Part 2. Adding one by one

3. Adding one by one

My colleague EHK[4] told me about a difficulty she experienced in her first encounter with arithmetic, aged 6. She could easily solve "put a number in the box" problems of the type

$$7 + \square = 12,$$

by counting how many 1's she had to add to 7 in order to get 12 but struggled with

$$\square + 6 = 11,$$

because she did not know where to start. Worse, she felt that she could not communicate her difficulty to adults. Her teacher forgot to explain to her that addition was commutative.

Another one of my colleagues, AB[5], told me how afraid she was of subtraction. She could easily visualise subtraction of 4 from 100, say, as a stack of 100 objects; after removing 4 objects from the top (by reverse counting: 100, 99, 98, 97), 96 are left. But what will happens if you remove 4 objects from the bottom of the stack?

A brief look at axioms introduced by Dedekind (but commonly called Peano axioms) provides some insight in EHK's and AB's difficulties.

3.1. Dedekind-Peano axioms. Recall that the Dedekind-Peano axioms describe the properties of natural numbers $\mathbb{N}$ in terms of a "successor" function $S(n)$. (There is no canonical notation for the successor function, in various books it is denoted $s(n)$, $\sigma(n)$, n', or even n^{++}, as in popular computer languages C and C^{++}.)

Axiom 1: 1 is a natural number.

Axiom 2: For every natural number n, $S(n)$ is a natural number.

Axioms 1 and 2 define a unary representation of the natural numbers: the number 2 is $S(1)$, and, in general, any natural number n is

$$S^{n-1}(1) = S(S(\cdots S(1)\cdots)) \quad (n-1 \text{ times}).$$

As we shall soon see, the next two axioms deserve to be treated separately; they define the properties of this representation.

Axiom 3: For every natural number n other than 1, $S(n) \neq 1$. That is, there is no natural number whose successor is 1.

Axiom 4: For all natural numbers m and n, if $S(m) = S(n)$, then $m = n$. That is, S is an injection.

The final axiom (Axiom of Induction) has a very different nature and is best understood as a method of reasoning about all natural numbers.

Axiom 5: If K is a set such that:

- 1 is in K, and
- for every natural number n, if n is in K, then $S(n)$ is in K,

then K contains every natural number.

[4] For the record: EHK is female, has a PhD in Mathematics, teaches mathematics at a highly selective secondary school.

[5] AB is female, has a PhD in Mathematics, teaches mathematics in a research-led university.

Thus, Dedekind-Peano arithmetic is a formalisation of that very counting by one that little EHK did, and addition is defined in precisely the same way as EHK learned to do it: by a recursion

$$\begin{aligned} m+1 &= S(m) \\ m+S(n) &= S(m+n). \end{aligned}$$

Commutativity of addition is a non-trivial (although still accessible to a beginner) theorem. To force you to feel some sympathy to poor little EHK and to poor little AB, I will prove it to you in the next section.

3.2. A brief digression: is 1 a number? Having postponed more serious work, we can spend a few minutes discussing Axiom 1: 1 *is a natural number.*

Even this axiom is not self-evident as it appears to be. In many languages, including English, the word 'number' can denote some collection or ensemble of objects with tacit understanding that it contains at least a few, and in any case more than one, objects. For example, a phrase

"A number of people feel that 1 is not a number"

makes sense and means that more than one person thinks that 1 is not a number. Such usage reflects an earlier stage of development of the system of numerals when 1 was not a number; numbers were made of ones, of basic units; but one is not made of ones.

What is very important for the history of mathematics, it appears that, for similar reasons, 1 was not a number for ancient Greek mathematicians, as evidenced in Euclid's *Elements*: Euclid careful separated the use of words 'number' and 'unit'.

And, as a digression within digression, I want to mention the issue of collective nouns—I shall discuss them again in later lectures, so the present deviation is not waste of time. The English language has a peculiar tendency to form or find a special word to denote groups of particular animals. For example, Englishmen say

a herd of cows,
a flock of sheep,
a pack of dogs,
a school of fish.

To illustrate how far things go, it will suffice to mention that ducks, while on water, form a *paddling*, while in flight they are a *flush.* Some nouns are absolutely obscure; for example, I found in Wikipedia a *sedge of bitterns*, but I do not even know what bitterns are.

