

---

**From the president**

Arthur Rosendahl

Many *TUGboat* readers will no doubt have heard of it by now, but I would be remiss if I didn't mention Don Knuth's *Claude's Cycles*<sup>1</sup> where he describes how a large language model (LLM) was able to prove a conjecture he had formulated recently. It starts with a bang:

Shock! Shock! I learned yesterday that an open problem I'd been working on for several weeks had just been solved by Claude Opus 4.6 — Anthropic's hybrid reasoning model that had been released three weeks earlier! It seems that I'll have to revise my opinions about “generative AI” one of these days. What a joy it is to learn not only that my conjecture has a nice solution but also to celebrate this dramatic advance in automatic deduction and creative problem solving.

and goes on to describe carefully the steps taken, with addenda at the end about similar approaches using other LLMs.

This is indeed remarkable news, but what struck me most was not to learn that an LLM was able of such reasoning — although I had no idea until then — but that Knuth had taken the time to write a 6-page paper about it and revise it a couple of times. As soon as he learned something new, he wanted to share it with the world.

This cannot always be taken for granted. On a job interview a few years ago, I was explaining about a project I had worked on with a collaborator from a different field, and the interviewer interrupted me, seemingly surprised — and pleased — about hearing me say that I explain things to colleagues. I thought the story I was telling was quite the opposite: I was just doing my job, and the person doing the explaining was my colleague, who was the teacher on that occasion; but that was evidently not how it came across. The interviewer then opined that I would feel at home in the organisation I was applying to, because they also had a “sharing culture”. Eventually I was offered the position, took it, and did not learn much from coworkers in that place. With the benefit of hindsight, there were signs.

It takes time and skill to share knowledge, so it is perhaps no surprise that some organisations fall short. Here is another story: last autumn, I became fascinated with a Rubik's Cube that had emerged

---

<sup>1</sup> Available at [cs.stanford.edu/~knuth/papers/claude-cycles.pdf](https://cs.stanford.edu/~knuth/papers/claude-cycles.pdf). 28 February 2026, last revised 16 March 2026.

from a box after we moved houses, and was learning how to solve it. I came across a video, clearly aimed at beginners, explaining how to solve one particular step. It essentially showed a few tricks of the trade, but concluded with the surprising sendoff “and don't train . . . unless you suck”.

This was unexpected. Surely training is (almost) always a good thing, and beginners should be told to do it? I happen to think so. But regardless, the intention was there, and the final words might have been a simple slip of the tongue. In any case, lest it should be thought that the *cubing* community is riddled with bad and slightly rude teachers, I need to mention too that I did find other resources that were very good, such as a document<sup>2</sup> detailing another step of the solution, written with great care and humility, and containing actual encouragements and sound advice (“do train”). Most important, the author had managed to capture the essence of what he wanted to convey: whereas most Rubik's Cube solutions consist in a long list — 41 for this particular step — of formulae, usually called “algorithms” by the community, the document reduced all the possibilities to only three cases, and explained everything with words and a few figures. In doing so, the author not only wrote a very didactic piece but also restored the original meaning of the word algorithm, since that is what is described in the document: a single algorithm with a few branches. (At this point, I will leave the reader to wonder what was *actually* more important to me, good pedagogy or correct vocabulary.)

Great teachers know how to simplify difficult concepts for the benefit of their audience. We are privileged to count many of them in our little world of  $\text{\TeX}$ , starting with the first among us, who it seems will never stop writing. As a matter of fact, he did end the latest revision of *Claude's Cycles* with a plea to please leave him to work on *The Art of Computer Programming* in peace, so that gives me the last word: GO FORTH now and spread the knowledge that you have!

◇ Arthur Rosendahl  
Uppsala University  
TUG president  
`president (at) tug (dot) org`

---

<sup>2</sup> [docs.google.com/document/d/1npPMcy2Vv-jjUB7wr0gHE0rESuwUgba\\_D7vT20UzvTQ](https://docs.google.com/document/d/1npPMcy2Vv-jjUB7wr0gHE0rESuwUgba_D7vT20UzvTQ). For those interested, this solves the second step of the classic *speedcubing* method called CFOP. Beginners who would like to try their hand at it will thus need to solve the first step . . . but unfortunately one of the very few available presentations is the aforementioned video. It can, however, arguably be solved intuitively.