Completed versus Incomplete Infinity in Arithmetic

by Edward Nelson

Department of Mathematics, Princeton University

The *numbers* are 0, 1, 2, 3, ... The numbers form the simplest infinity, so if we want to understand infinity we should try to understand the numbers.

Rather than use the symbols 1, 2, etc. originating in India, let us use the notation of mathematical logic: the numbers are 0, S0, SS0, SSS0, . . . where S can be read as "successor". This notation expresses clearly the idea that the numbers are obtained by counting, one after the other.

There are at least two different ways of looking at the numbers: as a completed infinity and as an incomplete infinity. We shall not be far wrong if we call these the *Platonic* (P) and the *Aristotelian* (A) ways.

Now already in this talk I have made a serious error due to a pitfall of language. I have spoken of P and A as two ways of looking at "the numbers". But we shall see that P and A give two *different* number systems, not two ways of looking at the same number system.

Let us begin with P, which is the viewpoint of contemporary mathematics. The numbers form a completed infinity denoted by $\mathbb{N}$. Let the variables x, y, and so forth, range over $\mathbb{N}$. The basic property of $\mathbb{N}$ is *induction*:

> *Hypotheses*:
> 0 has the property p;
> if x has the property p, then Sx has the property p.
> *Conclusion*:
> for all x, x has the property p.

The first hypothesis is the *basis* and the second is the *inductive step*. For a specific number, such as SSS0, we do not need to assume induction to prove the conclusion from the hypotheses. For suppose that $p(0)$. Then successively by the inductive step we obtain $p(\mathrm{S0})$, $p(\mathrm{SS0})$, and finally $p(\mathrm{SSS0})$. Nevertheless, induction is a powerful assumption and many properties of numbers can be proved only by using induction. Here is an example.

Let p be the following property of x: there exists a non-zero number y that is divisible by all non-zero numbers z such that $z \leq x$. I claim that every number x has the property p. Certainly 0 has the property p (let $y = \mathrm{S}0$). Suppose that x has the property p, so that there exists a non-zero number y' that is divisible by all non-zero numbers z with $z \le x$. By induction, we need only prove the inductive step, that $\mathrm{S}x$ has the property p; that is, that there exists a non-zero number y that is divisible by every non-zero number z with $z \le \mathrm{S}x$. But this is true: let $y = y' \cdot \mathrm{S}x$ and consider any non-zero number z with $z \le \mathrm{S}x$. Then either $z \le x$, in which case it divides y' and hence $y = y' \cdot \mathrm{S}x$, or $z = \mathrm{S}x$, in which case also it divides y, concluding the proof.

In the early years of the twentieth century Bertrand Russell and Henri Poincaré exchanged polemics about the nature of induction. For Poincaré, induction is a logical principle, a kind of infinitely long syllogism. Russell maintained that induction is merely a verbal definition—the numbers are *defined* to be those objects for which induction holds. Neither questioned the legitimacy of induction.

In contemporary mathematics Russell's viewpoint prevails. The customary foundation of mathematics today is set theory, Zermelo-Fraenkel set theory with the axiom of choice (ZFC). Properties are reified as sets. The number 0 is defined to be the empty set, the set that has no elements. For any set x, one defines its successor $\mathrm{S}x$ to be the set whose elements are the elements of x together with x itself. A set X is said to be *inductive* in case 0 is an element of X and whenever x is an element of X then its successor $\mathrm{S}x$ is also an element of X. The *axiom of infinity* of ZFC asserts that there exists an inductive set. Then one proves that there exists a unique smallest inductive set and one defines $\mathbb{N}$, the set of all numbers, to be this set.

Expressed verbally, what this amounts to is this. A property is *inductive* in case it satisfies the basis and the inductive step; an object is a *number* in case it has every inductive property. Thus a property that objects may have, that of being a number, is defined in terms of the collection of all properties that objects may have. This is an impredicative definition. Impredicativity deeply troubles some thinkers; others find it unproblematical. Contemporary mathematics is thoroughly impredicative.

