The Electronic Mathematician

Why I no longer have to do my problem sheets

James Arthur

Data Science Cornwall, August 2021

Summary

Introduction

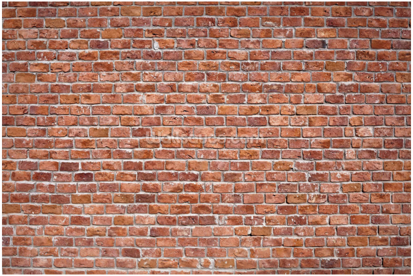
Academic Mathematics

Lean

Lean Demo

A wall

Imagine a wall.



A wall

Imagine a hole in the wall.

A wall

Imagine a hole in the wall.

Slightly less solid, but we ignore this, we carry on with the wall.

A wall

Imagine a hole in the wall.

Slightly less solid, but we ignore this, we carry on with the wall. More holes appear and **crack**. The wall collapses and falls.

How to stop the fall?

1. If this were really a wall, we would just mend it and fill the holes with bricks.

How to stop the fall?

1. If this were really a wall, we would just mend it and fill the holes with bricks.
2. This isn't any old wall as you can't see where the holes are, the missing bricks seem much like the ones in the wall.

How to stop the fall?

1. If this were really a wall, we would just mend it and fill the holes with bricks.
2. This isn't any old wall as you can't see where the holes are, the missing bricks seem much like the ones in the wall.
3. Hence, you can't find which bricks to replace.

How to stop the fall?

1. If this were really a wall, we would just mend it and fill the holes with bricks.
2. This isn't any old wall as you can't see where the holes are, the missing bricks seem much like the ones in the wall.
3. Hence, you can't find which bricks to replace.
4. So the answer is prevention not cure.

Academic Error

So what are we really talking about?

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Academic Error

So what are we really talking about? We are talking about the structure of academic maths.

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Academic Error

So what are we really talking about? We are talking about the structure of academic maths.

1. From 1996 to 2019 alone there were 214299 papers written in the UK on the whole of mathematics [1]

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Academic Error

So what are we really talking about? We are talking about the structure of academic maths.

1. From 1996 to 2019 alone there were 214299 papers written in the UK on the whole of mathematics [1]
2. Everybody makes mistakes and we can't blame the people for them, but they are an academic problem. If a paper is wrong, then they are our missing bricks.

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Academic Error

So what are we really talking about? We are talking about the structure of academic maths.

1. From 1996 to 2019 alone there were 214299 papers written in the UK on the whole of mathematics [1]
2. Everybody makes mistakes and we can't blame the people for them, but they are an academic problem. If a paper is wrong, then they are our missing bricks.
3. So, what's the 'prevention'?

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Academic Error

So what are we really talking about? We are talking about the structure of academic maths.

1. From 1996 to 2019 alone there were 214299 papers written in the UK on the whole of mathematics [1]
2. Everybody makes mistakes and we can't blame the people for them, but they are an academic problem. If a paper is wrong, then they are our missing bricks.
3. So, what's the 'prevention'? Well formalisation.

[1] Source: https://www.scimagojr.com/countryrank.php?area=2600

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

1. Taking a brick out of the wall.

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

1. Taking a brick out of the wall.
2. Breaking it down into it's constituent parts.

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

1. Taking a brick out of the wall.
2. Breaking it down into it's constituent parts.
3. Rebuilding it from those parts.

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

1. Taking a brick out of the wall.
2. Breaking it down into it's constituent parts.
3. Rebuilding it from those parts.
4. Checking the state of the brick (true or false).

Formalisation

The cure requires a programme called an Interactive Theorem Prover and an activity we coin as formalisation. Formalisation requires the following steps,

1. Taking a brick out of the wall.
2. Breaking it down into it's constituent parts.
3. Rebuilding it from those parts.
4. Checking the state of the brick (true or false).
5. Dealing with it.

Summary

Proofs, Theorems, Lemmas

Mathematics is built on many different structures, much like our mortar, clay and aggregate bricks.

- **Lemma:** Smaller less important results, like Zorns Lemma.
- **Theorem:** Big results, these are the famous ones, like Fermat's Last Theorem.
- **Proofs:** The reason why the above are true.

What is a proof?

Our bricks are proofs and papers. So what is a proof,

What is a proof?

Our bricks are proofs and papers. So what is a proof,

- A proof is a string of logical deductions.

What is a proof?

Our bricks are proofs and papers. So what is a proof,

- A proof is a string of logical deductions.
- It's also a way of mathematical expression.

What is a proof?

Our bricks are proofs and papers. So what is a proof,

- A proof is a string of logical deductions.
- It's also a way of mathematical expression.
- To prove something you must take your reader on a journey, through things that you know are true to a final fact that you want the reader to believe is true.

What is a proof?

Our bricks are proofs and papers. So what is a proof,

- A proof is a string of logical deductions.
- It's also a way of mathematical expression.
- To prove something you must take your reader on a journey, through things that you know are true to a final fact that you want the reader to believe is true.
- This doesn't mean every proof is readable though, proofs often take rough and rocky mountain paths instead of a nice stroll though the botanical gardens.

