How LLMs could Fool a Proof Checker: The Risks of Inconsistent Assumptions in Automated Proof Systems

Alberto Gandolfi[1][0000−0001−6956−7513]

NYU Abu Dhabi ag189@nyu.edu

The integration of large language models (LLMs) with proof assistants promises to enhance mathematical reasoning, blending intuitive human interaction with rigorous verification [1,**?**,**?**]. However, this coupling might introduce subtle risks. In this talk, we investigate how LLMs, proof checkers, and proof assistants handle inconsistent assumptions: sets of assumptions that can lead to seemingly valid but ultimately flawed proofs. Our focus is on GPT-o1 as an LLM and Isabelle and Coq as proof assistants.

We use the case study of Minimal Probability, defined as a theory of statements about expectations of a finite number of random variables. This theory is rich enough to cover a variety of simple applications and exercises, including known elementary paradoxes. Probability theory offers advantages for this study due to its status as a formal theory connected to simple, concrete situations and its susceptibility to subtle inconsistencies involving probabilities, expectations, conditional probabilities, and independence.

Key findings from our investigation into the behavior of LLMs and proof assistants when faced with inconsistent assumptions include:

- When the possibility of inconsistency is not mentioned, GPT-o1 often detects simple inconsistencies but fails in more complex cases.
- When the possibility of inconsistency is raised in the prompt, GPT-o1 decides about inconsistencies at a level comparable to human experts.
- Proof checkers (e.g., Isabelle and Coq used solely to verify a submitted proof) do not detect contradictions, even when apparent.
- Proof assistants (e.g., Isabelle and Coq with active intervention on justifying proof steps) might flag inconsistencies, but only occasionally.
- GPT-o1 can be instructed to deceive the proof checker: it can create an inconsistent set of assumptions, opposite conclusions, and the proof checker code in such a way that both Isabelle and Coq would certify either conclusion as correct.

These findings highlight how potential inconsistencies could infiltrate probabilistic modeling of seemingly simple situations, leading LLMs and proof assistants to produce seemingly valid but ultimately meaningless proofs. Historically, when experienced humans designed setups, such risks were remote. However, these risks could grow as machines take on more control and assumptions become increasingly complex. Although possibly remote, the mere possibility that an LLM could deliberately craft assumptions to gain certification of unwarranted conclusions is particularly concerning.

This suggests the need for potential remedies or at least mitigating strategies. Our proposed actions include:

(0) Raising awareness of the issue to direct efforts toward mitigating the potential dangers.
(1) Maintaining human supervision of assumptions before integrating them into the proof environment for as long as possible.
(2) Using multiple proof assistants—built on diverse logical foundations—to increase the likelihood of detecting contradictions.
(3) Integrating automated consistency checks or specialized consistency-verification tools to promptly flag contradictions.

We explore (2) in some examples with some moderate success.

As for (3), we consider Minimal Probability, and we show that its axiomatization in a semantically complete first-order theory allows for the algebraization of assumptions, transforming probabilistic problems into polynomial relations. Consistency is then equivalent to the nonemptiness of a semi-algebraic set, a decidable problem analyzable using tools like cylindrical decomposition [4], which fully classifies consistent

and inconsistent assumptions. This procedure has $\exists\mathbb{R}$ complexity, and we implemented it in Mathematica for the considered examples.

The talk will present examples, tests, and mitigating actions. Here is a summary:

	Direct flag, or proof rejection Query A			Flag for proof of Query B in same file or chat			Flag for Query B proof in separate file or chat			Detection of inconsistent assumptions after prompt			Detection of inconsistent assumptions by cylindrical decomposition (with execution times in msec.)
	GPT	I	C	GPT	I	C	GPT	I	C	GPT	I	C	Mathematica
Ex. 1 (n=1)	Y	N	N	N/A	N	N	Y	N	N	N/A	N	Y	Y (1.2)
Ex. 2 (n=2)	Y	N	N	N/A	N	N	Y	N	N	N/A	N	Y*	Y (0.9)
Ex. 3 (n=2)	N	N	N	Y	N	N	N	N	N	Y	N	N	Y (2.6)
Ex. 4 (n=3)	N	N	N	Y	N*	N	N	N	N	Y	N	N	Y (24.5)
Ex. 5 (n=4)	N	N	N	Y	N*	N	Y	N	N	N/A	N	N	Y (43.6)
Ex. 6 (n=5)	N/Y*	N	N	N/Y*	N*	N	N/Y*	N	N	Y	N	N	N ($> 10^6$)
Ex. 6' (n=1)													Y (1.2)
Ex. 12 (n=2)	N/A	N	N	N/A	N	N	N/A	N	N	N/A	N	Y*	Y (2.9)
Ex. 13 (n=2)	Y	N	N	N/A	N	N	Y	N	N	N/A	N	Y	Y (2.3)

Table 1. Summary of Flagging and Detection Across Examples by GPT-o1 (GPT), Isabelle (I), and Coq (C) in the six inconsistent examples when asked to derive one consequence in the first proof, and a contrasting one in the second proof. The second line of Ex. 6 uses reduced variables, Ex. 12 was created by GPT. Ex. 13 is an extreme example of possible deceiving. Examples 7 to 11 are consistent examples which are used separately to investigate GPT's ability to avoid false positives when prompted to detect a contradiction. The last column indicates the performance with execution times of Cylindrical Decomposition in Mathematica.
Y stands for Yes, the inconsistency has been flagged, N for No, N/A for not applicable.
A star indicates that the performance changes if the request is to find more complex proof steps, or the proof structure is outlined and only simple steps are to be filled in.
n indicates the number of involved basic events or variables.

References

1. Tao, Terence. "Machine assisted proof." Notices of the American Mathematical Society, to appear (2024).
2. Shulman, Michael. "Strange new universes: Proof assistants and synthetic foundations." Bulletin of the American Mathematical Society 61.2 (2024): 257-270.
3. Song Peiyang, Kaiyu Yang, and Anima Anandkumar. "Towards large language models as copilots for theorem proving in lean." arXiv preprint arXiv:2404.12534 (2024).
4. Bochnak, Jacek; Coste, Michel; Roy, Marie-Françoise. Real Algebraic Geometry. Translated from the 1987 French original. Revised by the authors. Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)], 36. Springer-Verlag, Berlin, 1998.