As a foundation for arithmetic, this definition leaves something to be desired. Certainly arithmetic, the theory of numbers, is the most primitive of all mathematical theories. Yet it is being based on the far more sophisticated theory of sets. Furthermore, the procedure of reifying properties as sets—replacing a property, an intensional concept, by the extensional set of all objects having the property—leads to well-known contradictions if it is carried out naively, as Russell himself discovered in his famous paradox of the set of all sets that are not elements of themselves.

Closely related to induction is the construction of numbers by *primitive recursion*. This is done by specifying the value when $y = 0$ (the *basis*) and then the value for $\mathrm{S}y$ in terms of the value for y (the *recursive step*). Then, for example, the value when $y = \mathrm{SSS0}$ is determined: we are given the value for $y = 0$, from this we obtain the value when $y = \mathrm{S0}$, and then the value when $y = \mathrm{SS0}$, and finally the value when $y = \mathrm{SSS0}$. We introduce addition $+$, multiplication $\cdot$, exponentiation $\uparrow$, superexponentiation $\Uparrow$, and so forth, as follows:

$$\begin{array}{ll}
x + 0 = x, & x + \mathrm{S}y = \mathrm{S}(x + y); \\
x \cdot 0 = 0, & x \cdot \mathrm{S}y = x + (x \cdot y); \\
x \uparrow 0 = \mathrm{S0}, & x \uparrow \mathrm{S}y = x \cdot (x \uparrow y); \\
x \Uparrow 0 = \mathrm{S0}, & x \Uparrow \mathrm{S}y = x \uparrow (x \Uparrow y);
\end{array}$$

and so forth. Then

$$\begin{array}{ll}
x + y = \mathrm{S} \ldots \mathrm{S}x & \text{with } y \text{ occurrences of S}, \\
x \cdot y = x + \ldots + x & \text{with } y \text{ occurrences of } x, \\
x \uparrow y = x \cdot \ldots \cdot x & \text{with } y \text{ occurrences of } x, \\
x \Uparrow y = x \uparrow \ldots \uparrow x & \text{with } y \text{ occurrences of } x,
\end{array}$$

and so forth. These are primitive recursions. Ackermann showed how to go beyond primitive recursion. Let us denote $+$, $\cdot$, $\uparrow$, $\Uparrow$, and so forth, by F_0, F_1, F_2, F_3, and so forth. Then a version of the Ackermann function is the function A whose value on y is $A(y) = y\,\mathrm{F}_y\,y$. This function is recursive—for any y we have a mechanical procedure specified for computing $A(y)$—but it grows far faster than any primitive recursive function. This construction is another example of Cantor's diagonal method, which Cantor used to show that the continuum (the real numbers) is uncountable, a larger infinity than $\mathbb{N}$, and which was also the basis of Russell's paradox and Gödel's incompleteness theorems.

The general notion of a recursive function is intended to express the notion of an algorithm. The word *algorithm* comes from the name of the author, al'Khwarizmi, of the highly influential treatise *Al-Jabr wa-al-Muqabilah* written ca. 820 (and the word *algebra* itself comes from the title of the book!). But it was over eleven hundred years before the problem of defining what is meant by an algorithm—what a recursive function is—was addressed. Ackermann's procedure showed the impossibility of giving an explicit syntactical definition of the notion of a recursive function, for then they could all be listed and the diagonal construction employed to yield a new recursive function.

