What makes a mathematician tick?

Kevin Buzzard

Imperial College London

ITP2019, 9th September 2019.

- $3^2 + 4^2 = 9 + 16 = 25 = 5^2$. Similarly $5^2 + 12^2 = 13^2$ and so on.
- Fermat's Last Theorem says that there are no solutions in positive integers x, y, z, n to $x^n + y^n = z^n$ if $n \geq 3$.
- 20 years ago, mathematicians proved this theorem. It took us 350 years to find the proof.
- The proof would be thousands of pages long if written out in full in pdf format.
- I don't really see any obstruction to checking the proof using a computer proof verification system.
- But it would take a long time (many person-decades).
- The funny thing is, if it got formalised, most "proper mathematicians" would not really care at all.
- Why not?
  1. Because Fermat's Last Theorem is *old news*.
  2. And because formalising it would teach us "proper mathematicians" nothing that we didn't already know.

What does the proof of Fermat's Last Theorem look like?

- In 1990 Ken Ribet proved that Fermat's Last Theorem followed from the semistable Shimura–Taniyama conjecture.
- The semistable Shimura–Taniyama conjecture is a conjectural relation between certain things called "elliptic curves" and certain things called "modular forms".
- It takes roughly 50 hours of MSc level algebraic number theory lectures to develop enough theory to *state* it.
- Wiles and Taylor–Wiles proved it in 1995.
- Their proof used the cohomological interpretation of local and global class field theory, the deformation theory of finite flat group schemes, Galois cohomology, etale cohomology, theory of the mod p reduction of moduli spaces of elliptic curves, the Langlands–Tunnell theorem, harmonic analysis, algebraic geometry and arithmetic geometry from the 1970s and 1980s.
- It would take 50 more hours of PhD level number theory seminars to *define these objects.*

- In 2000 the *full* Shimura–Taniyama conjecture was proved.
- We can now prove certain higher-dimensional versions of the Shimura–Taniyama conjecture using extensions of the ideas of Wiles and Taylor–Wiles, and some new ideas. Our new proofs are much longer. And we have profound conjectural generalisations of the Shimura–Taniyama conjecture, some still completely inaccessible.
- This is why we care about the proof of Fermat's Last Theorem. *Because it taught us new proof techniques.*

- I believe that no living human knows all the details of the proof of Fermat's Last Theorem.
- However, the proof is modular, making it much easier to judge. My community has accepted the proof. Indeed, we gave Wiles the Abel prize.
- We have a community of elders, who rule on whether work is correct.
- The methods of the elders are subtle. They smell new work. The elders have very sensitive noses, and will probably spot if something is making a bad smell.
- *These methods have worked for hundreds of years, and continue to work.*
- "We don't need your computer proof checkers, we know how to check things by ourselves."

- *The typical methods used by computer scientists to advertise computer proof systems are not of interest to mathematicians.*
- Automatic theorem provers are giving us incomprehensible 20,000 line proofs of statements about quasi-groups which no “proper mathematician” is interested in.
- Interactive theorem provers are being used to verify classical old theorems such as the Prime Number Theorem or the Odd Order Theorem, following the blueprint laid out by mathematicians. This work gives mathematicians *no interesting new insights*.
- Let me stress that I am *extremely interested personally* in this sort of work mentioned above. But most of my colleagues are not.
- I have seen the tools that are coming out of this area and I am *convinced that one day they will change mathematics.*
- But for the change to occur, *mathematicians must become involved.* So how are we going to make this happen?

What gets a mathematician excited?

Number one thing: an announcement of a proof of a famous conjecture.

Why? Because if a conjecture is famous, it is probably hard.

So a proof of it will almost always involve at least one *beautiful new idea.*

This idea will probably be in *raw primitive form*, but we can attempt to "understand the technique", generalise it, and then use it to prove even harder things.

Indeed, mathematicians *excel* at knowing how far all currently known ideas will take us, and an elder can often see a new idea and then, mere *minutes* later, have a clear vision as to how much further the idea will be able to take us.

Mathematics can be viewed as a series of challenges. A series of mountains to climb.

The elders know which ones have been climbed, and they also know why we cannot climb certain other mountains. Their understanding of the landscape is very profound.

Mathematicians fashion new tools out of thin air, and pass them around in pdf or lecture format. The tools are simply ideas. We use new tools to scale the mountains.

Sometimes mathematicians make new *definitions*. Whole new mountain ranges can then appear in an instant.

Sometimes the experts get it wrong. An unconquered mountain once became strategically important. I effortlessly walked to the top of it, using only tools other people had made. I wrote an extremely short paper which probably nobody ever read. And then a second paper with the application, which lots of people read.

A second thing mathematicians get excited about: the *statement* of a *profound conjecture* – even if if it has not been proved. Because conjectures provide guidance.

Here are The Clay Mathematics Institute's "Millennium Prize Problems", announced in 2000:

1. Poincaré conjecture;
2. P vs NP;
3. Hodge Conjecture;
4. Riemann Hypothesis;
5. Navier-Stokes existence and smoothness;
6. Birch and Swinnerton-Dyer conjecture;
7. Yang–Mills existence and mass gap.

Of these, the Poincaré conjecture is proved, Yang–Mills is not a precise question, and the other five are open problems.

As far as I know, the *statement* of the Riemann Hypothesis has been formalised in Isabelle/HOL, but several of the others have not been formalised in any other theorem prover at all.

Here is a *special case* of the Birch and Swinnerton-Dyer conjecture.

Say A, B are fixed rational numbers.
Let E denote the equation $y^2 = x^3 + Ax + B$.
Now let's attach some data to E.

