logic-ex

1 Teaching logic usually involves software for automated marking and tracking students' progress. At Warwick we've been using the leading software package for nine years. It has many virtues, but it's quite clunky, it hasn't been significantly updated and it costs each student around £45, which many object to paying.

2 Partly because of resistance to the high costs and partly because students complain about using the software package, I've been searching for alternative software for my course for a while. Several people, including the philosopher David Velleman, have had a go, but nothing remotely comparable to the leading software package is available.

3 So I wrote some software myself, a web application for logic teaching. It is currently deployed and ready to use for my course with around three hundred students and six sessional teachers.

4 I'm reasonably confident that the new software will provide a better experience for the students, and that it will improve the way we teach logic, meaning we can raise standards.

My software is already better than the leading software package in many respects, and with more work it could be better in every respect.

5 I want this software to be what practically everyone uses for teaching logic anywhere in the world.

I want to improve how logic is taught by providing software that encourages students to experiment, harnesses their willingness to help each other, and channels informative feedback on their mistakes.

I also want to make teaching university-level logic easy enough that more of it can be done outside university, whether in schools or through self-teaching.

I want more people to learn more logic than they would have done without the software.

6 Achieving this involves some challenges: we need to market the software, build a community around it and keep improving it.

7 How can we get the resources we need to market the software, build a community around it, continue developing it and guarantee its survival?

- How can we promote rapid, widespread adoption?
- Should we focus on students (like Khan Academy) or lecturers (as the leading software does) or both?
- What is the best way to gain the sort of trust needed for a lecturer to switch to using the software, which is a big, multi-year commitment on her part?

- What sort of community involvement should we aim for?
- What are reasonable short-term aims?