Invention of collective nouns for groups of people from various professional groups is a popular genre of English humor; to my taste,

a number of mathematicians

appears to be one of the more obvious solutions.

4. Properties of addition

There are several alternative forms of notation for the successor function: $S(n)$, $s(n)$, $\sigma(n)$, n' and even n^{++}, the latter used in programming languages C and C++. I shall use notation n'; as the reader will soon see, it is very convenient—and natural—to write a symbol for the successor function *after* the number that has to be incremented.

Figure 9. Guido Reni. A fragment of *The Rape of Helena*, 1631. Musée du Louvre. Source: *Wikipedia Commons*. Public domain.

In this new notation, the recursive rule for addition looks like

$$n + 1 = n' \qquad (4)$$

$$n + m' = (n + m)'. \qquad (5)$$

I will also use Axiom 5 in a more conventional form, obviously equivalent to the original one.

Axiom 5: Assume that a certain statement about numbers If K is a set such that:

- the statement is true for 1 (*Basis of Induction*)
- if the statement is true for a natural number n (*Inductive Assumption*) then it is true for the next number n' (*Inductive Step*).

Then the statement is true for all natural numbers.

I will prove two canonical properties of addition.

4.1. Associativity of addition.

Theorem 4.1. *Assume that + is a binary operation which satisfies conditions* (4) *and* (5)*. Then + is associative, that is,*

$$(a + b) + c = a + (b + c)$$

for all a, b, c.

Proof. The proof will use induction on c.

Basis of Induction.

$$\begin{aligned}(a+b)+1 &\overset{\text{by (4)}}{=} (a+b)' \\ &\overset{\text{by (5)}}{=} a+b' \\ &\overset{\text{by (4)}}{=} a+(b+1).\end{aligned}$$

Inductive Assumption:

$$(a+b)+c = a+(b+c).$$

Inductive Step.

$$\begin{aligned}(a+b)+c' &\overset{\text{by (5)}}{=} ((a+b)+c)' \\ &\overset{\substack{\text{by inductive}\\ \text{assumption}}}{=} (a+(b+c))' \\ &\overset{\text{by (5)}}{=} a+(b+c)' \\ &\overset{\text{by (5)}}{=} a+(b+c').\end{aligned}$$

□

4.2. Commutativity of addition. We shall start with a very special, but crucially important case.

THEOREM 4.2. *Assume that + is a binary operation which satisfies conditions* (4) *and* (5). *Then*

$$1+a = a+1$$

for all a.

PROOF. We shall prove the theorem by induction on a.

Basis of Induction.

$$1+1 = 1+1.$$

There is nothing to prove here.

Inductive Assumption:

$$1+a = a+1.$$

Inductive Step.

$$\begin{aligned}1+a' &\overset{\text{by (5)}}{=} (1+a)' \\ &\overset{\substack{\text{by inductive}\\ \text{assumption}}}{=} (a+1)' \\ &\overset{\text{by (4)}}{=} (a')' \\ &\overset{\text{by (4)}}{=} a'+1.\end{aligned}$$

□

THEOREM 4.3. *Assume that + is a binary operation which satisfies conditions* (4) *and* (5). *Then + is commutative, that is,*

$$a+b = b+a$$

for all a and b.

Proof. We shall prove the theorem by induction on b.
Basis of Induction: Theorem 4.2.
Inductive Assumption:

$$a + b = b + a.$$

Inductive Step.

$$\begin{aligned}
a + b' &\overset{\text{by (5)}}{=} (a+b)' \\
&\overset{\substack{\text{by inductive}\\ \text{assumption}}}{=} (b+a)' \\
&\overset{\text{by (4)}}{=} (b+a)+1 \\
&\overset{\text{by Theorem 4.1}}{=} b+(a+1) \\
&\overset{\text{by Theorem 4.2}}{=} b+(1+a) \\
&\overset{\text{by Theorem 4.1}}{=} (b+1)+a \\
&\overset{\text{by (4)}}{=} b'+a.
\end{aligned}$$

□

5. Dark clouds

Notice that I was careful to formulate Theorems 4.1–4.3 in the most cautious way, by emphasising their conditional nature:

> *if $+$ is a binary operation which satisfies conditions* (4) *and* (5) *then ...*

The reason for my restraint is that writing down conditions (4) and (5) does not mean to define a function.

