$2 + 2 = 4$.

Kevin Buzzard

Imperial College London

18th Feb 2019.

What is this talk going to be about? It has four parts.

- Induction and recursion
- What is a number?
- $2 + 2 = 4$
- Should we teach maths to all year ten kids?

Let's watch a YouTube video of some things falling over.

Why did they fall over?

Let's now be *scientists*, and carefully investigate the basic logic of why all those things fell over.

There are *two basic ingredients* to making a whole bunch of things fall over like that. What are they?

The conclusion.

There were two key ingredients, for making everything in a line fall over:

Ingredient 1. Someone pushes over the first thing, so it falls over.

Ingredient 2. If a thing falls over, then the next thing falls over.

Conclusion: If we have *both those ingredients*, everything in the line will fall over.

A more mathematical example.

My eccentric uncle gave me £53 when I was born.

And then every year, on my birthday, he comes to visit me and gives me either £10 or £20, depending on how rich he's feeling that particular year.

It's really hard to work out the total amount of money he has given me!

But what do you think the *last digit* of that number is?

Hypothesis: the last digit of the total number of pounds he has given me is *always 3*.

Let's be scientists, and let's prove this hypothesis using the mathematical technique called *induction* (which is just the "things falling over" technique).

We need to check that we have the two ingredients which make the technique work.

My uncle gives me £53 pounds when I was born. Then every year he gives me £10 or £20.

Claim: the total number of pounds he has given me, ever since I was born, always ends in 3.

Ingredient 1: the beginning.
When I was born, he gave me £53, and that ends in three.

Ingredient 2: getting a year older.
Let's say that when I was x years old, the number of pounds he'd given me so far ended in three. And then let's say it's my birthday and I am $x + 1$ years old today, and he gives me some more money – either £10 or £20.

If a number ends in 3, and you add ten to it, then the new number you get ends in 3.

And if a number ends in 3, and you add 20 to it, then the new number you get ends in 3.

So *if the total ended in 3 before my birthday, it ended in 3 after my birthday!*

So let's consider the total amount my uncle has given me.

- When I was born, it ended in 3 (by ingredient 1).
- So after my 1st birthday, it ended in 3 (by ingredient 2).
- So after my 2nd birthday, it ended in 3 (by ingredient 2).
- So after my third birthday, it ended in 3 (by ingredient 2).
- ...
- After my 50th birthday, it ended in 3 (by ingredient 2).

It always ends in 3.

Numbers!

What is a number?

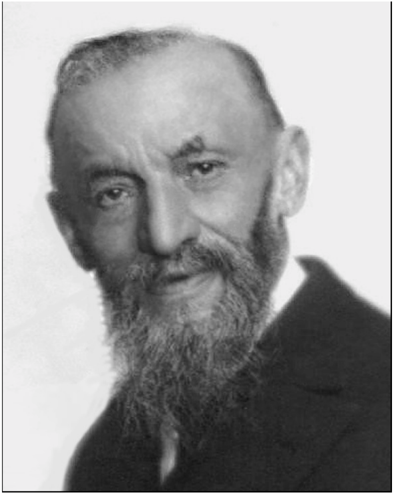
People have had an intuitive idea of what a number is ever since ancient shepherds thousands of years ago wanted a system for making sure nobody had pinched any of their sheep. Later on, numbers were used in engineering, finance, navigation and so on.

One thing is for sure – numbers are *really useful* for doing important real-world things.

But after a while, logicians began to wonder what the true nature of number really was – how could you "define" a number, without talking about numbers?

There are lots of kinds of numbers, but let's concentrate today on the easiest kind – the *natural numbers* 0, 1, 2, 3, 4, ... When I talk about a *number* in this talk, I mean a *natural number.*

Giuseppe Peano



In 1889, the Italian logician Giuseppe Peano formulated *axioms*, which define the natural numbers. Let me show you a modern-day version of his axioms.

- Axiom 1: 0 is a number.
- Axiom 2: The *successor* of a number is a number (i.e. the number after a number, is a number).
- That's it.

How do you think Peano defined 1? How do you think he defined 53?

Let's go and make some numbers using `Lean`, a computer language based on logic. [cut to Lean]

Peano's definition of a number.

- Axiom 1: 0 is a number.
- Axiom 2: The *successor* of a number is a number.
- That's it.

Now here's a cool thing. If we use Peano's definition of a number, we can begin to understand why the principle of induction actually works.

Let's say we want to do something for every number. Let's say we've done it for zero. And let's say that if we've done it for a number x, then we've done it for the successor of x (the number after x). Then, because those two axioms above are *the only way we can make numbers*, we must have done it for every number.

It's just like the falling blocks.

Let's think about the function which adds 2 to a number.

Remember – the computer knows *nothing* about numbers, except that `zero` is a number, and if `x` is a number then `S(x)` is a number.

We want to make this function which adds 2 to a number. Let's build it using Peano's inductive principle.

First we have to say what the answer is when we add 2 to 0. It's 2.

Now here's the hard part.

Let's say we already know what we get when we add 2 to x. Let's say $2 + x = y$. Now what do we want the answer to be when we add 2 to the number after x? We want it to be the number after y.

Let's teach this to the computer. [cut to Lean again]

Let's now prove that $2+2=4$.
Before we start on this, remember how we defined the "add two" function? We defined $2+0$ to be 2, and we have already defined $2+x$ then we defined $2+S(x)$ to be $S(2+x)$. So...

$$\begin{aligned} 2+2 &= 2+S(1) \text{ (by definition of 2)} \\ &= S(2+1) \text{ (by definition of +)} \\ &= S(2+S(0)) \text{ (by definition of 1)} \\ &= S(S(2+0)) \text{ (by definition of +)} \\ &= S(S(2)) \text{ (by definition of +)} \\ &= S(3) \text{ (by definition of 3)} \\ &= 4 \text{ (by definition of 4)} \end{aligned}$$

So $2+2=4$. QED!

Now let's watch the computer do it. [cut to Lean]

The computer knows basic logic, and then we teach it numbers, and then it's instantly an expert in numbers.

What other maths do you think we can teach it?

And how long do you think it would take us?

Lean was written by Leonardo de Moura, from Microsoft Research, in 2014. Unlike many Microsoft products, Lean is *free* and *open source*, so I like using it, because anyone can use it.

At Imperial College London, a team of undergraduates, led by me, is teaching that computer program our entire maths degree.

Last term, we managed to teach it all of my "introduction to proof" course which I teach the first year undergraduates ("year 14").

Last week, we taught it how to do all the questions on the exam I set last summer.

By the time you're old enough to go to university, we will be doing all the 3rd year pure mathematics courses. Some of them we have already half-done.

We're writing up beautiful course notes, which don't have any mistakes in – the computer checks all the maths!

When we've finished teaching the computer our entire maths degree, I want to see if we can start getting it to do its own research.

Because AI and machine learning are moving *incredibly* quickly these days, one has to be optimistic.

And once computers can do maths better, faster and more accurately than humans – one has to start asking what the point of teaching maths to every schoolchild up to the age of 16 really is. Why not just teach it to those that *like* doing it, like me?

Thank you for your attention!