Mechanized Mathematics

Jacques Carette[1]

[1]Department of Computing and Software, McMaster University
www.cas.mcmaster.ca/~carette

Abstract. If one were designing an entirely new mathematical assistant, what might it look like? Problems and some favoured solutions are presented.

In the 50 years since McCarthy's " Recursive Functions of Symbolic Expressions and Their Computation by Machine", what have we learned about the realization of Leibniz's dream of just being able to utter "Calculemus!"[1] when faced with a mathematical dilemma?

In this talk, I will first present what I see as the most important lessons from the past which need to be heeded by modern designers. From the present, I will look at the context in which computers are used, and derive further requirements. In particular, now that computers are no longer the exclusive playground for highly educated scientists, usability is now more important than ever, and justifiably so.

I will also examine what I see as some principal failings of current systems, primarily to understand some major mistakes to avoid. These failings will be analyzed to extract what seems to be the root mistake, and I will present my favourite solutions.

Furthermore, various technologies have matured since the creation of many of our systems, and whenever appropriate, these should be used. For example, our understanding of the *structure* of mathematics has significantly increased, yet this is barely reflected in our libraries. The extreme focus on efficiency by the computer algebra community, and correctness by the (interactive) theorem proving community should no longer be considered viable long term strategies. But how does one effectively bridge that gap?

I personally find that a number of (programming) *language-based* solutions are particularly effective, and I will emphasize these. Solutions to some of these problems will be illustrated with code from a prototype of MathScheme 2.0, the system I am developing with Bill Farmer and our research group.

[1] Let us calculate!