Abstract. Formalization is the translation of informal mathematical text into verifiable arguments. From Hilbert and Bourbaki to the QED manifesto, mathematicians have pursued this dream of a single source of truth, a single repository of all mathematical knowledge. Today, this dream is much closer than one might think. In this talk, we present the formalization problem and highlight the fundamental gap between informal proofs and formal reasoning. This is followed by a discussion on the current frontier of automatic formalization. We conclude with an outlook on future directions, as well as implications for the working mathematician.

On the Automatic Formalization of Mathematical Reasoning

Omar Elshinawy[‡]

Uni Münster

April 17, 2026[†]

[‡]o.elshinawy@uni-muenster.de, elshenawyom.github.io

[†]Datathon 2ϕ26, Maths Society @ Constructor University

A tweet?

A tweet?

Tweets

Omar El Shenawy @elshenawyom · Dec 18, 2022

why bother then 💀 https://t.co/Xkmvn3x0Se

Corollary 2.3. *Let c be a number and let the assumptions be as in the theorem. Then*

$$\lim_{x \to a} cf(x) = cL.$$

Proof. Clear.

Corollary 2.4. *Let the notation be as in the theorem. Then*

$$\lim_{x \to a} (f(x) - g(x)) = L - M.$$

Proof. Clear.

0 0

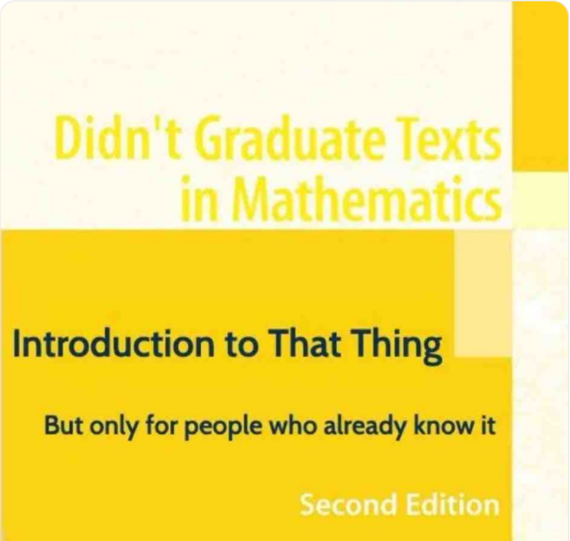
ID 1604518237659504640

View on Twitter

Another one?

Omar El Shenawy @elshenawyom · Oct 13, 2023

RT @AlgebraFact: https://t.co/zax19R7wsi



Final One I promise

Common proof techniques

Proof by intimidation Trivial!

Proof by cumbersome notation The theorem follows immediately from the fact that $\left|\bigoplus_{k\in S}\left(\mathfrak{K}^{\mathbb{F}^{\alpha}(i)}\right)_{i\in\mathcal{U}_k}\right| \preccurlyeq \aleph_1$ when $[\mathfrak{H}]_{\mathcal{W}} \cap \mathbb{F}^{\alpha}(\mathbb{N}) \neq \emptyset$.

Proof by inaccessible literature The theorem is an easy corollary of a result proven in a hand-written note handed out during a lecture by the Yugoslavian Mathematical Society in 1973.

Proof by ghost reference The proof my be found on page 478 in a textbook which turns out to have 396 pages.

Circular argument Proposition 5.18 in [BL] is an easy corollary of Theorem 7.18 in [C], which is again based on Corollary 2.14 in [K]. This, on the other hand, is derived with reference to Proposition 5.18 in [BL].

Proof by authority My good colleague Andrew said he thought he might have come up with a proof of this a few years ago...

Internet reference For those interested, the result is shown on the web page of this book. Which unfortunately doesn't exist any more.

