ProofBench: a proposal for a Lean-verified benchmark for proof formalisation and development

Nathan Bowler

March 2025

1 Overview

When the mathematical ability of an AI system is tested using a benchmark, it typically has to answer hundreds or even thousands of questions. Since the benchmark is used repeatedly on a variety of AI systems, a major constraint is that the accuracy of the answers to the questions must be automatically checkable. As a result, recent state-of-the-art advanced mathematics benchmarks such as MATH or FrontierMath typically rely on multiple choice questions or questions with numerical answers. Although this makes verification of the answers straightforward, it limits the depth of mathematical reasoning which can be tested. We propose a benchmark in which the answers are proofs formalised in Lean, which allows testing of a greater depth of mathematical understanding whilst still allowing automatic verification.

The developers of FrontierMath explain their reasons for not taking this approach as follows.

> Firstly, current AI models are often not trained to write formalized proofs in specialized languages. We wanted to ensure that the problems were measuring genuine reasoning skills, rather than skill at writing formalized proofs. Secondly, Lean's mathematical library, mathlib, doesn't fully cover the undergraduate math curriculum, let alone the scope of research math, which limits the fields such a benchmark could measure.

However, writing formalised proofs is an important skill in its own right, and the emergence of AI systems with that skill is likely to be revolutionary regardless of the ability of those systems to develop original proofs. Our proposed benchmark would be ideal for testing the extent to which AI systems possess that skill.

Furthermore, once AI systems which are capable of formalisation are available, they can be combined with our proposed benchmark to test the abilities of

AI systems which produce proofs couched in natural language, since such proofs could be first automatically formalised then automatically verified using Lean.

Similarly, such systems would allow the second objection to be rapidly overcome in later iterations of the benchmark, since the formalisation of the remainder of the undergraduate curriculum (and any necessary postgraduate material) would be greatly accelerated through automatic formalisation.[1]

Since the skills involved in formalising proofs are closely related to (a proper subset of) the skills needed to produce original proofs, we expect that AI systems capable of proof formalisation will precede those capable of proof development, and that a broad system trained to have both skills could be better at each skill individually than systems specialised for either task.

Not only are the skills required for the two domains of proof formalisation and development closely related, the boundary between them is hard to draw clearly. A precise and detailed proof is easier to formalise than a vague proof sketch. The higher the level of the proof, the closer the problem becomes to one of pure proof development. That is the reason for addressing both problems with a single benchmark. By so doing, we are also able to evaluate, for each question, where on this continuum the current system's abilities lie.

2 Structure of the benchmark

Each question in the benchmark will consist of a theorem statement formalised in Lean, together with some additional helpful information. The theorem will be an unpublished result at the level which it would be reasonable to give as a short starter question to a PhD student. The theorem statement will include formalised versions of any background tools or known statements to be used in the proof, to the extent that these are not already included in the standard Lean library.

The helpful information will consist of:

1. A statement of the same theorem in natural language.

2. A sketch proof of the theorem in natural language, at the level of abstraction which would be typical when suggesting an approach to the problem to a PhD student.

3. A proof of the theorem in natural language, at the level of abstraction which would be typical for a journal article. For the difficulty level of the questions we have in mind, these proofs will of course be much shorter than a typical journal article - about five pages long.

4. A formal outline of a proof produced using the Lean Blueprint tool.

[1] In any case, this second objection weighs more heavily for benchmarks like FrontierMath, with an emphasis on breadth of mathematical knowledge, than for benchmarks which are focused on mathematical insight and problem-solving ability.

The theorems will be chosen to cover a broad range of pure mathematical disciplines, and will be classified by discipline. For the evaluation portion of the benchmark, the questions will be kept secret to avoid contamination of the training data.

When tested using the benchmark, an AI system will be prompted with the formal theorem statement, together with some subset of the helpful information (for example, (1)-(4), (1)-(3), (1) and (2), (1) and (3), (1) only, (4) only or no helpful information at all). Its response will be compiled in Lean as a proof of the theorem in the question, and the answer will be judged correct if does indeed compile to give a formal proof of that theorem.

Thus the benchmark will allow flexible testing both of the relative strengths of AI systems in different mathematical disciplines, as well as their relative abilities on the different but related tasks of formalising and developing proofs.

The ability to formalise proofs provided in natural language can be tested by providing the helpful information of types (1) and (3). The ability to develop new proofs can be tested by providing (in increasing order of difficulty) (4) only or no helpful information at all. The other configurations of information sketched above interpolate between these tasks.

Note that the provision of a formalised theorem statement is essential, even when using the benchmark to test formalisation ability, since it allows the answers to be automatically evaluated.

There is an additional standard procedure to make benchmarks resistant to contamination of training data and to estimate the extent of that contamination by making the benchmarks *functional*. In our case, this could be implemented by sometimes randomising the strings used to name any new terms that are defined for the formal theorem statement.

We anticipate some difficulty in persuading working mathematicians to donate currently unpublished ideas at this level to be used in the benchmark, since they may have been hoping that they or their students could incorporate these ideas into future publications. To mitigate this issue, we propose rewarding problem contributors with a small bounty and a certificate of contribution.