Another problem is that if you look at the proofs of Theorems 4.1–4.3, you notice that they do not refer to Axioms 3 and 4 and are based entirely on the Induction Axiom, Axiom 5. Therefore if we can exhibit a "toy version" of a system of natural numbers where we have a distinguished element 1, and the successor function S, and the Induction Axiom, but have no Axioms 3 and 4, we shall still should be able to define addition by conditions (4) and (5), and perhaps some other functions.

David Pierce [**22**] suggests to take for such "toy model" a system of residues $\mathbb{Z}/n\mathbb{Z}$ modulo n, with residue 1 in the role of the distinguished element, and with a successor function

David Pierce makes an incisive comment:

> Indeed, if one thinks that the recursive definitions of addition and multiplication—
>
> $$\begin{aligned} n+0 &= n, \\ n+(k+1) &= (n+k)+1; \\ n\cdot 0 &= 0, \\ n\cdot(k+1) &= n\cdot k+n \end{aligned}$$

> —are *obviously* justified by induction alone, then one may think the same for exponentation, with
>
> $$\begin{array}{rcl} n^0 & = & 1 \\ n^{k+1} & = & n^k \cdot n. \end{array}$$
>
> However, while addition and multiplication are well-defined on $\mathbb{Z}/n\mathbb{Z}$ (which admits induction), exponentiation is not; rather, we have
>
> $$\begin{array}{rcl} (x, y) & \mapsto & x^y \\ (\mathbb{Z}/n\mathbb{Z})^* \times \mathbb{Z}/\phi(n)\mathbb{Z} & \to & \mathbb{Z}/n\mathbb{Z}, \end{array}$$

where $(\mathbb{Z}/n\mathbb{Z})^*$, as usual, denotes the group of invertible elements of the residue ring $\mathbb{Z}/n\mathbb{Z}$.

Indeed, the recursive definition of exponentiation fails in $\mathbb{Z}/3\mathbb{Z}$:

n	n^2	n^3	$n^3 \times n$	n^4
2	1	2	1	2

but holds in $\mathbb{Z}/6\mathbb{Z}$:

n	n^2	n^3	n^4	n^5	n^6	n^7
1	1	1	1	1	1	1
2	4	2	4	2	4	2
3	3	3	3	3	3	3
4	4	4	4	4	4	4
5	1	5	1	5	1	5
6	6	6	6	6	6	6

The former is an exception rather than rule, as clarified by David Pierce's theorem.

THEOREM 5.1. (David Pierce [**22**]) *The identities*

$$a^1 = a, \qquad a^{b+1} = a^b \times a \tag{6}$$

hold on $\mathbb{Z}/n\mathbb{Z}$ if and only if $n \in \{0, 1, 2, 6\}$.

I share David Pierce's indignation at the state of affairs [**22**]:

> Yet the confusion continues to be made, even in textbooks intended for students of mathematics and computer science who ought to be able to understand the distinction. Textbooks also perpetuate related confusions, such as suggestions that induction and 'strong' induction (or else the 'well-ordering principle') are logically equivalent, and that either one is sufficient to axiomatize the natural numbers. [...]
>
> This is one example to suggest that getting things straight may make a pedagogical difference.

But I have to admit that I shared the widespread ignorance until David Pierce brought my attention to the issue—despite the fact that, in a calculus course that I took in the first year of my university studies, the lecturer (Gleb Pavlovich Akilov) explicitly proved the existence of a function of natural argument defined by a recursive scheme [**1**].

To save our theory from collapse, in the next section we shall prove the existence of addition.

Exercises.

Exercise 5.1. Prove Theorem 5.1 for prime values of n. You may wish to use Fermat's Theorem:

> If p is a prime integer and $0 < a < p <$ then
> $$a^p \equiv a \bmod p.$$

Exercise 5.2. Then try to prove Theorem 5.1 in full generality.

