MFCS

Dedekind-Peano Axioms

Klaus Sutner
Carnegie Mellon University
Fall 2022

1 **Dedekind-Peano Axioms**

2 **Self-Similarity, Induction, Recursion**

3 *** Truth and Proofs**

Dedekind-Peano

Richard Dedekind and Guiseppe Peano gave an axiomatization of the natural numbers in the late 1800s.

- 0 is a natural number.
- The successor of a natural number is a natural number.
- 0 is not a successor.
- Two different natural numbers have different successors.
- Principle of Mathematical Induction
  If a property obtains at 0 and is inherited by the successor of every natural number with this property, then this property holds of all natural numbers.

This is the minimalist version, usually one includes axioms for addition, multiplication and order.

Formalizing Dedekind-Peano

successor[†]

$$S(x) \neq 0 \qquad\qquad S(x) = S(y) \Rightarrow x = y$$

addition

$$x + 0 = x \qquad\qquad x + S(y) = S(x + y)$$

multiplication

$$x \cdot 0 = 0 \qquad\qquad x \cdot S(y) = (x \cdot y) + x$$

order

$$\neg(x < 0) \qquad\qquad x < S(y) \Leftrightarrow x = y \vee x < y$$

[†]We have left off the pesky universal quantifiers.

Infinity

These axioms use basic symbols 0, S, $+$, $\cdot$ and $<$, all with their obvious meaning.

The successor/addition/multiplication axioms are all due to Richard Dedekind, and are really incredibly clever.

The successor axioms imply that there are infinitely many natural numbers: S is injective but not surjective (0 is not in its range). This is a brilliant way to describe infinite sets in FOL.

The axioms controlling addition, multiplication and order are all based on recursion. They were proposed about half a century before computability theory came along, and another 30 years before programming languages supporting recursion existed.

Induction Axiom

So far, the axioms are quite feeble, we need a way to express induction. So let P be any property of the naturals and adopt the induction axiom

$$P(0) \wedge \forall\, z \left(P(z) \Rightarrow P(S(z))\right) \Rightarrow \forall\, z\, P(z)$$

Aside: This is a white lie: as stated, this is not first-order, we would need to quantify over P which requires a second-order system.

$$\forall\, P \left(P(0) \wedge \forall\, z \left(P(z) \Rightarrow P(S(z))\right) \Rightarrow \forall\, z\, P(z) \right)$$

Second-order logic is a hot mess (and really just set theory in disguise).

Standard Workaround

We want to keep everything at the first-order level, where life is enjoyable, gratifying and pleasant. So we use a trick: we replace the predicate $P(z)$ by an arbitrary first-order formula $\psi(z)$ (with free variable z).

$$\psi(0) \wedge \forall\, z \left(\psi(z) \Rightarrow \psi(S(z))\right) \Rightarrow \forall\, z\, \psi(z)$$

This is a so-called axiom schema, we get one axiom for each choice of ψ. So there are infinitely many induction axioms, but that's not much of a problem at all.

The idea that one can replace vaguely defined notions like "definite property" by first-order formulae is now over 100 years old and has conquered the world.

Success!!!

So now we have a pretty axiom system that describes

> successor, addition, multiplication, order and induction.

It has been empirically verified over the last 125 years that these axioms are enough to derive all the basic properties of the natural numbers.

For example, I strongly suspect that all the homework assignments in MFCS that do not deal with stuff like the reals could be formally proven using just the Dedekind-Peano axioms[†]

[†]You would go stark raving mad if you tried to really carry out this project. Maybe you could do it using the latest LEAN libraries, I'm not sure.

Self-Similarity

Here is another way to think about induction. We are interested in

> Objects that are similar to parts of themselves.