The problem was solved in three seemingly different ways in the 1930s by Church, Gödel, and Turing (respectively a professor at Princeton University, a permanent member of the Institute for Advanced Study in Princeton, and a graduate student at Princeton University, I mention with parochial pride), and the three definitions all turned out to be equivalent. Gödel himself said that Turing's was the best definition. Turing's work laid the theoretical foundation for computers and his paper [6] of 1936 still makes interesting reading, though one must be aware that in this paper *computer* means "one who computes". Turing's definition can be described as follows. Consider a computer program (this is a concrete syntactical object) taking numbers as input. Then it is an *algorithm* in case for every input it eventually halts and outputs a number as value (this is an abstract semantic concept). The halting problem is algorithmically unsolvable; this means that for a general computer program there is no way to tell whether or not it is an algorithm other than to search through all the infinitely many possible inputs and for each of them patiently to wait—forever if need be—to see whether a value is output. This is completed infinity with a vengeance! (In one of the Oz books the Scarecrow says, "We may be trapped here forever!" The Patchwork Girl asks, "How long is forever?" and the Scarecrow answers, "That is what we shall soon find out.")

* * *

Now let us turn to an Aristotelian (as I am calling it) critique of these ideas, regarding the numbers as an incomplete infinity. We may remark that etymologically "incomplete infinity" is a redundant phrase, since the very word *infinite* means unfinished.

As we have seen, induction and primitive recursion are based on the impredicative notion of the numbers as a completed infinity. Let us introduce the notion of a *counting number*. This is a primitive notion of A-arithmetic, and rather than attempt to define it we state the axioms that we assume for it. These are the basis and the inductive step, namely

0 is a counting number;
if y is a counting number, so is $\mathrm{S}y$.

This is all that we assume about the notion, and in particular we do not postulate that all numbers are counting numbers.

Using Arabic numerals (so called although they originated in India and were transmitted to the West largely by the Persian al'Khwarizmi), let us ask, given a specific number y defined by primitive recursion, say $y = 2 \Uparrow 5$ or $y = 2 \Uparrow 2 \Uparrow 5$, whether we can prove that it is a counting number. Now to say that there is an obvious proof in y steps is circular, because steps are things that are counted, so we can only count the steps if y is indeed a counting number. Notice that $2 \Uparrow 5 = 2 \uparrow 2 \uparrow 2 \uparrow 2 \uparrow 2 = 2 \uparrow 65536$ is a super-astronomically large number, and $2 \Uparrow 2 \Uparrow 5$ is $2 \uparrow \ldots \uparrow 2$ with that number, $2 \Uparrow 5$, of occurrences of 2.

We make the following two definitions:

1. x is an *additive number* in case for all counting numbers y, $x + y$ is also a counting number;
2. x is a *multiplicative number* in case for all additive numbers y, $x \cdot y$ is also an additive number.

Then the following theorems hold:

3. If x is an additive number, then x is a counting number.
4. If x is a multiplicative number, then x is an additive number.
5. If x is a multiplicative number, then x is a counting number.
6. If x and z are additive numbers, so is $x + z$.
7. If x and z are multiplicative numbers, so is $x + z$.
8. If x and z are multiplicative numbers, so is $x \cdot z$.

The proofs are easy. For 3, let x be an additive number. Apply the definition 1 to $y = 0$, which is a counting number. Then $x + 0$ is a counting number, but $x + 0 = x$.

For 4, let x be a multiplicative number. Apply the definition 2 to $y = 1$, which is easily seen to be an additive number. Then $x \cdot 1$ is an additive number, but $x \cdot 1 = x$.

Note that 5 is a consequence of 4 and 3.

For 6, let x and z be additive numbers, and let y be any counting number. By definition 1, we need to show that $(x + z) + y$ is a counting number. Now $z+y$ is a counting number, by definition 1, so $x+(z+y)$ is a counting number, again by definition 1. But $x + (z + y) = (x + z) + y$.

For 7, let x and z be multiplicative numbers, and let y be any additive number. By definition 2, we need to show that $(x + z) \cdot y$ is an additive number. Now $z + y$ is an additive number, by definition 2, and $x + z$ is an additive number, also by definition 2. Hence $(x \cdot y) + (z \cdot y)$ is an additive number by theorem 6. But $(x \cdot y) + (z \cdot y) = (x + z) \cdot y$.

