

## Editorial

# Towards Computer Aided Mathematics

In his autobiography<sup>1</sup> Bertrand Russell characterizes mathematics as follows: “It seems to me now that mathematics is capable of an artistic excellence as great as that of any music, perhaps greater; not because the pleasure it gives (although very pure) is comparable, either in intensity or in the number of people who feel it, to that of music, but because it gives in absolute perfection that combination, characteristic of great art, of godlike freedom, with the sense of inevitable destiny; because, in fact, it constructs an ideal world where everything is perfect and yet true.” Actually the perception of mathematical research as an artistic discipline has a long history and a significant number of today’s mathematicians share this view. In contrast, however, Russell himself dedicated large parts of his life to defending logicism, that is, the view that mathematics is reducible to logic, and together with Alfred Whitehead he proposed in his influential *Principia Mathematica* an axiomatic system to build all mathematics upon.

The two viewpoints – mathematics as an art versus logicism – may appear contradictory at first. They are not though, if we separate the different aspects of mathematical practice. The invention and shaping of new mathematical structures based on mathematical knowledge as well as on aesthetic and social criteria or the discovery of the essential arguments in complex mathematical proof, for instance, are activities that typically require human ingenuity. On the other hand the verification and grounding of already pre-structured and established chunks of mathematics in foundational systems or the search for simple (sub-)proofs are examples of tasks that often require far less ingenuity.

Some overoptimistic and improperly reflected predictions in the field of artificial intelligence and automated reasoning on the mechanization and automation of mathematics have unfortunately generally questioned the role of human ingenuity in mathematics without making the above distinction clear. Unlike in chess, however, where human intelligence is no longer dominating over machine intelligence, it seems to me that human ingenuity will remain dominant in many essential aspects in mathematics research and education for a long time to come. Taking our distinction above into account this does not mean, however, that there is no need for assistance systems for mathematics and Russell would presumably be delighted to see that today several chunks

---

<sup>1</sup> See Bertrand Russell (1872-1970), *Autobiography*, George Allen and Unwin Ltd, 1967, vol. 1 (1872-1914), pp.158-159.

of mathematics have already been built up from foundational logical systems within different proof assistants.

*So, what is an assistance system for mathematics and what is it good for?*

The notion of an assistance system for mathematics adopted here characterizes an integrated environment of tools supporting a wide range of typical research, publication and knowledge management activities. Examples of mathematical activities are computing, proving, solving, modeling, verifying, structuring, maintaining, searching, inventing, paper writing, explaining, illustrating, and possibly others. Clearly, some of them require a high amount of human ingenuity while others do not. An assistance system for mathematics should support activities for which practical and robust solutions exist, that is, at the moment predominantly those which require less human ingenuity.

Meanwhile an impressive range of mathematical support tools is actually available, for instance, computer algebra systems (e.g., MAPLE and MATHEMATICA), interactive proof assistants (e.g., ISABELLE/HOL and COQ), automated theorem provers (e.g., VAMPIRE and OTTER), model checkers (e.g., SMV), partially integrated hybrid systems (e.g. OMEGA), search engines (e.g., GOOGLE), and publishing and typesetting packages (e.g., LATEX). The integration of one or several of these tools within a uniform environment leads to our notion of an integrated mathematics assistance system. The overall idea, however, is not to replace the mathematician but instead to support a fruitful symbiosis of human and machine intelligence in which the computer takes over tedious routine parts thus setting precious resources free for the human user.

An obvious and very prominent approach to the development of an assistance system for mathematics is the integration of off-the-shelf tools, for instance, automated theorem provers, decision procedures, and computer algebra systems, into interactive proof assistants. An important issue in this approach is the provision of transformational mappings between the different representations employed in the combined tools. Furthermore, the maintenance and effective management of formalized bits of mathematical knowledge in structured (and probably distributed and shared) knowledge bases has to be addressed. Syntactic and semantic search facilities are required for retrieving knowledge from these knowledge sources. Bridging the gap between informal multi-modal mathematical texts and fully formalized representations is just as important as the combination with powerful publication and typesetting packages. In order to reduce the duplication and multiplied encoding effort as currently still required in computer-supported mathematics, we need a smooth and formal transition from technical developments within an assistance system back and forth to high-quality publications. Another important issue is the development of powerful, uniform look-and-feel as well as effective user interfaces

which preferentially support a human-oriented rather than a machine-oriented interaction with the system. They should hide the minute representational and operational details of the integrated tools. Many support tools and the mathematical knowledge sources can ideally be shared between different assistance systems through the development of a mathematical semantic web.

*And who needs assistance systems for mathematics?*

Computer algebra systems and publishing tools, for example, are already routinely employed in mathematical research and practice today. Furthermore, interactive proof assistants and model checkers are nowadays used in industrial applications for formal software and hardware verification and quality assurance. On the other hand mathematics has existed for thousands of years without computer support and it is perfectly valid to doubt, as many working mathematicians actually do, that the immediate impact of the envisioned assistance systems will be overwhelming for the frontiers of mathematical research.

