Natural Numbers

Cezar Ionescu

WS 2023-24

Preliminary Remarks

In mathematics, **definitions** serve two related purposes:

- a **checklist** of features that must be fulfilled in order for an object to qualify as an instance of the defined concept
- a **guarantee** of the features that an object has if it is an instance of the defined concept.

A **theorem** usually states:

- that a given collection of objects with some assumed features is also guaranteed to have some other features
  - e.g.: a differentiable function must also be continuous
- that given a collection of objects with some assumed features, there must also exist one or more other objects that are guaranteed to have some stated features
  - e.g.: a bounded sequence contains a convergent subsequence
- that two *expressions* denote the same mathematical object
  - e.g.: $x + y = y + x$

A **proof** will correspondingly make use of the features that are guaranteed to be fulfilled by the given objects. Especially important for computer scientists are **proofs** of the existence of objects by means of *algorithmically constructing* such an object. In the example above, we do *not* have a general method for discovering the convergent subsequence, but we can construct one if we assume more about the original bounded sequence.

The Natural Numbers

Consider the following definition of natural numbers (Earl 2017)

Definition 1. *A **natural number** is a non-negative whole number. The set of natural numbers is denoted by $\mathbb{N}$. Note that by this definition 0 is a natural number; some texts consider only positive whole numbers to be natural.*

The idea is clear: $\mathbb{N} = \{0, 1, 2, \ldots\}$. Natural numbers are just about the simplest mathematical objects and the reader will certainly not have difficulties identifying a natural number from

something that is not a natural number (with the necessary additional remark about 0). Still, there are some flaws in this definition:

- The notion of *whole number* is assumed. But whole numbers are usually introduced *after* natural numbers, not before.
- It is clear that we identify a natural number, but it is not clear what sort of *guarantees* natural numbers bring with them.

For example: we know how to add and multiply numbers, but it is not clear how to define these operations, e.g., if we had to implement them "from scratch".

The most elementary feature of natural numbers (and the one we learn first) is that they can be used for *counting*: every natural number has a *successor*. A more rigorous definition is then

Definition 2. *The set of natural numbers, denoted by $\mathbb{N}$, is defined by the following conditions for membership in $\mathbb{N}$:*

1. $O \in \mathbb{N}$
2. *if* $n \in \mathbb{N}$, *then* $S(n) \in \mathbb{N}$
3. *nothing is in $\mathbb{N}$ unless it follows from these two clauses.*

This is an example of **inductive definition**.

Here S, the successor function, is a purely *typographical operation*. The elements of $\mathbb{N}$ are thus $\{O, S(O), S(S(O)), S(S(S(O))), \ldots\}$.

According to this definition, our "normal" natural numbers are a different way of denoting the "true" natural numbers.

Some programming languages, such as Idris and Agda, use precisely this definition for the natural numbers. The Agda version, for example, is

```
data Nat : Set where
  zero  :         Nat
  suc   : Nat ->  Nat
```

Using the operation S, we can define all other operations on natural numbers. For example, addition:

$$+ : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$$

$$O + m = m \tag{1}$$
$$S(n) + m = S(n + m) \tag{2}$$

Example:

$$
\begin{aligned}
& S(S(O)) + S(O) \\
= \quad & \{\text{second eqn for } +\} \\
& S(S(O) + S(O)) \\
= \quad & \{\text{second eqn for } +\} \\
& S(S(O + S(O))) \\
= \quad & \{\text{first eqn for } +\} \\
& S(S(S(O)))
\end{aligned}
$$

We have *proven* that $2 + 1 = 3$ by a **calculational proof**. Calculational proofs are very common in computer science, because they involve the *evaluation of expressions* by means of definitions. The process of evaluation of expressions is the primary mechanism for evaluating the time and space complexity of algorithms.

The format for displaying a calculation proof is attributed to Wim Feijen. The earliest book I found that uses this format throughout is *A Method of Programming*, Dijkstra and Feijen (Dutch 1984, German 1985, English 1988).

We can similary evaluate $S(O) + S(S(O))$ and we'll reach the same result. This shows that $S(S(O))+S(O) = S(O)+S(S(O))$. We can prove for any given $m, n \in \mathbb{N}$ that $m+n = n+m$, but how can we prove it for *all* of them?

Induction on Natural Numbers

The answer is the **principle of induction**: to prove that a property P holds for every natural number, it suffices to show that

1. $P(O)$ holds, and
2. if $P(n)$ holds for some $n \in \mathbb{N}$, then $P(S(n))$ also holds.

To show how this works, we prove another familiar property of addition: *associativity*.