Proof by avoidance *Chapter 3:* The proof of this is delayed until Chapter 7 when we have developed the theory even further. *Chapter 7:* To make things easy, we only prove it for the case $z = 0$, but the general case in handled in Appendix C. *Appendix C:* The formal proof is beyond the scope of this book, but of course, our intuition knows this to be true.

facebook.com/Mathematicx

My Two Cents

From here we get three themes:

My Two Cents

From here we get three themes:

- Compression of logic,

My Two Cents

From here we get three themes:

- Compression of logic,
- Omission of reasoning,

My Two Cents

From here we get three themes:

- Compression of logic,
- Omission of reasoning,
- Unverifiability, at least, to the working mathematician.

My Two Cents

From here we get three themes:

- Compression of logic,
- Omission of reasoning,
- Unverifiability, at least, to the working mathematician.

Let's take a step back.

The Mathematician's Dream

- **Hilbert's Program** (1920s): Can we verify all mathematics mechanically?

The Mathematician's Dream

- **Hilbert's Program** (1920s): Can we verify all mathematics mechanically?
- **Bourbaki** (1935+): Rigorous, unified foundations

The Mathematician's Dream

- **Hilbert's Program** (1920s): Can we verify all mathematics mechanically?
- **Bourbaki** (1935+): Rigorous, unified foundations
- **QED Manifesto** (1994): A single repository of all math

The Mathematician's Dream

- **Hilbert's Program** (1920s): Can we verify all mathematics mechanically?
- **Bourbaki** (1935+): Rigorous, unified foundations
- **QED Manifesto** (1994): A single repository of all math

Question. What happened?

Another step back

Another step back

Formalization. = translate informal argument → formal proof

Another step back

Formalization. = translate informal argument → formal proof

- **Informal.**
  Corollary 2.3 Let c be a number and let the assumptions be as in the theorem. Then

$$\lim_{x \to a} cf(x) = c \cdot L.$$

Another step back

Formalization. = translate informal argument → formal proof

- **Informal.**
  Corollary 2.3 Let c be a number and let the assumptions be as in the theorem. Then

$$\lim_{x \to a} cf(x) = c \cdot L.$$

Proof.

Another step back

Formalization. = translate informal argument → formal proof

- **Informal.**

  Corollary 2.3 Let c be a number and let the assumptions be as in the theorem. Then

  $$\lim_{x \to a} cf(x) = c \cdot L.$$

  Proof. Clear.

Another step back

Formalization. = translate informal argument → formal proof

- **Informal.**
  Corollary 2.3 Let c be a number and let the assumptions be as in the theorem. Then

$$\lim_{x \to a} cf(x) = c \cdot L.$$

Proof. Clear. Just kidding.

Another step back

Formalization. = translate informal argument → formal proof

- **Informal.**
  Corollary 2.3 Let c be a number and let the assumptions be as in the theorem. Then

  $$\lim_{x\to a} cf(x) = c \cdot L.$$

  Proof. Clear. Just kidding. You need two *ingredients*:
  1. $\forall \varepsilon > 0 \; \exists \delta > 0 \; \forall x, \; 0 < |x - a| < \delta \implies |f(x) - L| < \varepsilon,$
  2. $|cf(x) - cL| = |c| \, |f(x) - L|.$

  Then applying 2. to 1. gives the result.

Observations

A proof could to be portrayed as a *dependency structure*.

Observations

A proof could to be portrayed as a *dependency structure*.

- "By Lemma 2 and Theorem 1..." $\sim$ Lemma 2 + Theorem 1

Observations

A proof could to be portrayed as a *dependency structure*.

- "By Lemma 2 and Theorem 1..." $\sim$ Lemma 2 + Theorem 1
- "From the above..." $\sim$ Lemma 2 + Theorem 1 $\rightarrow$ Main result

Observations

A proof could to be portrayed as a *dependency structure*.

