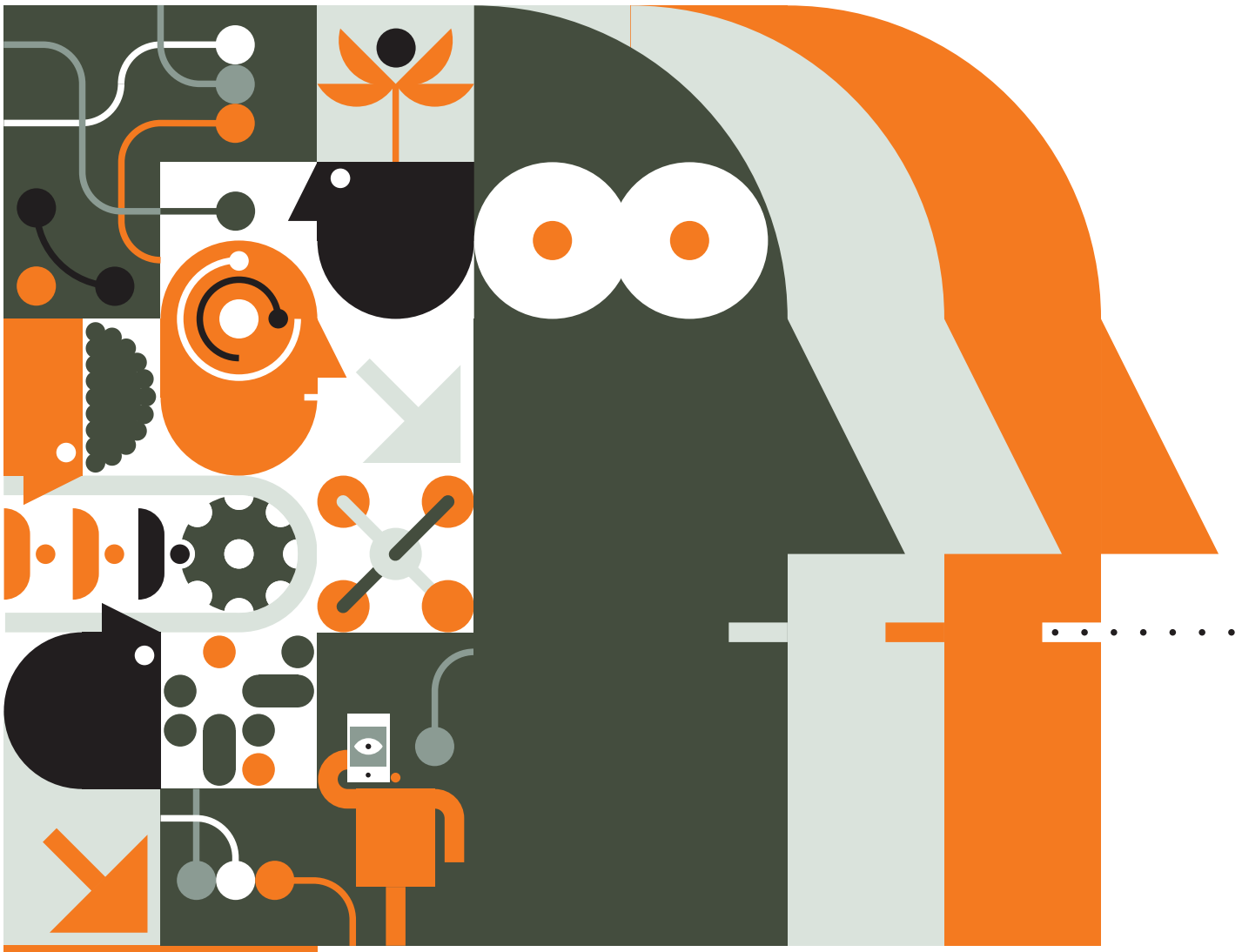
COMMUNICATIONS OF THE ACM

CACM.ACM.ORG

03/2025 VOL.68 NO.03



# Large Language Model Use in Crowd Work

The AI Alignment Paradox

AI as Catalyst for Biodiversity Understanding

The Sustainability Gap for Computing: Quo Vadis?

Association for Computing Machinery

a new underpinning that was far easier to implement.

In Li's work with colleagues on a formalized version of another textbook on number theory, the group found that a formal definition of a key type of function of complex numbers would be better expressed by the use of an arithmetic expansion, rather than the more-intuitive approach used in the informal source. That approach would help underpin a larger set of dependent proofs, he explained. In some projects, these considerations have led to changes to underlying definitions, sometimes repeatedly.