For 8, let x and z be multiplicative numbers, and let y be any additive number. By definition 2, we need to show that $(x \cdot z) \cdot y$ is an additive number. Now $z \cdot y$ is an additive number, by definition 2, so $x \cdot (z \cdot y)$ is an additive number, also by definition 2. But $x \cdot (z \cdot y) = (x \cdot z) \cdot y$.

Now we can prove that $2 \Uparrow 5$ is a counting number. Let

$$a_0 = 2,\ a_1 = a_0 \cdot a_0,\ a_2 = a_1 \cdot a_1,\ \ldots,\ a_{16} = a_{15} \cdot a_{15}.$$

Then $a_{16} = 2 \Uparrow 5$. It is easily seen that 2 is a multiplicative number. Applying theorem 8 sixteen times, we see that $a_1, a_2, \ldots$, and $a_{16} = 2 \Uparrow 5$ are multiplicative numbers, so $2 \Uparrow 5$ is a counting number by theorem 5, concluding the proof. But no one will ever prove that $2 \Uparrow 2 \Uparrow 5$ is a counting number.

Why can't we continue the sequence of definitions and theorems? Suppose we define

> 9. x is an *exponentiable number* in case for all multiplicative numbers y, $x \uparrow y$ is a multiplicative number.

But then we cannot prove

> 10. If x and z are exponentiable numbers, so is $x \uparrow z$.

The problem is that exponentiation is not associative, for in general $x \uparrow (z \uparrow y) \neq (x \uparrow z) \uparrow y$, and we used the associativity of addition and multiplication to prove theorems 6 and 8. In fact, one can prove the following: there does not exist a property p for which it is possible to prove that if $p(x)$ then x is a counting number and that if $p(x)$ and $p(z)$ then $p(x \uparrow z)$. This theorem is not easy; it uses the deep theorem of Hilbert and Ackermann on quantifier elimination. See Chapter 18 of the author's *Predicative Arithmetic* [4].

In short, from an A point of view, exponentiation of numbers is not a well-defined concept. Consequently, A-mathematics is *much* weaker than P-mathematics. For example, the theorem on divisibility that we proved by induction cannot be established. Nevertheless, A-mathematics is worth pursuing for several reasons. One is that a truly surprising amount of advanced mathematics can be developed from this point of view with great simplification of the technical tools involved; see the author's *Radically Elementary Probability Theory* [5].

Another reason for developing A-mathematics is its connection with problems of computational complexity, one of the most active fields of mathematics and theoretical computer science. We examine this connection now.

Turing, Church, and Gödel answered—provided one accepts $\mathbb{N}$ as a completed infinity—the question "what is an algorithm?". With the advent of digital computers the question arose, "what is a *feasible* algorithm?". A consensus among students of computational complexity emerged, that the right definition is a *polynomial-time function.* These are the functions such that there exist a Turing machine (computer program) and a polynomial π such that for any number y the program halts and yields a value after at most $\pi(\log y)$ steps. This is a complicated definition that at first sight seems rather arbitrary. But Bellantoni and Cook [1] [2], and also Leivant [3], gave an equivalent definition, which can be described as follows.

Consider again primitive recursion. We want to define $F(x, y)$, so we specify a value when $y = 0$ and then specify a value for $F(x, \mathrm{S}y)$ in terms of x and the value for $F(x, y)$:

$$F(x, 0) = G(x), \quad F(x, \mathrm{S}y) = H\big(x, F(x, y)\big).$$

Here G and H are given in terms of 0, S, and previously defined functions. Then we have

$$\begin{aligned} F(x, 0) &= G(x), \\ F(x, \mathrm{S}0) &= H\big(x, F(x, 0)\big) = H\big(x, G(x)\big), \\ F(x, \mathrm{SS}0) &= H\big(x, F(x, \mathrm{S}0)\big) = H\big(x, H\big(x, G(x)\big)\big), \\ F(x, \mathrm{SSS}0) &= H\big(x, F(x, \mathrm{SS}0)\big) = H\big(x, H\big(x, H\big(x, G(x)\big)\big)\big), \end{aligned}$$

and so forth all the way up to the value of $F(x, y)$.