- "By Lemma 2 and Theorem 1..." $\sim$ Lemma 2 + Theorem 1
- "From the above..." $\sim$ Lemma 2 + Theorem 1 $\rightarrow$ Main result

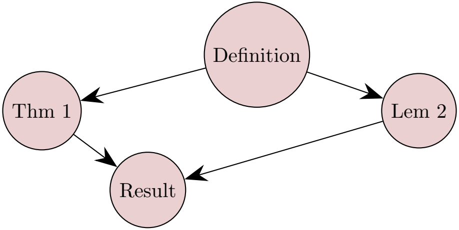
That is,

Observations

A proof could to be portrayed as a *dependency structure.*

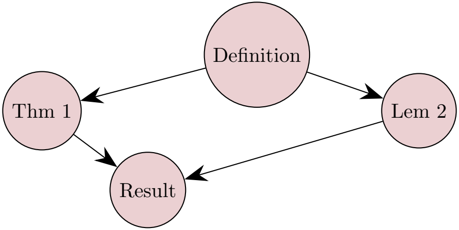
- "By Lemma 2 and Theorem 1..." $\sim$ Lemma 2 + Theorem 1
- "From the above..." $\sim$ Lemma 2 + Theorem 1 $\rightarrow$ Main result

That is,



The Proof DAG

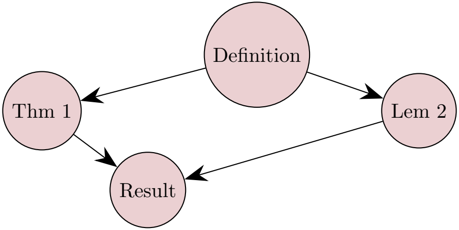
This is a graph where



- Statements, lemmas, theorems are **nodes**

The Proof DAG

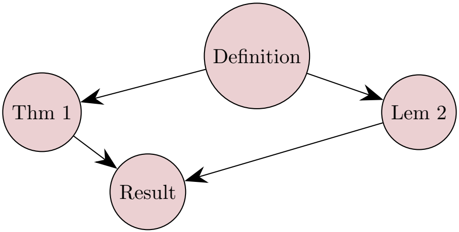
This is a graph where



- Statements, lemmas, theorems are **nodes**
- **Edges** signify ogical dependency

The Proof DAG

This is a graph where



- Statements, lemmas, theorems are **nodes**
- **Edges** signify ogical dependency

It is also **Acyclic.** This is called a *Directed Acyclic Graph* (DAG).

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

whereas standard formal proofs

- have $10\times$ the number of nodes,

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

whereas standard formal proofs

- have $10\times$ the number of nodes,
- are complete, but verbose.

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

whereas standard formal proofs

- have $10\times$ the number of nodes,
- are complete, but verbose.

INSIGHT. An informal proof is a lossy compression of its DAG.

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

whereas standard formal proofs

- have $10\times$ the number of nodes,
- are complete, but verbose.

Insight. An informal proof is a lossy compression of its DAG.
Problem. One often compromises completeness or readability.

The Compression Problem

Informal proofs *compress* the DAG. In such a proof, one finds

- Key ideas only
- Some hand-waving *Follows from standard...* arguments

whereas standard formal proofs

- have 10× the number of nodes,
- are complete, but verbose.

Insight. An informal proof is a lossy compression of its DAG.
Problem. One often compromises completeness or readability.
Question. Can we *reliably* recover the DAG from a compressded proof?

What happened?

QED Manifesto:

- create tens of proof assistants

What happened?

QED Manifesto:

- create tens of proof assistants
- hope the community will verify all proofs

What happened?

QED Manifesto:

- create tens of proof assistants
- hope the community will verify all proofs

Recent shift: Buzzard, Scholze, and the *Lean Community*

- Formalize contemporary proofs (FLT, Liquid Tensor Experiment)

What happened?

QED Manifesto:

- create tens of proof assistants
- hope the community will verify all proofs

Recent shift: Buzzard, Scholze, and the *Lean Community*

- Formalize contemporary proofs (FLT, Liquid Tensor Experiment)
- use that as an excuse to build libraries

What happened?

QED Manifesto:

- create tens of proof assistants
- hope the community will verify all proofs

Recent shift: Buzzard, Scholze, and the *Lean Community*

- Formalize contemporary proofs (FLT, Liquid Tensor Experiment)
- use that as an excuse to build libraries

$\implies$ still not enough.

Autoformalization

Autoformalization. Automatic formalization.

The mathematician's dream is

1. **Input.** An informal proof (lossy text)

Autoformalization

Autoformalization. Automatic formalization.

