

Formal Mathematical Theory Exploration in Theorema

Bruno Buchberger
Johannes Kepler Universität, Linz, Austria

We consider the entire process of mathematical theory exploration: invention of mathematical concepts (axioms and definitions), consistency check for axioms, invention and proof of mathematical propositions, invention of mathematical problems, invention and verification of methods (in particular algorithms) for solving problems. We aim at computer-supporting this process in all phases and at building up and managing verified formal mathematical knowledge bases as a result of this process. Our logic frame is predicate logic in a special version and implementation called "Theorema".

In the course, we will consider several case studies, notably the theory of Groebner bases, and will discuss the heuristics of the process of mathematical theory exploration and possibilities for computer-support in this process. We distinguish between "top-down" and "bottom-up" approaches to mathematical theory exploration. In both approaches, formula schemes play an important role. A formula scheme is a formula in which a couple of unspecified function and predicate symbols occur. Depending on the context, a formula scheme can be used as definition scheme, axiom scheme, proposition scheme, problem scheme, or algorithm scheme and may help both in partially automating the bottom-up and the top-down approach to mathematical theory exploration.

As a special sub-problem within computer-supported mathematical theory exploration, we consider computer-supported algorithm synthesis by the "lazy thinking" method. In this method, we start from a formal problem specification and any algorithm scheme. We try to prove (automatically) the correctness of the algorithm scheme w.r.t. the specification. Typically, this proof will fail because nothing is known about the unspecified "sub-algorithms" (function and predicate symbols) occurring in the scheme. Now, with a special technique, we analyze the failing proof object and try to guess specifications for the unspecified sub-algorithms which will make the correctness proof succeed. Now, we will either be able to identify algorithms in the given knowledge base that meet the specifications for the sub-algorithms. Or we can apply the lazy-thinking technique again for synthesizing suitable sub-algorithms. We have recently shown that this natural algorithm synthesis technique is powerful enough for synthesizing non-trivial algorithms like the Groebner bases algorithm.

The theory of Groebner bases (within commutative algebra) is chosen as a benchmark example for formal mathematical theory exploration because it is both a theory that can be applied for automating theorem proving in other areas of mathematics, e.g. geometry and differential equations, as well as a theory whose automation by the above theory exploration methods is attractive, non-trivial and interesting.

Computer-support for mathematical theory exploration has recently become a very active field of research, in which worldwide approximately 20 research groups are currently involved that are organized in the so called "MKM" (Mathematical Knowledge Management) Network. The ultimate goal is to come up with systems that will support the working mathematicians in their research, teaching, and also publishing activity and to build up formal mathematical knowledge bases that will be accessible, over the web, for automated reasoners of various kinds.

References

1. B. Buchberger, C. Dupre, T. Jebelean, F. Kriftner, K. Nakagawa, D. Vasaru, W. Windsteiger. *Symbolic Computation and Automated Reasoning*, In: Proc. of CALCULEMUS 2000, M. Kerber and M. Kohlhase (eds.), A.K. Peters, Natick, Massachusetts, pp. 98-113, 2000
2. B. Buchberger. *Algorithm Supported Mathematical Theory Exploration: A Personal View and Strategy*, In: Proc. of AISC 2004, B. Buchberger, J. Campbell (eds.), Springer LNAI, 3249, Springer, 2004
3. B. Buchberger. *Towards the Automated Synthesis of a Gröbner Bases Algorithm*, RACSAM, 98/1, 2005
4. B. Buchberger, A. Craciun. *Algorithm Synthesis by Lazy Thinking: Examples and Implementation in Theorema*, In: Proc. of the Mathematical Knowledge Management Workshop, F. Kamareddine (ed.), Electronic Notes on Theoretical Computer Science, Elsevier, pp. 24-59 <http://www.sciencedirect.com/science/journal/15710661>
5. B. Buchberger. *Groebner Rings in Theorema: A Case Study in Functors and Categories*. SFB Report No. 2003-49, Johannes Kepler University Linz, Spezialforschungsbereich F013, November, 2003

The participants of the Summer School can download the above articles from B. Buchberger's home page www.risc.uni-linz.ac.at/people/buchberger (Goto "links / news")

Please respect the copyright of the various publishers.