For this to make sense, y must be a counting number, since the definition is a step-by-step construction from 0 to S0, to SS0, ..., and finally to y, but even if y is a counting number *we do not know that the value $F(x, y)$ is itself a counting number* (unless we make the Platonic postulate that all numbers are counting numbers). A *predicative recursion* is a primitive recursion in which all the recursions are over counting numbers only.

Let us see how this works. There is no problem with addition:

$$x + 0 = x, \quad x + \mathrm{S}y = \mathrm{S}(x + y).$$

This makes sense for any number x and any counting number y. Now suppose that both x and y are counting numbers. Then there is no problem with multiplication:

$$x \cdot 0 = 0, \quad x \cdot \mathrm{S}y = (x \cdot y) + x$$

since we have predicatively defined the sum of any number, such as $x \cdot y$, and any counting number, such as x. These are examples of predicative recursion. But exponentiation is impredicative. In the construction

$$x \uparrow 0 = \mathrm{S}0, \quad x \uparrow \mathrm{S}y = x \cdot (x \uparrow y)$$

the number $x \uparrow y$ is not known to be a counting number—but the predicative recursion giving multiplication is predicatively defined only when the second argument is a counting number. And indeed, exponentiation is infeasible.

Bellantoni and Cook, and Leivant, prove that a function is a polynomial-time function (a feasible algorithm) if and only if is is constructed by predicative recursion. This is a simple and beautiful characterization: there is nothing concerning Turing machines or polynomials in this description, but it is equivalent to the complicated definition given earlier.

* * *

In conclusion, regarding the numbers as an incomplete infinity offers a viable and interesting alternative to regarding the numbers as a completed infinity, one that leads to great simplifications in some areas of mathematics and that has strong connections with problems of computational complexity.

The two ways, P and A, of regarding numbers lead to different number systems. What is a finite number for P is not necessarily a finite number for A. In contemporary mathematics, the notion of finite is defined in terms of the completed infinity $\mathbb{N}$. There is no clear concept of the finite in terms of which the infinite can be defined as not-finite. One goes in the opposite direction in contemporary, Platonic, mathematics and defines the finite as not-infinite.

Perhaps we should hold an interdisciplinary conference on the finite. As Horatio said, "There are fewer things in heaven and earth, Hamlet, than are dreamt of in your philosophy."

References

[1] Stephen Bellantoni and Stephen Cook, "A new recursion-theoretic characterization of the poly-time functions", *Computational Complexity*, 2:97–110, 1992.

[2] Stephen Bellantoni, *Predicative Recursion and Computational Complexity*, Ph.D. Thesis, University of Toronto, 1992.
ftp://ftp.cs.toronto.edu/pub/reports/theory/cs-92-264.ps.Z

[3] Daniel Leivant, "Ramified recurrence and computational complexity I: Word recurrence and poly-time", in Peter Cole and Jeffrey Remmel, editors, *Feasible Mathematics II*, Perspectives in Computer Science, pages 320-343, Birkhauser-Boston, New York, 1994.

[4] Edward Nelson, *Predicative Arithmetic*, Mathematical Notes #32, Princeton University Press, Princeton, New Jersey, 1986.

http://www.math.princeton.edu/~nelson/books/pa.pdf

[5] —, *Radically Elementary Probability Theory*, Annals of Mathematics Studies, Number 117, Princeton University Press, Princeton, New Jersey, 1987.
http://www.math.princeton.edu/~nelson/books/rept.pdf

[6] A. M. Turing, "On computable numbers, with an application to the Entscheidungsproblem", Proceedings of the London Mathematical Society, Series 2, 42, pp. 230–265, 1936. Errata in 43, pp. 544–546, 1937.
http://www.abelard.org/turpap2/tp2-ie.asp