"Over time, it becomes increasingly difficult to refactor definitions or lemmas due to growing dependencies," Li said. "AI could potentially assist with this, much like other code-assistance tools, such as Copilot."

Other goals may provide targets that AI can serve more easily than auto formalization itself. Patrick Massot, professor of mathematics at France's University of Paris-Saclay, argues one lingering issue with the current formal languages like Coq and Lean is they are too opaque to non-computer scientists. An "informalizer" would help scholars read the verified proofs. It could, as a byproduct, provide the foundation for building interactive math textbooks, in which students are able to drill down into the background of any proof or lemma they see.

A couple of teams have used LLMs to try to do the work. Though LLMs make fewer mistakes here than in formalization, the task demands much higher accuracy than they can deliver. In his work on informalization with Massot, Kyle Miller, assistant professor at the University of California at Santa Cruz, has been exploring the use of more traditional symbolic AI techniques. This involves far more manual engineering than training an LLM. Simply mapping the grammar of a language like Lean into English is not enough; it needs more changes. For example, the code created to check a proof formally can contain a lot of repetition that the tool would ideally remove from the human-readable version.

If successful, the work on informalizers in turn may help close the loop for theorem-proving engines based on LLMs and similar technologies. The output from these tools would provide a rich resource of synthetic data that can be used to retrain the AI engines as they create new proofs.

**Patrick Massot argues one lingering issue with current formal languages like Coq and Lean is that they are too opaque to non-scientists.**

The multiple feedback loops that are now appearing may mean the logjam that has held up one of computer science's major applications is finally breaking. But it may take a lot more research into hybrid schemes to strengthen AI's reasoning skills and, perhaps, finally make autoformalization work for mathematicians.

### Further Reading


*Avigad, J.*
Mathematics and the Formal Turn; *Bulletin of the American Mathematical Society*. 61 (2024), 225-240

*Li, Zhaoyu, Sun, J., Murphy, L., Su, Q., Li, Zenan, Zhang, X., Yang, K., and Si Xujie*
A Survey on Deep Learning for Theorem Proving; arXiv:2404.09939 (2024)

*Karatarakis, M.*
Leveraging Large Language Models for Autoformalizing Theorems: A Case Study; Ninth Conference on Artificial Intelligence and Theorem Proving (AITP 2024)

*Tarrach, G., Jiang, A.Q., Raggi, D., Li, W., and Jamnik, M.*
More Details, Please: Improving Autoformalization with More Detailed Proofs; AI for MATH Workshop at the International Conference on Machine Learning (ICML 2024)

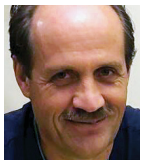
*McCarthy, M.*
Computer Programs for Checking Mathematical Proofs; *Proceedings of the Fifth Symposium in Pure Mathematics of the American Mathematical Society*, pages 219–227 (1961)


**Chris Edwards** is a Surrey, U.K.-based writer who reports on electronics, IT, and synthetic biology.



# ACM Member News

## COMPUTING AT THE EDGE



**Daniel Grosu is a professor of computer science in the College of Engineering at Wayne State University in Detroit, MI.**

**Grosu earned his undergraduate degree in automatic control from Romania's Technical University of Iasi. He went on to obtain both his master's and doctoral degrees in computer science from the University of Texas at San Antonio.**

**After receiving his Ph.D. in 2003, Grosu joined the faculty at Wayne State University, where he has remained.**

**Grosu's research centers on cloud and edge computing, parallel and distributed algorithms, graph algorithms, approximation algorithms, scheduling and load balancing, and topics at the border of computer science, game theory, and economics.**

**"I am currently focused on designing resource- and task-allocation algorithms for edge and cloud computing," Grosu said. He added that these algorithms take into account the mobility of users, the capabilities of available resources, the data that is needed, and then make the best decisions to allocate these tasks and resources.**

**Grosu notes monetizing and pricing these resources in edge computing is an issue he is endeavoring to solve. "Basically, you have to design pricing mechanisms to accurately price the resources in edge computing so that a provider earns a profit, but at the same time the users get a fair price as well," he explained.**

**Grosu believes there is still a lot of work to be done in edge computing as deployments increase, especially with the emergence of autonomous vehicles.**

**"They'll need a lot of infrastructure support from the edge," Grosu concluded.**

***—John Delaney***