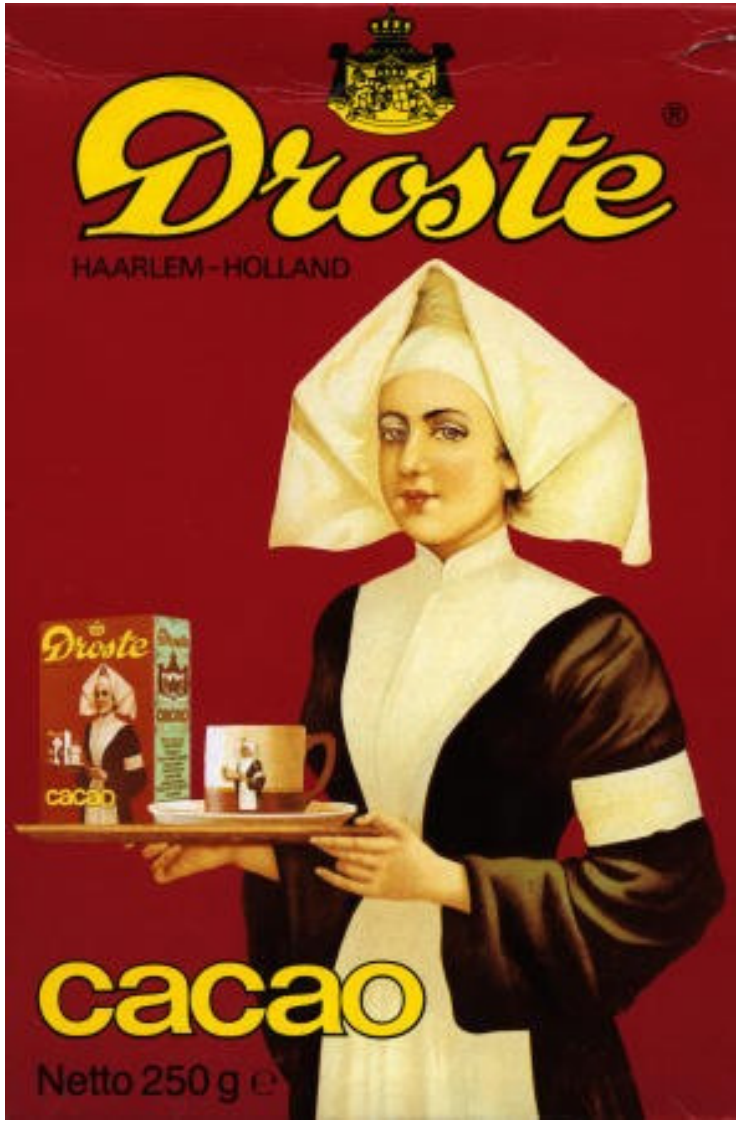
So these objects are somehow composed of simpler, smaller, yet similar objects. Since the smaller components are similar to the large object, they in turn can be decomposed into yet smaller ones, and so on.

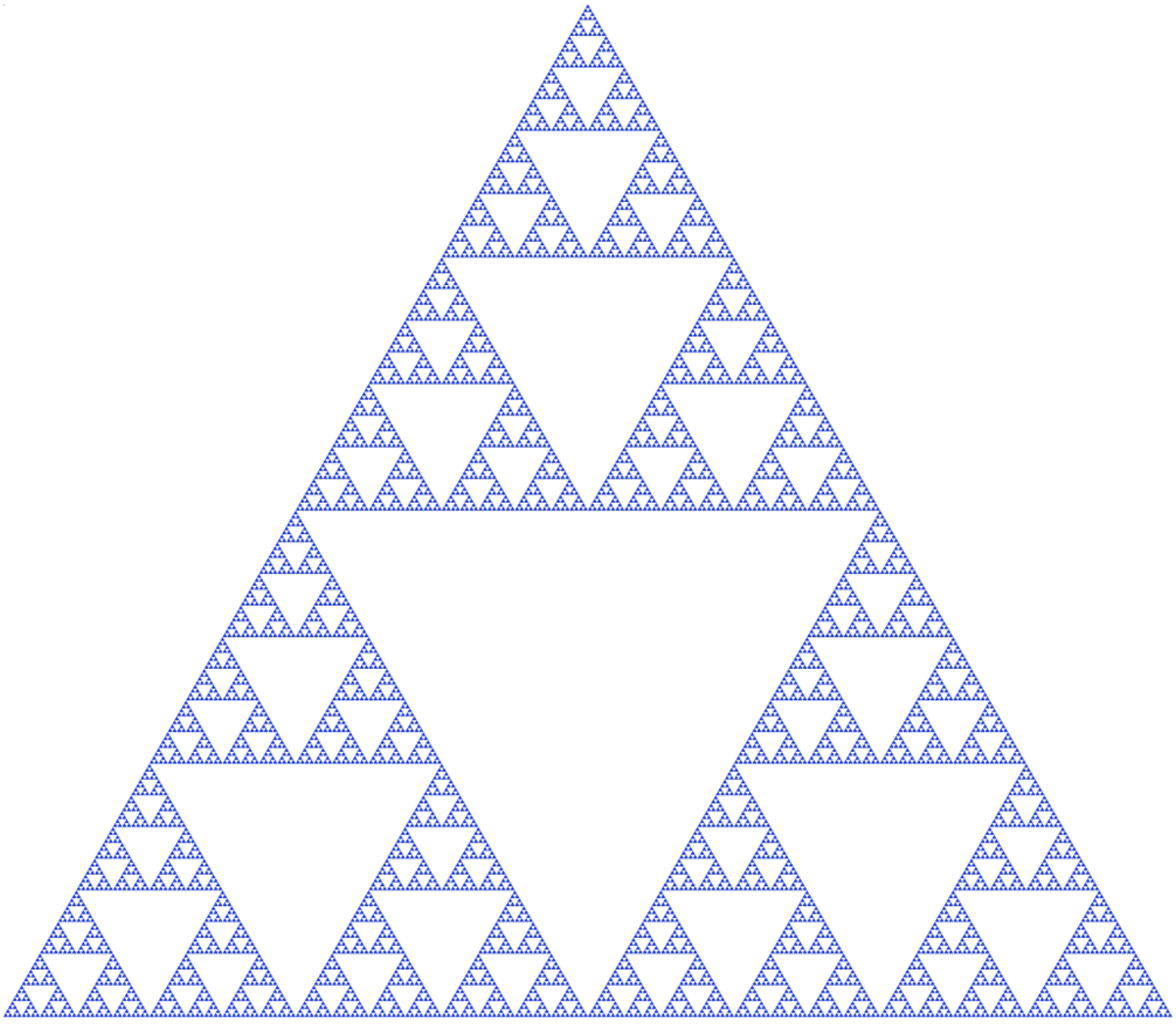
There are two scenarios:

- the decomposition goes on forever,
- after finitely many steps, the decomposition reaches indecomposable atoms, and stops.

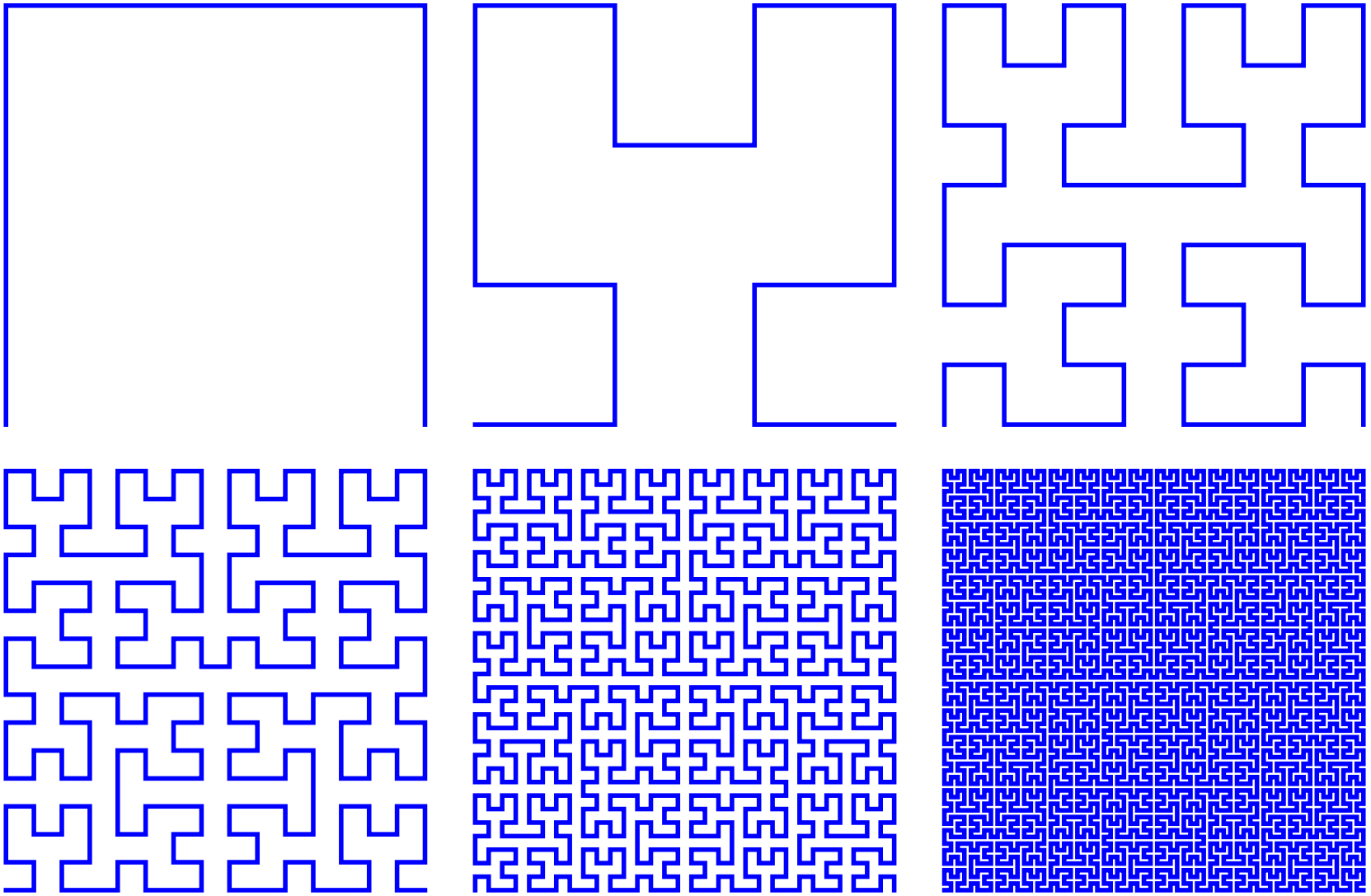
For computational purposes, the finite descent case is much more important, but in geometry the infinite version is easier to understand.