Proposition 1. *For all* $x, y, z \in \mathbb{N}$, $x + (y + z) = (x + y) + z$

Proof. We want to prove the property

$$P(x) \quad \equiv \quad \forall y, z \in \mathbb{N} \bullet (x + (y + z) = (x + y) + z$$

- Base case: $P(O)$

$$
\begin{aligned}
& O + (y + z) \\
= \quad & \{\text{fst. eqn. for } +\} \\
& y + z
\end{aligned}
$$

$$
\begin{aligned}
& (O + y) + z \\
= \quad & \{\text{fst. eqn. for } +\} \\
& y + z
\end{aligned}
$$

Therefore, $P(O)$ holds.

- Induction step: Assume $P(x)$ and show $P(S(x))$.

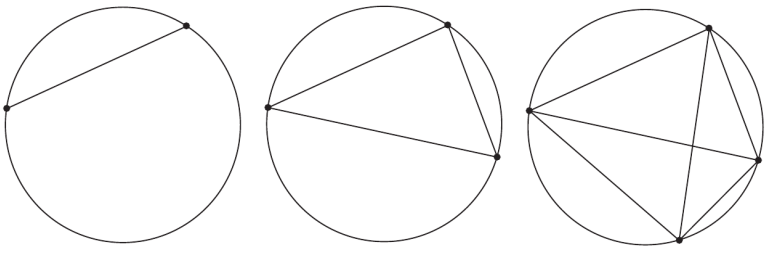
$$
\begin{aligned}
& S(x) + (y + z) \\
= \quad & \{\text{snd. eqn. for } +\} \\
& S(x + (y + z)) \\
= \quad & \{P(x)\} \\
& S((x + y) + z) \\
= \quad & \{\text{snd. eqn. for } +\} \\
& S(x + y) + z \\
= \quad & \{\text{snd. eqn. for } +\} \\
& (S(x) + y) + z
\end{aligned}
$$

This establishes $P(S(x))$

Therefore, $P(x)$ holds for all $x \in \mathbb{N}$. □

The following example (cited from Earl 2017, page 71) shows why proofs are necessary.

> To reinforce the need for proof, and to show how patterns can at first glance deceive us, consider the following example. Take two points on the circumference of a circle and take a line joining them; this line then divides the circle's interior into two regions. If we take three points on the circumference then the lines joining them will divide the disc into four regions. Four points can result in a maximum of eight regions (see figure below). Surely, then, we can confidently predict that n points will maximally result in 2^{n-1} regions.



$n = 2$ $n = 3$ $n = 4$

Further investigation shows our conjecture to be true for $n = 5$ but, to our surprise, when we take six points on the circle, the maximum number of regions attained is 31, no matter how we choose the points. Indeed the maximum number of regions attained from n points on the circumference is given by the formula

$$\frac{1}{24}(n^4 - 6n^3 + 23n^2 - 18n + 24)$$

Our original guess was way out! This is known as *Moser's circle problem*.

Other forms of Induction

In the following, we shall use the standard notation for natural numbers (0 for O and $n+1$ for $S(n)$).

Proposition 2. *(**Induction starting from** N) Let $N \in \mathbb{N}$ and $P(n)$ be a family of statements for $n \geq N$. Suppose that*

- *$P(N)$ is true*
- *for any $n \geq N$, if $P(n)$ is true, then $P(n+1)$ is true.*

Theorem 1. *(**Strong form of induction**) Let $P(n)$ be a family of statements for $n \in \mathbb{N}$. Suppose that*

- *$P(0)$ is true*
- *for any n, if $P(0)$, $P(1)$, ..., $P(n)$ are all true, then so is $P(n+1)$.*

Then $P(n)$ is true for all n.

Example 1. *Show that every $n \in \mathbb{N}_{\geq 1}$ can be written as $2^k \times l$ where $k, l \in \mathbb{N}$ and l is odd.*

Solution. We'll use the strong form of induction starting from 1.

- Base case: $P(1)$ holds because $1 = 2^0 \times 1$.
- Induction step: We assume $P(1)$, $P(2)$, ..., $P(n)$. Then:
  - if $n+1$ is odd, $P(n+1)$ holds because $n + 1 = 2^0 \times (n+1)$.
  - if $n+1$ is even, then $n + 1 = 2 \times m$ where $1 \leq m \leq n$.

$$
\begin{aligned}
& n+1 \\
= \quad & \{n \text{ even}\} \\
& 2 \times m \\
= \quad & \{P(m)\} \\
& 2 \times 2^k \times l \\
= \quad & \{\text{arithmetic}\} \\
& 2^{k+1} \times l
\end{aligned}
$$

Therefore, $P(n+1)$ also holds.