The mathematician's dream is

1. **Input.** An informal proof (lossy text)
2. **Process.** Decompress the DAG structure

Autoformalization

Autoformalization. Automatic formalization.

The mathematician's dream is

1. **Input.** An informal proof (lossy text)
2. **Process.** Decompress the DAG structure
3. **Output.** A formal, verifiable proof

Autoformalization

Autoformalization. Automatic formalization.

The mathematician's dream is

1. **Input.** An informal proof (lossy text)
2. **Process.** Decompress the DAG structure
3. **Output.** A formal, verifiable proof

Autoformalization

Autoformalization. Automatic formalization.

The mathematician's dream is

1. **Input.** An informal proof (lossy text)
2. **Process.** Decompress the DAG structure
3. **Output.** A formal, verifiable proof

This is graph reconstruction from partial information.
Should be possible, right?

Right?

Given a compressed proof, one has to infer

Right?

Given a compressed proof, one has to infer

- **Nodes.** What lemmas aren't mentioned?
  - *"The proof of Theorem 5 relies on an unlisted lemma from the appendix"*

Right?

Given a compressed proof, one has to infer

- **Nodes.** What lemmas aren't mentioned?
  - *"The proof of Theorem 5 relies on an unlisted lemma from the appendix"*
- **Edges.** Which lemmas depend on which?
  - *"Lemma 3 actually uses Lemma 1, even though it's not mentioned"*

Right?

Given a compressed proof, one has to infer

- **Nodes.** What lemmas aren't mentioned?
  - *"The proof of Theorem 5 relies on an unlisted lemma from the appendix"*
- **Edges.** Which lemmas depend on which?
  - *"Lemma 3 actually uses Lemma 1, even though it's not mentioned"*
- **Subgraphs.** What standard techniques are invoked?
  - *"By continuity and the intermediate value theorem..."*

Right?

Given a compressed proof, one has to infer

- **Nodes.** What lemmas aren't mentioned?
  - *"The proof of Theorem 5 relies on an unlisted lemma from the appendix"*
- **Edges.** Which lemmas depend on which?
  - *"Lemma 3 actually uses Lemma 1, even though it's not mentioned"*
- **Subgraphs.** What standard techniques are invoked?
  - *"By continuity and the intermediate value theorem..."*

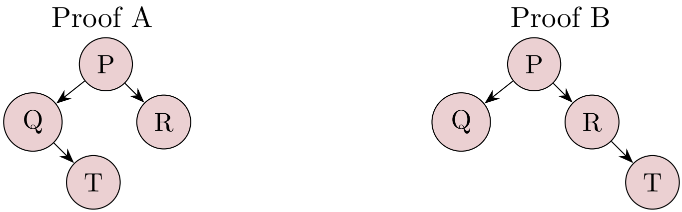
What could possibly go wrong?

Here's the catch

A single informal proof text can correspond to

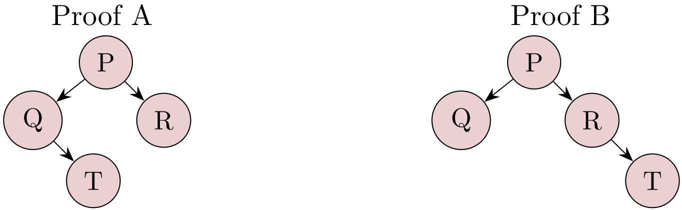
Here's the catch

A single informal proof text can correspond to



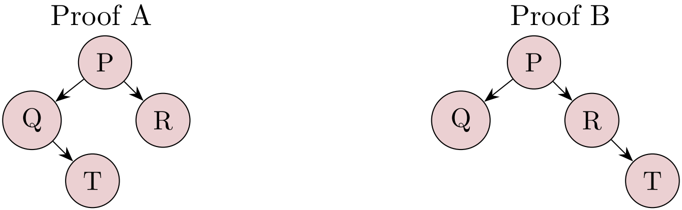
Here's the catch

A single informal proof text can correspond to



Here's the catch

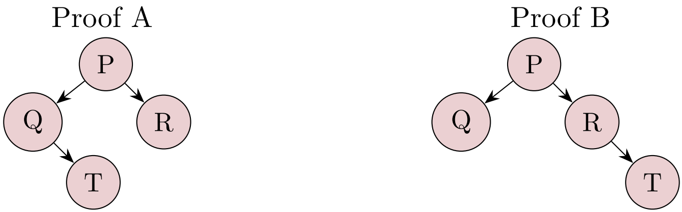
A single informal proof text can correspond to



Both are valid proofs, and lead to the same written text (maybe).