The Natural Numbers

As an aside, I would like to quickly define formally what I mean by the natural counting numbers.
In 1889, Peano proposed the following definitions for the positive counting numbers $(0, 1, 2, 3, 4, \dots)$. The following axioms were provided,

1. 0 is a natural number
2. Equality makes sense, so we can say $1 = 1$
3. $n + 1$ is a natural number (successor).
4. If $m = n$, then $m + 1 = n + 1$
5. There doesn't exist a natural number such that $0 = n + 1$.
6. If a statement is true for $n = 0$ and can be proved for $n + 1$ from an assumption for n, then it is true for all natural numbers (induction).

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$.

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$. Let us take a and b to just be natural numbers. Why is this true?

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$. Let us take a and b to just be natural numbers. Why is this true? Well we need to look at the brick, or the proof.

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$. Let us take a and b to just be natural numbers. Why is this true? Well we need to look at the brick, or the proof.

Sketch proof:

1. Take induction on b,

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$. Let us take a and b to just be natural numbers. Why is this true? Well we need to look at the brick, or the proof.

Sketch proof:

1. Take induction on b,
2. We have a base case of proving that $0 + a = a + 0$, which is simple. We can do this instantly.

Maths Proof

Let us take a statement we all should agree on, $a + b = b + a$. Let us take a and b to just be natural numbers. Why is this true? Well we need to look at the brick, or the proof.
Sketch proof:

1. Take induction on b,
2. We have a base case of proving that $0 + a = a + 0$, which is simple. We can do this instantly.
3. Now we have to show that $a + succ(b) = succ(b) + a$ assuming that $a + b = b + a$,

$$\begin{aligned} a + succ(b) &= succ(b) + a \\ succ(a + b) &= succ(b + a) && \text{as } succ(x) = x + 1 \\ succ(a + b) &= succ(a + b) && \text{by induction hypothesis} \end{aligned}$$

Lean Proof

Heres a proof that $a + b = b + a$,

Lean Proof

Heres a proof that $a + b = b + a$,

```
lemma add_comm (a b : ℕ) : a + b = b + a :=
begin
  induction b with base_case induction_hypothesis,
  { rw [zero_add, add_zero]
  },
  { rw [add_succ, induction_hypothesis, succ_add]
  }
end
```

Summary

What is Lean?

- Lean is an entirely functional programming language of sorts.

What is Lean?

- Lean is an entirely functional programming language of sorts.
- It is based of type theory, you create functions by creating what are seen as proper mathematical statements.

What is Lean?

- Lean is an entirely functional programming language of sorts.
- It is based of type theory, you create functions by creating what are seen as proper mathematical statements.
- You provide it with these by defining mathematical objects, like the sine function ($\sin\alpha$) and proving things about them, i.e. $\sin(2\alpha) = 2\sin\alpha\cos\alpha$.

What is Lean?

- Lean is an entirely functional programming language of sorts.
- It is based of type theory, you create functions by creating what are seen as proper mathematical statements.
- You provide it with these by defining mathematical objects, like the sine function ($\sin \alpha$) and proving things about them, i.e. $\sin(2\alpha) = 2 \sin \alpha \cos \alpha$.
- Technically what I call Lean and what is the maths library aren't the same thing, however I'm more interested in the maths side of things, so we shall take them as the same.

So what has been formalised?

We have formalised quite a lot of stuff, Lean now has 500,000 lines of proof, definition and statements.

So what has been formalised?

We have formalised quite a lot of stuff, Lean now has 500,000 lines of proof, definition and statements. That is a lot of Maths. With 23826 definitions, 52842 Theorems / Lemmas and 161 Contributors there's a high chance what you want to formalise can be formalised using Lean.

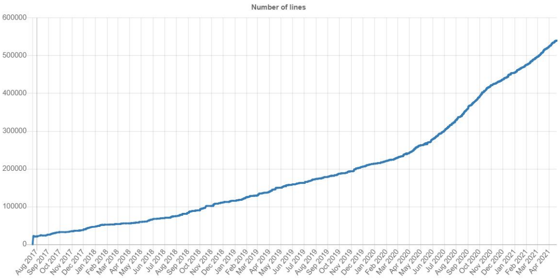


Figure: Number of lines of code over time.

So what has been formalised?

There is work in almost every field of pure mathematics and some fields of applied mathematics. As applied mathematics is presented in a slightly different way it is hard to work off the pure work at times as it's usually highly generalised to prevent code repetition.

So what has been formalised?

There is work in almost every field of pure mathematics and some fields of applied mathematics. As applied mathematics is presented in a slightly different way it is hard to work off the pure work at times as it's usually highly generalised to prevent code repetition.

I work in Analysis, so I work with things like defining and proving things about $\operatorname{arsinh} x$, i.e. $\sinh \operatorname{arsinh} x = x$. I have also worked on proving the astounding result that the area of the unit circle is π!

Summary

Introduction

Academic Mathematics

Lean

Lean Demo

Sine and Cosine

I'm going to quickly talk through a few bits and bobs before I start showing you some Lean.
We can talk about the unit circle,