□

Natural Numbers as Sets

We have explained why the principle of induction holds, but we haven't really given a rigorous proof of it. In fact, the principle of induction is part of our *specification* of natural numbers. To summarize our discussion so far, the natural numbers are characterized (specified) by the following properties, known as *Peano's axioms*:

1. 0 is a natural number

ii) If n is a natural number, then so is its successor $S(n)$ iii) Different natural numbers have different successors (conversely, if two natural numbers have the same successors, then they are equal). iv) 0 is not the successor of any natural number.

1. The principle of induction: If P is a property indexed by natural numbers (i.e., for each natural number n, $P(n)$ is a proposition), then if $P(0)$ is true, and if $P(n)$ implies $P(S(n))$ for every n, then $P(n)$ is true of all natural numbers.

How can we be sure that these requirements can be satisfied? This question seems to require a proof. A proof requires that we have certain assumptions which we can take for granted, and we show that the conclusion follows from those assumptions. If those assumptions are harder to understand or to believe than the conclusion, the proof seems relatively useless.

One set of assumptions which is accepted by most mathematicians today is that of *set theory*. We have seen in a previous lecture that we can represent or *implement* the natural numbers as sets. Using the axioms of set theory, we can show that either of the representations we have seen does fulfill the requirements set above. Therefore, if we believe the axioms of set theory, then we can accept that the conditions above are *consistent*, that is, that they can be simultaneously satisfied. As long as we do not ask anything more of natural numbers, then we also do not care which particular implementation of natural numbers is used. In this way, set theory is used as a *programming language*: we use it to show how certain specifications may be implemented.

There are other approaches as well. For example, we assume that the Peano axioms are satisfied by our 0 and $S(n)$ objects, as we have in fact done until this section. After all, although the principle of induction looks somewhat complicated, it does have an intuitive explanation, and is not at first sight less trustworthy than the axioms of set theory. In this approach, arithmetic with natural numbers is more elementary than set theory, and we use it to implement the other, more complex, mathematical objects. For example, we can encode pairs of natural numbers as natural numbers: (p, q) could be represented by, say, p^2q^3. With pairs of natural numbers, we can represent integers. Statments about integers become then statements about certain kinds of natural number.

It turns out that we do need some sets in order to be able to work with these constructions, but the sets are of a special and very limited kind. The sets used are called *types*. Every type is a (standard) set, but very few standard sets are actually types. Types are an important topic in computer science, in particular in the are of programming languages.

The Binomial Theorem

Definition 3. *Let $n, k \in \mathbb{N}$, $k \leq n$. The (n, k) th **binomial coefficient** is the number*

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Alternative notation: C_k^n.

Remark 1. *We shall also use the $\binom{n}{k}$ notation for the cases in which $k > n$ or $k \in \mathbb{Z}_{<0}$. In these cases, $\binom{n}{k} = 0$.*

Proposition 3. *Basic identities:*

- $\binom{n}{k} = \binom{n}{n-k}$
- $\binom{n}{0} = \binom{n}{n} = 1$
- $\binom{n}{1} = \binom{n}{n-1} = n$

Proposition 4. *Let $1 \leq k \leq n$. Then*

$$\binom{n}{k-1} + \binom{n}{k} = \binom{n+1}{k}$$

Remark 2. *This result shows that the binomial coefficients are the entries in Pascal's triangle:*

$$\begin{array}{lccccccccccc}
n=0 & & & & & & 1 & & & & & \\
n=1 & & & & & 1 & & 1 & & & & \\
n=2 & & & & 1 & & 2 & & 1 & & & \\
n=3 & & & 1 & & 3 & & 3 & & 1 & & \\
n=4 & & 1 & & 4 & & 6 & & 4 & & 1 & \\
n=5 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 \\
n=6 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1
\end{array}$$

Theorem 2. ***(Binomial Theorem)*** *Let $n \in \mathbb{N}$ and $x, y \in \mathbb{R}$. Then*

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

Proposition 5. *Let $n \in \mathbb{N}$. Then:*

- $\binom{n}{0} + \binom{n}{1} + \binom{n}{2} + \ldots + \binom{n}{n} = 2^n$
- $\binom{n}{0} + \binom{n}{2} + \binom{n}{4} + \ldots = 2^{n-1}$
- $\binom{n}{1} + \binom{n}{3} + \binom{n}{5} + \ldots = 2^{n-1}$

References

- *Towards Higher Mathematics*, Richard Earl, Cambridge University Press 2017