Here's the catch

A single informal proof text can correspond to

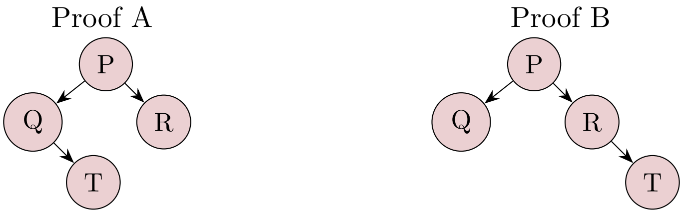


Both are valid proofs, and lead to the same written text (maybe).

Insight. Decompression is not unique

Here's the catch

A single informal proof text can correspond to



Both are valid proofs, and lead to the same written text (maybe).

Insight. Decompression is not unique

Takeaway. The decompression problem is *ill-posed*!

A Semantic Gap

The statement

"By a standard argument, (5) follows from (3) and (4)."

A Semantic Gap

The statement

"By a standard argument, (5) follows from (3) and (4)."

is semantically ambiguous. The writer might refer to

- direct calculation

A Semantic Gap

The statement

"By a standard argument, (5) follows from (3) and (4)."

is semantically ambiguous. The writer might refer to

- direct calculation
- a 3-page lemma from another paper

A Semantic Gap

The statement

> *"By a standard argument, (5) follows from (3) and (4)."*

is semantically ambiguous. The writer might refer to

- direct calculation
- a 3-page lemma from another paper
- a highly nontrivial argument to the reader

The Landscape

Here's a glimpse of the research frontier,

The Landscape

Here's a glimpse of the research frontier,

- **M2F** (Math-to-Formal): Build skeleton DAGs from text
    - Identify main statements and structure

The Landscape

Here's a glimpse of the research frontier,

- **M2F** (Math-to-Formal): Build skeleton DAGs from text
  - Identify main statements and structure
- **LeanAgent**: Expand and verify the DAG
  - Given a partial proof, fill in missing steps

The Landscape

Here's a glimpse of the research frontier,

- **M2F** (Math-to-Formal): Build skeleton DAGs from text
  - Identify main statements and structure
- **LeanAgent**: Expand and verify the DAG
  - Given a partial proof, fill in missing steps

Observation. Approximations to the reconstruction problem.

What's Next?

A (partial) solution to the compression problem has implications, possibly

- Better structured paper formats?
- Hybrid human-AI verification?
- Standardizing common subgraphs?
- Use formalization as a writing tool
- Get feedback on sketches
- Collaborate with machines

Consequence. AI will replace (a subset of) Mathematicians (?)

A Proposal

Effectively, we have a missing value prediction task. We propose a learning project titled

Masked Language Modeling for Graphs

- Input: A DAG with *some nodes & edges removed*
- Task: Predict what was masked
- Tool: Neural networks, language models

Strategies:

- Mask nodes, predict them from the graph structure
- LLMs: "Fill in the missing lemma that should go here"
- Graph neural networks: Learn from examples of complete proofs

Let's talk!

Bibliography

- D. Hilbert, *Mathematical Problems*, Proc. Int. Congress of Mathematicians, 1900.
- QED Manifesto, *Towards Computer-Aided Mathematics*, 1994.
- M2F: Math-to-Formal Translation, *arXiv preprint*, 2026.
- LeanAgent: Lifelong Learning for Theorem Proving, *arXiv preprint*, 2024.

Acknowledgements

Many thanks to

- Ahmed Maghri for the warm invitation
- Pablo Santos Guerrero for the fruitful discussions
- the MS@CU!

That's it honestly

Questions?