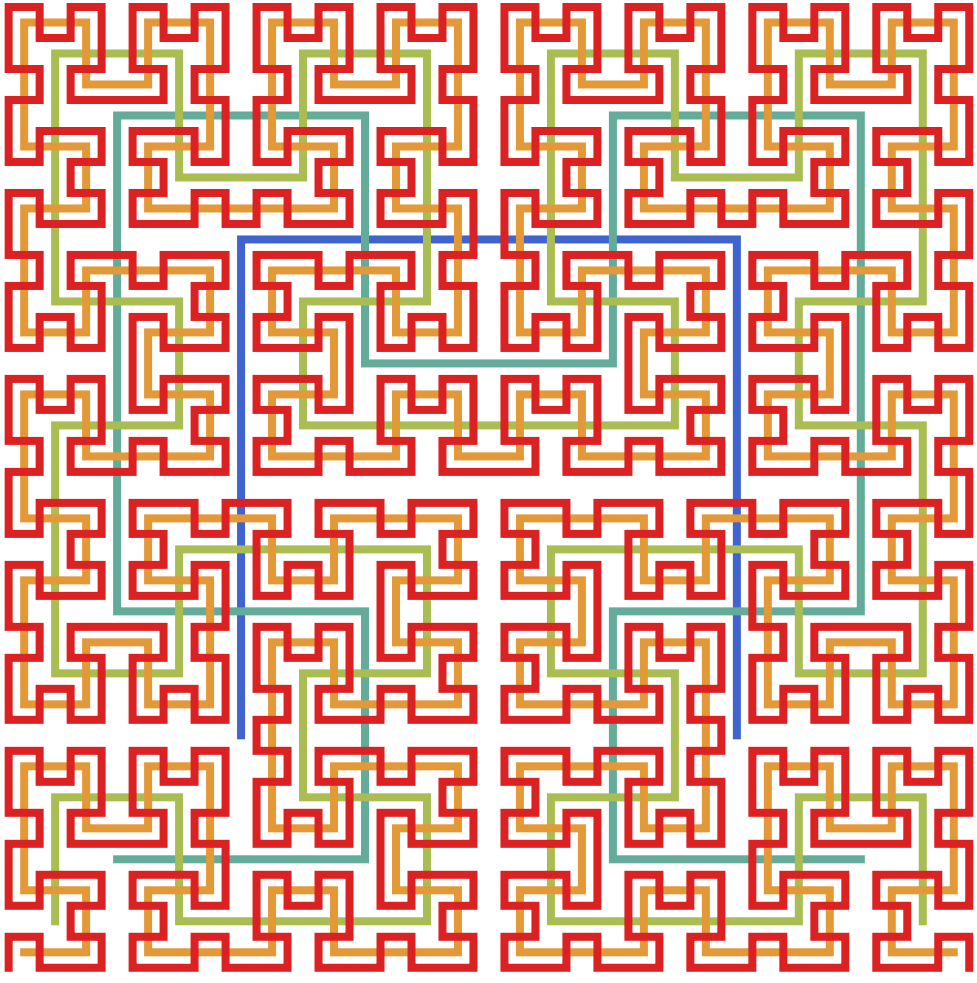
Droste Effect





Hilbert





A Miracle

Hilbert recursively defines a family of curves, maps from the unit interval to the unit square.

$$H_n : [0,1] \to [0,1]^2$$

The Hilbert curve is the limit of these guys:

$$H : [0,1] \to [0,1]^2 \qquad H(r) = \lim H_n(r)$$

Lo and behold, H is a continuous bijection between $[0,1]$ and $[0,1]^2$.

