理看起来非常可信，于是我们给出下面这个假设。

**假设 2.6**（非正式的）存在一个数系  $\mathbf{N}$ ，我们称  $\mathbf{N}$  中的元素为自然数，而且公理 2.1~公理 2.5 对  $\mathbf{N}$  均成立。

在下一章中，一旦规定了在集合与函数中所使用的记号，我们将进一步对这个假设作出更精确的描述。

**注 2.1.12** 我们把这个数系  $N$  称为自然数系。当然，我们会考虑到这样一种可能的情况：存在多个自然数系，比如，印度—阿拉伯数系  $\{1, 2, 3, \dots\}$  和罗马数系  $\{O, I, II, III, IV, V, VI, \dots\}$ ，而且嫌麻烦的话，我们可以把它们看作不同的数系。但是，这些数系显然是完全等价的（用专业语言来描述是同构的），这是因为我们可以构造出一一对应关系： $0 \leftrightarrow O$ ， $1 \leftrightarrow I$ ， $2 \leftrightarrow II$ ，等等，这个映射把印度—阿拉伯数系中的零对应到罗马数系中的零，并且还保持了增量运算（例如，如果把 2 对应到 II，那么  $2++$  将对应到  $II++$ ）。对这种类型的等价关系更加精确的表述，参见习题 3.5.13。因为自然数系的所有变形都是等价的，所以认为存在不同的自然数系是毫无意义的，并且我们只使用单独一个自然数系去做数学研究。

对于假设 2.6，我们将不做证明（尽管最终我们会把它包含在有关集合论的公理中，参见公理 3.7），并且它是我们做出的唯一一个关于数系的假设。现代分析论中非常引人注目的一个成就是，只从上述五个非常基本的公理和集合论中的某些附加公理出发，就能构造出其他所有的数系、生成函数，并进行我们通常所做的全部代数和微积分研究。

**注 2.1.13**（非正式的）自然数系有一个非常有趣的特点：尽管每一个自然数都是有穷的，但是由自然数所构成的集合却是无穷大的；也就是说，虽然  $\mathbf{N}$  是无穷大的，但是  $\mathbf{N}$  是由各个有穷的元素构成的（整体大于它的任意部分）。不存在无穷大的自然数；倘若比较熟悉有穷和无穷的概念，那么我们可以利用公理 2.5 来证明这种说法。（显然 0 是有穷的。另外，如果  $n$  是有穷的，那么  $n++$  自然也是有穷的。因此，根据公理 2.5 可知，所有自然数都是有穷的。）这样看来，自然数系能够趋向于无穷大，但是它不可能真的取到无穷大；无穷大不是自然数。（存在其他的数系，使得“无穷大”是该数系的元素，例如：基数系、序数系以及  $p$  进数系，然而它们并不遵循归纳法原

理，而且完全不在本书的讨论范围之内。)

注 2.1.14 注意，我们对自然数的定义是公理化的而非构造性的。我们还没有告诉你什么是自然数（为此，我们不提出下面这样的问题：数是由什么组成的，它们是物理对象吗，它们度量的对象是什么，等等）。我们只列出一些你利用自然数可以做的事情（事实上，目前我们已经定义过的自然数上的运算只有增量运算）以及自然数所具备的一些性质。数学研究就是这样进行的：将研究对象进行抽象处理，只关心研究对象具备什么样的性质，并不在乎研究对象是什么或者它们意味着什么。如果人们想做数学研究，那么一个自然数是否表示算盘珠子的某种排列，或者计算机内存中字节的某种编排，又或者某些不具备物理形态的更加抽象的概念，这都无关紧要。只要你能让它们增长，能够判断它们中的任意两个是否相等，并且对它们可以做相加和相乘这样的算数运算，那么从数学研究的目标来说，它们就是真正的数（当然，前提是它们能够满足那些必要的公理。）从其他的数学对象出发来构造自然数也是有可能的。例如，从集合出发来构造自然数，然而有多种方法构造一个自然数系的有效模型可以，而争论到底哪个模型是“真”的，至少对一个数学家来说是毫无意义的。只要它满足所有的公理，并能够正确的运作，那么对于数学研究来说，它就是足够好的模型。

注 2.1.15 从历史的角度来说，实现对数的公理化处理是近代才发生的事情，距今只有一百多年。在那之前，数一般被认为不可避免地与某些外部概念密切关联，例如：计算集合的势、测量线段的长度或者计算物体的质量等。在人们的认知不得不从一个数系转向另一个数系之前，上面这种对数的理解是合理且有效的。比如，通过数珠子的方式去理解数，对数的概念化是很有好处的（例如通过数珠子很容易就形成 3 和 5 的概念），但是对于  $-3$ 、 $\frac{1}{3}$ 、 $\sqrt{2}$  或  $3+4i$  这样的数来说，数珠子的方法就不怎么起作用了。因此，数论中每一次伟大的进步——负数、无理数、复数甚至是数字 0——都会带来大量不必要的哲理烦恼。数可以通过公理来抽象地理解而不需要借助任何实物模型，这是 19 世纪后期的一个伟大发现。当然，在方便的情况下，数学家可以使用实物模型来帮助自己更好地展现直观认识并加深理解，但是当这些模型对研究造成阻碍的时候，它们也会被轻易地抛弃。

根据前面这些公理，可以得到这样一个结论：现在我们能够递归地定义序列。假设我们希望通过下列方式来构造一个数列  $a_0, a_1, a_2, \dots$ ：首先定义  $a_0$  的基值，例如  $a_0 := c$ ，其中  $c$  是某个固定的数。然后令  $a_1$  为关于  $a_0$  的某个函数， $a_1 := f_0(a_0)$ ，令  $a_2$  是关于  $a_1$  的某个函数， $a_2 = f_1(a_1)$ ，一次类推。一般地，我们记作  $a_{n++} := f_n(a_n)$ ，其中  $f_n$  是某个  $\mathbf{N}$  到  $\mathbf{N}$  的函数。利用前面所有的公理能够推出，对任意给定的自然数  $n$ ，上述过程将对数列中的元素  $a_n$  给出一个单一的值。更准确地说 [3]：



1. 严格来说，这个非正式的定义存在另外一个问题：我们尚未定义“集合”以及“元素”的概念。因此，在本章剩余的部分中，除了有关非正式的讨论内容，我们将尽量避免涉及集合以及集合中的元素。
2. 这是一个用逆否命题来重新阐述原命题涵义的例子，更多的细节见 A.2 节。从命题的逆向来考虑，如果  $n = m$ ，那么  $n++ = m++$ ；这就是被应用到  $++$  运算上的替换公理（见 A.7 节）。
3. 严格地说，该命题需要定义