- Let $L(s)$ denote the L-function of E (here $s \in \mathbb{C}$);
- let $Ш$ denote the Tate–Shafarevich group of E;
- let C denote the real number which is the product of the Tamagawa factors for E at all places;
- let S be the set of solutions to E with x and y rational numbers.

Conjecture: if S is finite, then

$$L(1) = \frac{|Ш| \times C}{(1 + |S|)^2},$$

and if S is infinite then $L(1) = 0$.
[NB here $|X|$ denotes the size of X.]

OK so let's imagine formalising the *statement* of BSD!

$$L(1) = \frac{|Ш| \times C}{(1 + |S|)^2}.$$

Definition of the function $L(s)$: it is an infinite sum. A tricky theorem of Hasse shows that the infinite sum converges for $Re(s) > 1\frac{1}{2}$. The conjecture is about the value at $s = 1$. How do we extend the function to $s = 1$? The only known way is to use the proof of the Shimura–Taniyama conjecture, which is thousands of pages long. So basically we have to formalise a proof of Fermat's Last Theorem to make the left hand side make sense.

The right hand side still is not known to make sense. The definition of $Ш$ is early PhD level mathematics – maybe ten hours of material? The group $Ш$ is conjectured, but not known, to be finite.

Computers are a million miles from proving any of this. Asking when they will prove it is the *wrong question*.

Here's an idea, coming from Microsoft Research.

The *International Mathematical Olympiad* is a yearly competition where the best high school kids from 100+ countries try to solve six tricky maths puzzles.

"Proper mathematicians" might turn their noses up at these puzzles (even though some of us were good at them when we were kids...). However they are *difficult*, so it would be some sort of achievement if a computer were to solve one of these puzzles.

Lean is a theorem prover being developed at Microsoft Research. MSR have proposed the IMO Grand Challenge. They want to write a tactic in Lean which solves IMO problems.

I think this is a great idea because it is (a) an interesting approach, (b) it might well work, and (c) it will show the world that "computers can solve hard maths problems".

Here is an idea, coming from Tom Hales.

Before a computer can find a proof of Birch–Swinnerton-Dyer, it has to be able to understand the statement.

So it needs to understand the following definitions: elliptic curves, modular forms, the statement of the Taniyama–Shimura conjecture, group cohomology, Galois cohomology, the Galois theory of local and global fields, analytic and algebraic ranks of elliptic curves, unramified cocycles, and the L-function of a modular form.

This is MSc and early PhD level mathematics. A post-doc working with me could formalise all that stuff in Lean within a couple of years.

And what would the pay-off be? Stating one conjecture? We would be able to state 20 conjectures and also 20 deep theorems. This is the sort of stuff that Wiles and Taylor needed. We would be able to *begin to tell computers the statements of what we know and believe to be true.* We'd be able to tell the computers which mountains we believe we can scale.

If mathematicans started to formalise difficult and deep *definitions* in theorem provers, and *statements* of modern conjectures/theorems instead of their proofs, then before long we would have a *completely new kind of database*.

If teams of people started popping up and formalising basic modern mathematical definitions, we could have this database within a few years.

We would have some sort of a record of what humans claim to have achieved in mathematics. Could a computer AI learn from this? Maybe.

Could humans use this database for search? Surely!

Are computer proof systems up to this task? Johan Commelin, Patrick Massot and myself have proved this.

In 2018 the Fields Medallists were announced: Peter Scholze, Alessio Figalli, Akshay Venkatesh, and Caucher Birkar.

Each of these people proved at least one brilliant theorem in the last few years.

None of our computer proof systems can, in their current states, easily formalise the *statements* of *any* of these theorems. The *definitions of the objects* are all missing.

This is because mathematicians understand the definitions but don't use the provers, and computer scientists use the provers but don't know the maths.

Peter Scholze's work involved defining a new geometric object called a *perfectoid space*. In May this year, Patrick Massot, Johan Commelin and myself (all mathematicians, all Lean users) formalised the *definition* of a perfectoid space in Lean. A *definition*! Our work is a proof that Lean can handle a towering modern mathematical definition.

A perfectoid space is a geometric object with *prescribed local behaviour*. Let me explain a simpler example first.

A *topological space* is a general geometric math object. A *2-dimensional manifold* is a topological space which "looks 2-dimensional everywhere". What does this mean?

Well, mathematicians are happy with the idea that a disc in the plane is 2-dimensional. A 2-dimensional manifold M is a topological space which can be written as a union $M = \bigcup_{i \in I} M_i$, where each M_i is isomorphic to a disc in the plane.

M is a topological space, so the M_i are topological spaces, isomorphic (as topological spaces) to discs. In particular, we need that a disc is a topological space. This is not hard.

A perfectoid space is also geometric object which locally looks like a model space.

A perfectoid space is a topological space with a lot of extra structure (e.g. a presheaf of complete topological rings satisfying a bunch of axioms), which can be covered by subspaces, each of which is isomorphic to the spectrum of a Huber pair.

For this definition to even typecheck, we need to check that the spectrum of a Huber pair has all this structure and satisfies these axioms. [Analogous to the result that a disc is a topological space.]

Defining the structure and checking the axioms is around 10,000 lines of code.

We finished formalising the definition in May. We are writing the API for this definition. Our goals: (1) give more examples of perfectoid spaces. (2) State Scholze's Tilting Correspondence (a theorem about perfectoid spaces).

Why are we doing this? Because one last thing that gets mathematicians excited is *sexy complicated new definitions* which can be used to prove new theorems.

Right now, a number theory PhD student can download Lean and then download a perfectoid space onto their computer, and play with that perfectoid space.

Within a year, a number theory PhD student will be able to download lots of examples of perfectoid spaces, and do far more stuff with them.

It's not new proofs of theorems. But it's a *new way of learning about mathematical objects*. And because we can't do proofs yet, why don't we try it?