This is why we need proofs.
Intuition only goes so far.

Top Down: Recursion

One way to exploit self-similarity is to use it to define operations: we explain the result of the operation on a big object in terms of the results on smaller objects, until we hit rock-bottom.

Tired old standard example:

$$f(0) = 1$$
$$f(x) = x\, f(x{-}1)$$

This is one way to define the factorial function. An arguable less obfuscatory definition is

$$f(x) = \prod_{i=1}^{x} i$$

Bottom Up: Induction

Alternatively, we can start at the small objects and then work our way up to larger ones, exploiting the fact that we already have the results for the smaller guys.

For example, in recursion, the computation of $50!$ contains a similar sub-computation for $49!$, which contains a sub-computation for $48!$ and so on, down to $0!$ where the recursion stops.

By the same token, we understand the computation for $0!$ (the result is just 1) and can use it to build up the computation of $1!$, then $2!$, then $3!$ and so on, all the way up to $50!$.

Killer App

This machinery becomes really powerful when

- we construct some objects by induction, and then
- prove their properties, again by induction.

This idea is used in countless places in math and TCS, but even in plain applications like the design of data structures.

Induction versus Recursion

It is a convention to always talk about induction when it comes to proofs, no one ever says "proof by recursion."

But for constructions both terms are used, we can define a structure or an operation by induction or by recursion.

In the world of algorithms, recursion is often easier to express than induction: just write down the recursion, the system organizes the computation. Alas, it may be slower than a careful bottom-up approach (dynamic programming).

Natural Numbers

DP axioms S_1 and S_2 together with induction have the purpose to pin down the naturals as the following sequence:

$$\mathbb{N} = \{0, S(0), S(S(0)), S(S(S(0))), \ldots, S^k(0), \ldots\}$$

Correspondingly, to prove an assertion $\psi(z)$ for all $z \in \mathbb{N}$, it suffices to show that

- $\psi(0)$
- $\psi(z)$ implies $\psi(S(z))$

So we have to establish the base case, and show that the induction hypothesis (IH) implies induction conclusion (IC).

A Proof

To avoid confusion with intuitive addition, let's write

$$\mathsf{add}(x, 0) = x$$

$$\mathsf{add}(x, S(y)) = S(\mathsf{add}(x, y))$$

This is an addition algorithm of sorts, but not the kind you are familiar with: those are all based on representing numbers in a particular numeration system (usually decimal or binary), and then explaining how to manipulate the digits.

Using successors is rather similar to using base 1 notation, no lookup table is needed on how to add individual digits.

Proving Commutativity

As an example for an application of the Dedekind-Peano axioms, let us prove that the addition function is commutative. The proof will be rather overly formal, but is will show clearly where the axioms are used.

Again, to make clear that we are dealing with a function defined formally by the axioms, we write

$$\mathsf{add}(x, y) \quad \text{instead of} \quad x + y$$

The careful, pseudo-formal argument is rather tedious, but everybody should do this kind of thing a couple of times, before reverting back to a to less formal presentation.

Batten down the hatches. We will need a few auxiliary results.

Auxiliary Claim

Claim (1)

$\mathsf{add}(0, y) = y$

Proof. Let $\psi(z)$ be $\boxed{\mathsf{add}(0, z) = z}$

Base case: $\psi(0)$: $\mathsf{add}(0, 0) =_{\mathsf{add}_1} 0$

Inductive step:

IH: $\psi(z)$: $\mathsf{add}(0, z) = z$

IC: $\psi(S(z))$: $\mathsf{add}(0, S(z)) = S(z)$

$$\mathsf{add}(0, S(z)) =_{\mathsf{add}_2} S(\mathsf{add}(0, z)) =_{\mathrm{IH}} S(z)$$

□

And Another

Claim (2)

