Naproche Natural Language Formalizations with LLMs

Peter Koepke[1] and Joshua Wirtz[1]

Mathematical Institute, University of Bonn, Germany
koepke@math.uni-bonn.de
https://naproche.github.io

Introduction. LLMs and LLM-based systems excel in producing computer code. Strong similarities between programming and writing formal mathematical proofs motivate the deployment of LLMs in interactive theorem proving (ITP), either assisting human experts, or in unsupervised autoformalizations [14]. In our note we describe formalization experiments with LLMs in the Naproche natural language proof system.

The Naproche **Na**tural **Pro**of **Che**cker [10, 4] aims to make ITP natural and close to mathematical practice. Naproche uses the controlled natural language ForTheL (**For**mula **The**ory **L**anguage) [12] as its input language, and strong automated theorem proving (ATP) by E [13] or Vampire [1] to enable natural proof granularities. Typeset texts in the LaTeX dialect of ForTheL are immediately readable by mathematicians. Naproche formalizations have been carried out on a variety of mathematical theories [3, 6, 7], based on classical logic and set theory.

Naproche formalization can be seen as translations of unrestricted mathematical texts into the simplified ForTheL language. As LLMs are trained in natural language translations, they seem predestined for this task. In our project we start with a chat with ChatGPT [11] to generate a proof of Euclid's lemma in Naproche. In subsequent experiments, the human expert is replaced by LLM-based agentic systems. Our mathematical examples concern simple theorems from F. Wiedijk's list [15]. The use of ForTheL as a naturally readable formalization language adds interesting features to the formalization enterprise which we mention in our final remarks.

There are related autoformalization projects around Naproche like [8].

Chatting About Elementary Number Theory. Euclid's lemma, which is omnipresent in elementary number theory, states: *if a prime number p devides the product $a * b$ of natural numbers then p divides a or p divides b* (see the Wikipedia article [2]). In a chat with the free version of ChatGPT (GPT-5 Thinking) the LLM produced a translation from a Wikipedia proof of Euclid's lemma to Naproche. Starting from a small Naproche axiomatization `01.ftl.tex` about the addition and multiplication of natural numbers we requested to correct or extend the already existing file `[n].ftl.tex`, like:

Human: *Please extend the file by a new subsection on the nonstrict order on the natural numbers.*

Human: *Could you translate the following proof from Wikipedia into a Naproche lemma?*

After many hours the proof in `98.ftl.tex` was finally accepted by Naproche and also by LaTeX. The first few lines of the readable LLM-generated proof read:

Proof (by induction on $(a * b)$) Let a, b, n be natural numbers.

Suppose n divides $(a*b)$ and a and n are coprime. Take a natural number q such that $(n*q) = (a*b)$. Suppose $0 < a$ and $0 < b$ and $0 < n$ and $0 < q$.

Case $n = a$. Then a and a are coprime. Hence $a = 1$. Therefore n divides b. End.

Agentic Autoformalization of Mathematical Theories. Next we sought to automate the chat experiment using the VSCode integrated development environment (IDE) [9] with the GitHub Copilot Chat extension [5] and GPT-5.3-Codex with reasoning effort set to Xhigh in agent mode. After preparing the local environment we issued the following prompt:

Human: *Formalize Euclid's lemma in Naproche. The formalization must follow the proof by induction from Wikipedia. The formalization must be self-contained. The formalization must be written to a* *single file named euclids_lemma.ftl.tex. The formalization must successfully verify when running the command './verify.sh'.*

After approximately 9 minutes, the agent produced a verified ForTheL file, `euclids_lemma.ftl.tex`. While inspecting the formalization confirmed that it is self-contained and superficially follows the inductive proof from Wikipedia, we identified significant logical flaws. The formalization over-relies on axioms, often axiomatizing statements that could be deduced from prior ones. Furthermore, the following axiom is introduced that is false in the standard model:

Axiom. *Assume $a < n$ and $n * q = a * b$ and b is positive. Then $n - a | a * (b - q)$ and $b - q$ is positive and $a * (b - q) < a * b$.*

Luckily, this is not exploited in the further argument.

