The Objectives

of Formal Mathematics

and the Workshop on Formal Gröbner Bases Theory

Bruno Buchberger
Research Institute for Symbolic Computation
Johannes Kepler University, Linz, Austria

Talk at Workshop "Formal Gröbner Bases Theory"
RICAM - RISC, Linz, Austria
March 6, 2006

Formal Mathematics

Computer mathematics on the "object level": Given a mathematical theory (e.g. commutative algebra, graph theory, geometry, etc.)

- ○ a problem formulated in the theory is solved by an algorithm formulated in the theory.

Computer mathematics on the "meta-level" (= Formal Mathematics): the process of developing a mathematical theory is supported by algorithms ("automated", "mechanized", ...).

The process of developing a mathematical theory has, at least, the following ingredients:

- ○ inventing (and "verifying") axioms and definitions,
- ○ inventing and proving / disproving propositions,
- ○ inventing (and "verifying") problems,
- ○ inventing and proving methods (algorithms),
- ○ managing knowledge generated in the theory.

(Mathematical Theory Exploration Paradigm # Isolated Theorem Proving Paradigm).

⏮ ◂ ▸ ⏭ 3 of 7

Reflection

The algorithms used in formal mathematics (for a given theory) are / contain often algorithms developed on the object level of some other theory (e.g. algorithm for computing Gröbner bases can be used in generating proofs / disproofs for a wide class of formulae in geometry). Therefore, formal mathematics must contain means for describing and verifying its reasoners. (Reflection.)

(Proved Provers Paradigm.)

⏮ ◂ ▸ ⏭ 4 of 7

The Future of Formal Mathematics

In a certain sense, mathematics started when it started to be "formal".

In terms of a clear notion of algorithm, the foundation of formal mathematics was laid in the 20th century.

Formality in mathematics will develop to a degree not at all expected, desired, and anticipated currently by the majority of mathematicians.

The ingredients for this development are here. (The were not available 40 years ago.)

The amount of work needed to make it happen is still enormous but not beyond the working power of (a small part of) the current mathematical community.

Formality will drastically change the way how mathematics is researched, taught, and applied.

The formalization of mathematics will be never completed, the horizon of the unformalized expands as formalization of some parts of mathematics is achieved. (This is a good reason to work for formalization and not a reason not to work for formalization.)

Formal mathematics needs a different (better) kind of mathematicians.

⏮ ◀ ▶ ⏭ 5 of 7

Tools for Formal Mathematics

Logical Tools

■ For the Verification Steps

"Reasoners" (not only "general" ones but, more importantly "special" ones; i.e. reasoners that are valid only for special theories.)

"Solvers" ("Model Checkers", etc.)

"Retrievers" (essentially, they are reasoners).

■ For the Invention Steps

Schemes (... functors): can be "applied" (by substitution) in order to propose definitions, propositions, problems, algorithms in bottom-up style.

Analysis of failing proofs: can be used for proposing propositions, problems (specifications) in a top-down style.

Others: see talks.

Organizational Tools

Notebook organization,

Label management,

...

⏮ ◀ ▶ ⏭ 6 of 7

Formal Gröbner Bases Theory

Can be used as a benchmark theory for formal mathematics.

Was for me the personal motivation why I started the Theorema project in 1995:

- Wanted to write a textbook on Gröbner Bases; this is still handcrafting.
- Wanted to do research on generalized Sylvester matrices and possible additional "criteria"; formal proofs essential and difficult.
- So many papers on GB, very different level of reliabilty, originality, etc.

By now, a few groups started to develop a formal Gröbner bases theory.

Acknowledgement to *Theorema* Group: T. Jebelean, W. Windsteiger, T. Kutsia, M. Rosenkranz, F. Piroi, and docs.

⏮ ◀ ▶ ⏭ 7 of 7

Examples

Generating conjectures from failing proofs: inductive proofs.

Schemes for inventing definitions, propositions, problems, algorithms.

Functor notation for schemes.