$\mathsf{add}(x, S(y)) = \mathsf{add}(S(x), y)$

Proof. Let $\psi(z)$ be $\boxed{\mathsf{add}(x, S(z)) = \mathsf{add}(S(x), z)}$

Base case: $\psi(0)$: $\mathsf{add}(x, S(0)) = \mathsf{add}(S(x), 0)$

$$\mathsf{add}(x, S(0)) =_{\mathsf{add}_2} S(\mathsf{add}(x, 0)) =_{\mathsf{add}_1} S(x) =_{\mathsf{add}_1} \mathsf{add}(S(x), 0)$$

Inductive step:

IH: $\psi(z)$: $\mathsf{add}(x, S(z)) = \mathsf{add}(S(x), z)$

IC: $\psi(S(z))$: $\mathsf{add}(x, S(S(z))) = \mathsf{add}(S(x), S(z))$

$$\begin{aligned}\mathsf{add}(x, S(S(y))) &=_{\mathsf{add}_2} S(\mathsf{add}(x, S(y))) \\ &=_{\mathrm{IH}} S(\mathsf{add}(S(x), y)) \\ &=_{\mathsf{add}_2} \mathsf{add}(S(x), S(y))\end{aligned}$$

□

Commutativity

Lemma

$\mathsf{add}(x, y) = \mathsf{add}(y, x)$

Proof. Let $\psi(z)$ be $\boxed{\mathsf{add}(x, z) = \mathsf{add}(z, x)}$

Base case: $\psi(0)$ is Claim 1.

IH: $\psi(z)$: $\mathsf{add}(x, z) = \mathsf{add}(z, x)$

IC: $\psi(S(z))$: $\mathsf{add}(x, S(z)) = \mathsf{add}(S(z), x)$

$$\begin{aligned}
\mathsf{add}(x, S(z)) &=_{\mathsf{add}_2} S(\mathsf{add}(x, z)) \\
&=_{\mathrm{IH}} S(\mathsf{add}(z, x)) \\
&=_{\mathsf{add}_2} \mathsf{add}(z, S(x)) \\
&=_{\mathrm{C2}} \mathsf{add}(S(z), x)
\end{aligned}$$

□

A Horror

These proofs are a horror—for humans. For theorem provers they are not so bad, simply because they are extremely mechanical.

There is the problem of figuring out that one needs to prove the claims first, but once these subgoals are recognized, everything is pretty straightforward.

Obviously we are going to be much more relaxed about details in the future.

More Proofs

Claim

$x + (y + z) = (x + y) + z$

Claim

$x + y = x + z$ *implies* $y = z$

Claim

$\forall\, x\, \exists\, y\, (x = 0 \vee x = S(y))$

Claim

$x < y$ *iff* $\exists\, z\, (z \neq 0 \wedge x + z = y)$

Food for Thought

Problem: Axiomatize the integers.

In other words, modify Dedekind-Peano arithmetic so it describes the integers rather than just the naturals.

There are several ways of doing this. For example, one could try to use a combination of successor function and negation, using axioms such as

$$-0 = 0$$
$$--x = x$$
$$S(-S(x)) = -x$$

So negation is an involution with fixed point 0. Needless to say, S also has to be a bijection in this context.

Paradise?

One might think that this is the end of the story: given any valid claim in arithmetic, we can just sit down and hammer out a proof like the last one, using no more that the Dedekind-Peano axioms.

We would very much like the following to be true:

> A first-order sentence ψ is true over the natural numbers
>
> if, and only if,
>
> ψ can be proven in first-order logic from the Dedekind-Peano axioms.

Dire Warning

If only. As far as we know, only true statements can be proven from the Dedekind-Peano axioms (consistency).

It is also true that one can obtain many, many results this way.

Alas, not **all** valid statements about the naturals can be proven this way. Some would say: everything that's really important (and they would be wrong).

This is all Gödel's fault: his infamous Incompleteness Theorem ruined everything. Even worse, there is no way to fix these problems.

The reason why things go bad is that the Dedekind-Peano axioms do **not** pin down the naturals completely. There are other, unintended ways to interpret the axioms:

$$\mathbb{N}^\star = \mathbb{N} \quad \underbrace{\ldots\mathbb{Z}\ldots\mathbb{Z}\ldots\mathbb{Z}\ldots}_{\mathbb{Q}}$$

The non-standard "naturals" start out with a copy of $\mathbb{N}$, but that copy is followed by $\mathbb{Q}$-many copies of $\mathbb{Z}$, containing "infinitely large" numbers. If we extend this to the reals we get *actual* infinitesimals[†], arguably a much better sandbox to play calculus in.

You're welcome.

[†]This was done by A. Robinson, Non-Standard Analysis, in the 1960s