Autoformalization of Proofs within Given Theories. To prevent LLMs to invent their own axioms we then restricted the agentic system to only supply formal proofs within given axiomatic theories. In an existing Naproche formalization of the Cantor-Schröder-Bernstein theorem from set theory, we removed the formal proof and tasked the agent with:

Human: *Fill in the missing proof inside the Cantor-Schroeder-Bernstein formalization at archive/articles/source/cantor-schroeder-bernstein.ftl.en.tex. The proof must successfully verify when running the command './verify.sh'.*

After a runtime of approximately 12 minutes, the agent successfully filled in the missing proof. Inspecting the file confirmed that the surrounding formalization remained unmodified. The generated proof is well-structured and effectively utilizes the provided Knaster-Tarski fixed-point theorem to define the desired bijection:

Proof. *.... Consider an injective map f from x to y and an injective map g from y to x.*
Define $H(u) := x \setminus g[y \setminus f[u]]$ for $u \in \mathcal{P}(x)$. ...
H has a fixed point. .... Consider a fixed point X of H. .... Define

$$h(a) := \begin{cases} f(a) & : a \in X \\ g^{-1}(a) & : a \notin X \end{cases}$$

for $a \in x$. .... Then h is a bijection between x and y.

Remarks.

– Current LLMs can assist in NaProChe formalizations of non-trivial mathematics. Agentic systems can achieve fully automatic formalizations.

– The agentic approach can automatically take care of technical details like correct ForTheL syntax, LaTeX typesetting and Naproche proof checking.

– Human supervision is still essential to ensure mathematically reasonable formalizations.

– ForTheL provides an intermediate language layer which can be directly understood by humans and processed by the NaProChe proof assistant. This is especially valuable for evaluating generated proofs and for error detection by human experts.

– We suspect that Naproche's declarative proofs are better aligned with LLM reasoning than tactic-style proofs, due to their closer resemblance to informal mathematical discourse.

– Beyond proof certification, NaProChe autoformalizations may offer additional utility by producing mathematical texts that can serve as practical resources for students and researchers.

References

[1] About Vampire. URL: https://vprover.github.io.

[2] Wikipedia article: Euclid's lemma. URL: https://en.wikipedia.org/wiki/Euclid%27s_lemma.

[3] Adrian De Lon, Peter Koepke, and Anton Lorenzen. A natural formalization of the mutilated checkerboard problem in Naproche. In Liron Cohen and Cezary Kaliszyk, editors, *12th International Conference on Interactive Theorem Proving (ITP 2021)*, volume 193 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:11, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[4] Adrian De Lon, Peter Koepke, Anton Lorenzen, Adrian Marti, Marcel Schütz, and Makarius Wenzel. The Isabelle/Naproche natural language proof assistant. In André Platzer and Geoff Sutcliffe, editors, *Automated Deduction – CADE 28*, pages 614–624. Springer International Publishing, 2021.

[5] GitHub. Github copilot chat - visual studio marketplace, 2026. URL: https://marketplace.visualstudio.com/items?itemName=GitHub.copilot-chat.

[6] Peter Koepke. A natural language formalization of perfectoid rings in Naproche. Recording: https://www.youtube.com/watch?v=Babo06m2KyE, 2024. Talk at the *Prospects of Formal Mathematics* trimester program, Bonn, Germany.

[7] Peter Koepke and Patrick Schäfer. Formalizing the Solow model in Naproche. In Valeria de Paiva and Peter Koepke, editors, *Intelligent Computer Mathematics*, pages 357–370, Cham, 2025. Springer.

[8] Atle Hahn Peter Koepke Josef Urban Mario Carneiro, Adrian De Lon. Le miz s'approche: Informalization and autoformalization with mizar and naproche. Contribution to the informal proceedings of AITP 2025, 2025.

[9] Microsoft. Visual studio code, 2026. URL: https://code.visualstudio.com.

[10] Naproche Group. Naproche. URL: https://naproche.github.io.

[11] OpenAI. Chatgpt. URL: https://chatgpt.com.

[12] Andrei Paskevich. The syntax and semantics of the ForTheL language, 2007. URL: http://nevidal.org/download/forthel.pdf.

[13] Stephan Schulz. The E Theorem Prover. URL: https://wwwlehre.dhbw-stuttgart.de/~sschulz/E/E.html.

[14] Ke Weng, Lun Du, Sirui Li, Wangyue Lu, Haozhe Sun, Hengyu Liu, and Tiancheng Zhang. Autoformalization in the era of large language models: A survey. *arXiv*, (2505.23486), 2025. URL: https://arxiv.org/abs/2505.23486.

[15] Freek Wiedijk. Formalizing 100 Theorems. URL: https://www.cs.ru.nl/~freek/100/.