6. Landau's proof of the existence of addition

I decide to borrow *verbatim* a proof of the existence of addition from Edmund Landau's famous book *Grundlagen der Analysis* [**18**]. Also, I picked up from his book notation

$$S(n) = n'$$

which I had already used in my proofs. Although this is not emphasised by Landau, the proof of consistency of addition is not using Axioms 3 and 4. Are these axioms of any use at all? We shall return to this question later.

Theorem 6.1. [**18**, Theorem 4] *To every pair of numbers x, y, we may assign in exactly one way a natural number, called $x + y$, such that*

(1) $x + 1 = x'$ *for every* x,
(2) $x + y' = (x + y)'$ *for every* x *and every* y.

Proof. (A) First we will show that for each fixed x there is at most one possibility of defining $x + y$ for all y in such a way that $x + 1 = x'$ and $x + y' = (x + y)'$ for every y.

Let a_y and b_y be defined for all y and be such that

$$a_1 = x', \quad b_1 = x', \quad a_{y'} = (a_y)', \quad b_{y'} = (b_y)' \text{ for every } y.$$

Let $\mathcal{M}$ be the set of all y for which

$$a_y = b_y.$$

(I) $a_1 = x' = b_1$; hence 1 belongs to $\mathcal{M}$.
(II) If y belongs to $\mathcal{M}$, then $a_y = b_y$, hence by Axiom 2,

$$(a_y)' = (b_y)',$$

therefore

$$a_{y'} = (a_y)' = (b_y)' = b_{y'},$$

so that y' belongs to $\mathcal{M}$.

Hence $\mathcal{M}$ is the set of all natural numbers; i.e. for every y we have $a_y = b_y$.

(B) Now we will show that for each x it is actually possible to define $x + y$ for all y in such a way that

$$x + 1 = x' \text{ and } x + y' = (x + y)' \text{ for every } y.$$

Let $\mathcal{M}$ be the set of all x for which this is possible (in exactly one way, by (A)).
(I) For $x = 1$, the number $x + y = y'$ is as required, since
$x + 1 = 1' = x'$,
$x + y' = (y')' = (x + y)'$.
Hence 1 belongs to $\mathcal{M}$.

(II) Let x belong to $\mathcal{M}$, so that there exists an $x+y$ for all y. Then the number $x'+y=(x+y)'$ is the required number for x', since

$$x'+1=(x+1)'=(x')'$$

and

$$x'+y'=(x+y')'=((x+y)')'=(x'+y)'.$$

Hence x' belongs to $\mathcal{M}$. Therefore $\mathcal{M}$ contains all x. □

Landau's book is characterised by a specific austere beauty of entirely formal axiomatic development, dry, cut to the bone, streamlined. Not surprisingly, it is claimed that logical austerity and precision were Landau's characteristic personal traits.[6]

Grundlagen der Analysis opens with two prefaces, one intended for the student and the other for the teacher; we already quoted *Preface for the Teacher*, it is a remarkable pedagogical document. The preface for the student is very short and begins thus:

> 1. Please don't read the preface for the teacher.
> 2. I will ask of you only the ability to read English and to think logically-no high school mathematics, and certainly no higher mathematics. [...]
> 3. Please forget everything you have learned in school; for you haven't learned it.
>   Please keep in mind at all times the corresponding portions of your school curriculum; for you haven't actually forgotten them.
> 4. The multiplication table will not occur in this book, not even the theorem,
> $$2\times 2=4,$$
>   but I would recommend, as an exercise for Chap. I, section 4, that you define
> $$\begin{aligned}2 &= 1+1,\\ 4 &= (((1+1)+1)+1),\end{aligned}$$
>   and then prove the theorem.

Exercises.

Exercise 6.1. Follow Edmund Landau's advise and prove from the axioms of Peano arithmetic that

$$2\times 2=4.$$

7. Numbers in computer science

On of the contributors to my blog, a professional computer scientist, once left the following comment:

> *I would caution everyone ... not to confuse "mathematical thinking" with "The thinking done by computer scientists and programmers".*
>
> *Unfortunately, most people who are not computer scientists believe these two modes of thinking to be the same.*

[6] Asked for a testimony to the effect that Emmy Noether was a great woman mathematician, Landau famously said: "I can testify that she is a great mathematician, but that she is a woman, I cannot swear."

> *The purposes, nature, frequency and levels of abstraction commonly used in programming are very different from those in mathematics.*

This statement may appear to be extreme, but let us not to jump to conclusions and look first at a very simple example.