In recent years, however, we can observe a small but increasing number of success stories in computer aided mathematics. For example, the four color theorem has been proven in 1976 by Appel and Haken with significant computer support. This proof had a dubious status for a long time because a verification of it (by hand) seemed impossible. Recently, however, a formal verification within the assistance system COQ was reported by Georges Gonthier at Microsoft Research. Another success story is the verification of a proof of the prime number theorem with the system ISABELLE by Jeremy Avigad at Carnegie Mellon University in 2004.

Presumably the most important recent example is the computer supported proof of the Kepler's conjecture by Thomas Hales at Pittsburgh University. Kepler's conjecture is a problem in discrete geometry which has been unsolved for nearly 400 years. The submission of his results to the Annals of Mathematics resulted in an interesting and controversial debate. Robert D. MacPherson, the editor in chief of the Annals of Mathematics, gave a presentation at the symposium 'The nature of mathematical proof' of the British Royal Society in London in Fall 2004 in which he revealed how difficult it is to review results of this nature: a refereeing board of 12 mathematicians had finally given up to fully verify the proof after 4 years! They could still validate Hales' reduction of the original problem to a wide range of subproblems. However, they were not able to verify (nor to refute) the many subcriteria that Hales solved with significant computer algebra support. This happened for the first time in the history of mathematics! As Hilbert's famous perpetual call from the heart exemplifies: "Da ist das Problem, suche die Lösung. Du kannst sie durch reines

Denken finden, denn in der Mathematik gibt es keinen Ignorabimus”<sup>2</sup>, mathematicians have always held the belief that in principle we know – although we may err – if something is the case or not.

While mathematicians have thus given up on verifying the proof, Hales has started the Flyspeck project. The aim of this project is to reconstruct, formalize, and fully verify Hales complete proof in the assistance system HOL-LIGHT. This is an a posteriori attempt to apply assistance systems in a research frontier of mathematics and due to the complexity of the problem and the comparative mathematical and practical immaturity of today’s mathematical assistance systems this endeavor will certainly require several years of persistent work.

In the long run, however, the envisioned fully integrated assistance systems will support this new style of mathematics not a posteriori but from the very start, ideally with far less effort as currently still required and also at a more human-friendly interaction level.

*Is there some low hanging fruit?*

Yes, there is. Even in case of a failure of the ambitious Flyspeck project, the existing systems are already successfully used in less ambitious mathematics such as formal verification in computer science. In particular students who want to learn mathematics or engineers who want to apply mathematics – both groups are typically confronted with far less ambitious mathematical problems than Hales – may well and actually do already benefit from current mathematics assistance systems. In fact, proof assistants and model checkers have been widely used in applications for software and hardware verification. Also e-learning environments with integrated support tools increasingly attract attention in academia as well as in public applications.

*Why is it so difficult to build an integrated assistance system for mathematics?*

The challenge is to attack the scientific and technological gap between the targeted ideal mathematics assistance environments and the many weaknesses and shortcomings of the current systems. This requires in particular the combination of techniques and expertise from several research areas. Research progress and good research training in this multidisciplinary area can currently probably be best achieved by joining forces in research networks. One example is the European CALCULEMUS research training network (2000-2004),

---

<sup>2</sup> Engl.: There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no ignorabimus.

which puts an emphasis on the training of young researchers in the areas of computer algebra and deduction systems.

Actually, there are relatively few research groups which have sufficient expertise, background and critical mass to cover the whole spectrum of relevant research issues to build an all embracing assistance system for mathematics. This problem is actually analogous to the development of large and all-encompassing AI systems in general; in fact, these assistance systems can be seen as an instance of an ambitious, integrated and general AI system, which researchers claim also in other more common subfields of AI.<sup>3</sup> However, a broad research expertise is only one of the many essential requirements. Availability of human resources, in particular, talented and enthusiastic PhD students with strong implementational skills is another. In fact, most of the existing attempts at large and integrated assistance systems have been predominantly achieved with the help of generations of PhD students and post-docs.<sup>4</sup> Such a student-based development strategy imposes several challenges, not least of which is the software maintenance problem, which is particularly difficult for those groups which do not have the support of an experienced and long-term employed software engineer to control and guarantee a persistent high quality software development along uniform conventions. Probably even harder is the organization of a smooth knowledge transfer in order to pass crucial system expertise from one generation of students to the next. PhD students and researchers in the area of mathematics assistance systems need in addition to scientific talent and implementational skills a broad research interest, excellent communication skills, social competence and teamwork spirit. These requirements are unfortunately accompanied by less than optimal publication opportunities – in comparison, for instance, to the usual theoretical topics in mainstream computer science areas.<sup>5</sup>

---

<sup>3</sup> In their invited talks at this years AAAI-05 conference in Pittsburgh both Ronald J. Brachman and Marvin Minsky argued for building and analyzing large, integrated AI systems. I should think that the envisioned all-embracing assistance systems for mathematics actually cover a wide range of these typical characteristics an ambitious, integrated AI system shall have.