As further mitigation, after a quarantine period of one year, we will allow the contributors to publish the ideas if they wish, at which point the problem in question will be moved from the private to the public portion of the benchmark. The private portion will be refreshed with new questions on a rolling basis. This also allows for more difficult questions (which must be labeled as such according to a transparent framework) to be added over time as AI systems improve.

3 The ideal team

To produce a benchmark of this kind, a large and diverse team will have to work closely together. Ideally, this team would include:

- Five mathematics professors with very different mathematical backgrounds (for example algebra, analysis, statistics, combinatorics and logic), each

supported by a postdoc or 2 PhD students. Their role would be to coordinate and oversee a much larger group of hundreds of mathematicians, who would produce the mathematical parts of the benchmark (points (1), (2) and (3)) from the last section, as well as consulting on the formal parts (point (4), the formal theorem statement, and a model answer to each question). They would also edit the questions as appropriate and ensure their quality and the consistency of their difficulty levels.

- Five formalisation experts with close ties to disparate areas of the Lean community. Their role would be to coordinate the production of the formal parts of the benchmark (the actual production would again require a larger group), and to advise the mathematicians on which aspects make theorem statements more or less suitable for inclusion in the benchmark.
- Three or four software engineers with experience developing and deploying benchmarks for AI systems.
- Two administrative support staff, to coordinate the logistics of the project and organise travel and facilities.

Everyone who is involved should know enough about the work of the others to be able to communicate with them effectively. To some extent this can be achieved by involving people who already have relevant background knowledge outside their specialty, but it will also be necessary to meet regularly (roughly once every 2-3 months) in person to exchange ideas and important background knowledge. At least two members of the team should have experience in media relations.

4 Impact

Clearly, a benchmark by itself has no independent impact; instead it is the AI systems which are likely to have a significant impact on the world. However, in many cases the impact of the AI system is partly dependent on trust and proper understanding of its capabilities by the wider mathematical community, which is something that benchmarks can certainly contribute to. In this section we will focus on impact of this kind, divided into the impact of proof formalisation systems and proof development systems.

4.1 Proof formalisation

AI systems for automated proof formalisation would be revolutionary in several ways. Here are a few key areas where they would make a difference:

Workflow of professional mathematicians Even once we have developed a persuasive argument, working mathematicians spend a great deal of time checking their own arguments and those of their close collaborators for correctness. With automated formalisation, this part of the workflow

would become faster and more reliable, and would only need to be done once for each argument.

Student feedback Students who are learning how to structure mathematical proofs would get real-time feedback on the validity of their arguments, without having to wait for feedback from teaching assistants.

Refereeing for journals Once it is feasible and not too onerous, most journals will make it a requirement to submit a formal proof together with each paper. The role of referees will be limited to a focus on the interest and significance of the results. This may lead to more substantial shifts in the typical business model of journals.

Finding historical mistakes There have been a number of significant results for which the proofs were discovered decades later to be flawed. Famous examples include the Four Colour Theorem, the Jacobian Conjecture and Hilbert's 21st Problem. It would therefore be rather surprising if none of the remaining body of significant mathematical proofs were marred by similar flaws.

Once automated formalisation is possible, it will be feasible (although still extremely difficult) to carry out a project of formalising all mathematical work since the introduction of the modern foundations of mathematics. This would allow us to identify historical mistakes and lacunae, and would place the remaining body of mathematics on a firmer foundation.

Foundations of mathematics Automatic formalisation will give a strong basis on which to build systems for automatic transfer of arguments between different foundations, removing the current need for reliance on a single shared foundation. It will also allow automatic evaluation of the logical strength of proofs, extending their applicability. For example, any statement whose proof is found to only rely on intuitionistic logic will immediately generalise to a lifted statement about any topos.

4.2 Proof development

Although on one hand the revolutionary implications of AI systems capable of proof development are far more intuitively clear, on the other hand the full ramifications of such systems touch on so many aspects of life that it would be futile to try to summarise them here. So we restrict ourselves to a few central impacts.

Workflow of professional mathematicians Given the ability to rapidly check whether an approach to a problem is promising, professional mathematicians will be able to more quickly iterate on their ideas and to devote more attention to questions of importance, structure and overall mathematical strategy.

Workflow of professional scientists Theoretical work in the sciences is often slowed down by uncertainty over mathematical questions. By using AI proof development systems, working scientists could in some cases bypass the need to wait for progress on such problems from the mathematical community.

Software validation The question of whether a particular bundle of software, running on hardware satisfying certain formal specifications, will necessarily produce or avoid specified outcomes, is a mathematical one, although one which is generally unattractive to human mathematicians. With automated proof development systems we would be able to answer such questions.

5 Why now?

The last year has seen tremendous progress in the capacity of AI systems to address mathematical problems. Highlights include the high IMO score obtained by DeepMind's AlphaProof system, falling one point short of a gold medal, and substantial scores (roughly 10%) from o3 on the highly challenging FrontierMath benchmark.

On benchmarks based on the same principles as the one outlined above, but with lower level questions, scores have been rising rapidly, including for example a 66% cumulative score from ProofAug on the MiniF2F benchmark.

Given this rapid progress, and the logistical hurdles involved in constructing a benchmark of this kind, now is the time to begin work on it if we want it to be ready by the time it will be most useful.