Later in my lectures I will show you some tricks with an ordinary hand held calculator, but in my examples her I suggest to have a look at its more advanced big brother, MATLAB, an industry standard software package for mathematical (mostly numerical) computations. The following fragment of text is a screen dump of me playing with natural numbers in MATLAB.

```
>> t= 2

t =      2

>> 1/t

ans =     0.5000
```

What you see here is a basic calculation which uses floating point arithmetic for computations with rounding; lines starting with `>>` are my input, unmarked lines are MATLAB's response.

Next, let us make the same calculation with a different kind of integers:

```
>> s=sym('2')

s =     2

>> 1/s

ans =   1/2
```

Here we use "symbolic integers", designed for use as coefficients in symbolic expressions. You can see that in the first example 1/2 was rounded as 0.5000, in the second case 1/2 is written as it is, as a fraction.

Since MATLAB keeps in its memory the values of the variables s and t, we may force it to combine the two kinds of integers in a single calculation:

```
>>  1/(s+t)

ans =   1/4
```

We observe that the sum $s+t$ of a floating point number t and a symbolic integer s is treated by MATLAB a symbolic integer.

Examples involving analytic functions are even more striking:

```
>> sqrt(t)

ans =       1.4142

>> sqrt(s)

ans =       2^(1/2)
```

```
>> sqrt(t)*sqrt(s)

ans =        2
```

We see that MATLAB can handle two absolutely different representations of integers, remembering, however, the intimate relation between them.

MATLAB is written in C++. When represented in C++, even the simplest mathematical objects and structures appear in the form of (a potentially infinite variety of) *classes* linked by mechanisms of *inheritance* and *polymorphism.* This is a manifestation of one of the paradigms of the computer science: if mathematicians instinctively seek to build their discipline around a small number of "canonical" structures, computer scientists frequently prefer to work with a host of similarly looking structures, each one adapted for a specific purpose. We shall look in the next lectures how they keep control of this bestiary. For a time being, we have only to take note that we have to be prepared to look at many different number systems satisfying the Dedekind-Peano axioms.

8. Counting sheep

The last observation is nothing new if we turn our attention from computer languages to the natural human lore: we already dealt with "named" numbers. But "named" numbers can come in a much more extreme form, as *numerals* used for counting specific types of objects (most likely, they historically precede the emergence of the universal number system as we know it). In England, a popular slander about Yorkshiremen is that they use special numerals for counting sheep. Judging by the Lakeland Dialect Society website [**23**], local people proudly admit to sticking to the old ways. In Wensleydale, for example, the first ten sheep numerals are said to be

yan, tean, tither, mither, pip, teaser, leaser, catra, horna, dick.

If we turn to more modern times, it is entertaining to compare sheep numerals with Richard Feynman's joke [**11**]:

> You see, the chemists have a complicated way of counting: instead of saying "one, two, three, four, five protons", they say, "hydrogen, helium, lithium, beryllium, boron."

This a joke but we have to learn some lessons from it.

One lesson is that we have to distinguish between *ordinal* numerals, which express relative order of objects,

first, second, third, . . .

and *cardinal* numerals which express the cardinality of a set, the number of elements:

one, two, three, . . .

To my eye, in Feynman's joke the words

hydrogen, helium, lithium,. . .

look more like ordinal numerals.

In languages around the world, there is a remarkable diversity of systems of numerals, both ordinal and cardinal.

The Japanese language provides one on more striking examples. Here, different numerals are used for counting, for example, flat objects (like sheets of paper) and long slender objects (like pencils). I give a table of some of them:

	regular numbers	simple objects	flat things	long slender things
1	ichi	hitotsu	ichimai	ippo
2	ni	futatsu	nimai	nihon
3	san	mittsu	sanmai	sanbon
4	shi or yon	yottsu	yonmai	yohon
5	go	itsutsu	gomai	gohon
6	roku	muttsu	rokumai	roppon
7	shichi or nana	nanatsu	nanamai	nanahon
8	hachi	yatsu	hachimai	happon
9	ku or kyu	kokonotsu	kyumai	kyuhon
10	ju or jyu	tou	jumai	jyuppon