<sup>4</sup> An example is Peter Andrews' TPS system, which is based on the contributions of a row of students such as Dale Miller, Frank Pfenning, Dan Nesmith, Sunil Issar, Hongwei Xi, Matthew Bishop, and Chad Brown. Another example is our own OMEGA project with its long sequence of PhDs and postdocs.

<sup>5</sup> Fortunately publication opportunities have already significantly improved since N.G. de Bruijn's early pioneering project work on AUTOMATH. In a private conversation Prof. deBruijn told me that in the early days of AUTOMATH there was hardly a scientific community they could submit their papers to and discuss their ideas with. This is one of the unfortunate reasons why comparably little material on this influential pioneering work was publicly available.

## *The purpose of this special issue*

In this special issue we provide an overview of grown-up and well established assistance systems for mathematics as well as of some more recent attempts inspired by the same goal. The motivation thereby is twofold: to communicate results and to stimulate further work — ideally through mutual fertilization.

We briefly summarize the contributions.

- The TPS system is represented by the article from Peter Andrews and Chad Brown. It uses Church’s simple type theory for the formalization and foundations of mathematics and it combines natural deduction style interactive proof development with strong automated support by a background reasoner based on the mating method. TPS, which has been extensively employed as a support tool in university lectures, also provides a powerful mechanism to transform its background reasoner’s machine-oriented proofs into human-oriented natural deduction proof objects.
- The MIZAR system which is based on set theory, is represented by the articles of Adam Naumowicz and Josef Urban. Naumowicz presents some recent formalizations of mathematical results in MIZAR and contrasts his formal MIZAR texts with informal mathematical texts. Urban’s article focuses on MIZARMODE, an Emacs-based authoring environment for the MIZAR system. He also describes the proof assistance functions and tools available in MIZARMODE.
- The NUPRL system is addressed in the article by Stuart F. Allen, Mark Bickford, Robert L. Constable, Rich Eaton, Christoph Kreitz, Lori Lorgio, and Evan Moran. The authors present NUPRL’s foundational framework based on computational type theory, they discuss NUPRL’s distributed system architecture and they illustrate how NUPRL’s central database can be employed as a transactional system for formal mathematics.
- The article by Bruno Buchberger, Adrian Cračiu, Tudor Jebelean, Laura Kovács, Temur Kutsia, Koji Nakagawa, Florina Piroi, Nikola Popov, Judit Robu, Markus Rosenkranz, and Wolfgang Windsteiger presents the THEOREMA system, which adopts an interesting top-down approach for the formalization of mathematics. THEOREMA’s implementation is based on the well established MATHEMATICA computer algebra system. In their survey paper the authors illustrate mathematical theory exploration in THEOREMA by a case study and give an overview on some available reasoners and other organizational support tools.
- A combination of the generic theorem proving system ISABELLE with a proof planning system is addressed in the article of Lucas Dixon and Jacques Fleuriot. More precisely, the authors present an integration of the human-oriented proof script language ISAR, the ISABELLE core proof assistant and the proof planner ISAPLANNER, which is a descendant of the Edinburgh proof planners CLAM and  $\lambda$ -CLAM.

- The OMEGA system is represented by the article of Jörg Siekmann, Christoph Benzmüller and Serge Autexier. OMEGA is modular system with a central and hierarchically organized proof data structure supporting proof development and interaction at different levels of granularity. Several supplementary subsystems including automated deduction and computer algebra systems can be called during the search for a proof. Transformation tools support the representation and analysis of their results within OMEGA’s proof data structure. While OMEGA has many characteristics in common with systems like NUPRL, COQ, HOL, or ISABELLE/HOL, it also differs from these systems with respect to its focus on proof planning and in that sense it is more common to the proof planning systems developed at Edinburgh.
- SAD, a Ukrainian assistance system, is presented in the article of Alexander Lyaletski, Andrey Paskevich, and Konstantin Verichinine. It combines a human-oriented interface language with a foreground reasoner which in turn accesses different automated reasoning tools in the background. The user can interact with the system by taking on different roles in the foreground reasoner.
- The article by Claus Zinn describes a computational framework for mechanizing the analysis of carefully authored textbook proofs. Zinn’s proof-of-concept implementation is capable of processing simple textbook proofs and constitutes promising steps towards a natural mathematician-machine interface for proof development and verification.
- AUTOMATH, the Dutch pioneer system, is represented by the article of Freek Wiedijk. He employs his reimplementations of AUTOMATH as a logical framework in which he conducts a ‘comparative review of foundations of mathematics’.

This collection of articles reveals many similarities between the systems but also significant differences. An important challenge is to identify the best of today’s achievements and to integrate them into a single best practice environment. In order to achieve significant progress in our research area the best research strategy is debatable. Two options are “Let the best system win” and “Cooperate, modularize, and exchange components”. I personally advocate the latter – however, time will tell.

I am grateful to all research groups and researchers who submitted articles and who made this special issue possible. A special thanks also to the many reviewers for their effort and to Jörg Siekmann for his support for this special issue and even more for waking my enthusiasm for computer aided mathematics: CAM.

**Christoph Benzmüller**  
Saarland University, Saarbrücken, Germany  
September 08, 2005