Logic, Methodology and Philosophy of Science

Proceedings of the 14th International Congress (Nancy)

Logic and Science Facing the New Technologies

Edited by

Peter Schroeder-Heister,

Gerhard Heinzmann,

Wilfrid Hodges,

and

Pierre Edouard Bour

ISBN 978-1-84890-169-8

College Publications
Scientific Director: Dov Gabbay
Managing Director: Jane Spurr

http://www.collegepublications.co.uk

Printed by Lightning Source, Milton Keynes, UK

Mathematics and the New Technologies Part II: Computer-Assisted Formal Mathematics and Mathematical Practice

PETER KOEPKE

1 Introduction

Formal mathematics denotes the programme to carry out all of (pure) mathematics in complete formality: to express notions and statements in a *symbolic language* and to prove statements by derivations in a *symbolic calculus*. Due to the complexities of full formalizations this programme was at first merely an attractive vision, going back to ideas like Gottfried Leibniz's *characteristica universalis* and *calculus ratiocinator*. It was *theoretically* vindicated by Kurt Gödel's completeness theorem (Gödel 1929). In recent years, however, formal mathematics is becoming *practically* feasible, using computer support and automatic theorem proving.

Formal mathematics harmonizes with philosophical standpoints that view mathematics as a deductive science, and in particular with *formalism*. Advances in formal mathematics provide a body of *actual formalizations*, as opposed to the theoretical *formalizability* usually considered in formalism. This may shift the balance between various philosophies of mathematics towards formalism. Advances will also provide proof checking and proving tools for the mathematical practitioner, and they will influence the mathematical practice.

So the argument between conventional philosophies of mathematics and the *Philosophy of Mathematical Practice* may be dependent on concrete answers to questions like: Which proofs can be generated automatically? Can ordinary mathematical proofs, or intelligent but limited modifications thereof, be checked automatically? Can one make the application of formal mathematics just as natural as the use of other mathematical software like computer algebra systems or the LaTeX typesetting software?

P. Schroeder-Heister, G. Heinzmann, W. Hodges, and P. E. Bour (eds.), *Logic, Methodology and Philosophy of Science. Proceedings of the Fourteenth International Congress (Nancy), 409–426. 2014.*

So before embarking on philosophical speculations we try to give an impression of the potential of formal mathematics by appraising its current state and likely midterm developments. After a general introduction, we list important formal mathematics systems, in which substantial mathematical results have been proved or proof-checked. These systems use input and output languages reminiscent of programming languages. We suggest to improve the *naturalness* of formal mathematics by using (*controlled*) *natural languages* instead. The exploratory systems SAD and Naproche implement some of these ideas.

We expect that by combining best methods from a variety of systems formal mathematics will become stronger and in particular acceptable and applicable in ordinary mathematical work. This will also have significant philosophical implications.

2 Formal mathematics

Formal mathematics emerged alongside formal logic and modern abstract mathematics. In *The Principles of Mathematics* (Russell 1938, Preface to the First Edition, v) Bertrand Russell enunciates the standpoint of *logicism*:

> [...] that all pure mathematics deals exclusively with concepts definable in terms of a very small number of fundamental logical concepts, and that all its propositions are deducible from a very small number of fundamental logical principles [...].

He then formulates the programme of *formal mathematics*, to be pursued in a subsequent volume (Russell 1938, Preface to the First Edition, p. vi):

> The second volume [...] will contain chains of deductions, from the premisses of symbolic logic through Arithmetic, finite and infinite, to Geometry, [...].

This programme was partially realized by A. N. Whitehead and Russell in *Principia Mathematica* (Whitehead & Russell 1910-1913). Gödel begins his article on the incompleteness theorems by describing the state of formal mathematics at the time (Gödel 1931, 144, translation: 145):

> Die Entwicklung der Mathematik in der Richtung zu größerer Exaktheit hat bekanntlich dazu geführt, daß weite Gebiete von ihr formalisiert wurden, in der Art, daß das Beweisen nach einigen wenigen mechanischen Regeln vollzogen werden kann. Die umfassendsten derzeit aufgestellten formalen Systeme sind das System der *Principia Mathematica (PM)* einerseits, das

> Zermelo-Fraenkelsche (von J. v. Neumann weiter ausgebildete) Axiomensystem der Mengenlehre andererseits. Diese beiden Systeme sind so weit, daß alle heute in der Mathematik angewendeten Beweismethoden in ihnen formalisiert, d.h. auf einige wenige Axiome und Schlußregeln zurückgeführt sind.
>
> *The development of mathematics toward greater precision has led, as is well known, to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules. The most comprehensive formal systems that have been set up hitherto are the system of* Principia mathematica (PM) *on the one hand and the Zermelo-Fraenkel axiom system of set theory (further developed by J. von Neumann) on the other. These two systems are so comprehensive that in them all methods of proof today used in mathematics are formalized, that is, reduced to a few axioms and rules of inference.*

First-order set theory and in particular the Zermelo-Fraenkel system ZFC with the axiom of choice is commonly accepted as the natural foundation of modern structure-orientated mathematics. There is a considerable degree of agreement between ontology and semantics since many basic notions are defined set-theoretically, e.g.:

> A *group* is a *set* together with [...].

By Gödel's completeness theorem (Gödel 1929) there is *complete* agreement between syntax and semantics: every proof can be replaced by a formal derivation (in set theory). These observations underpin the programme of formal mathematics: to *actually* produce formal derivations from informal proofs.

3 On the feasibility of formal mathematics

Principia Mathematica turned out to be a project of unexpected dimensions and difficulties. Only a small part of the intended matter could be covered. Russell wrote in his autobiography (Russell 1998, 155):

> [...] my intellect never recovered from the strain.

Nicolas Bourbaki who worked towards a complete and systematic exposition of mathematics claimed the unfeasibility of complete formalizations (Bourbaki 2004, 10, 11):

> [...] such a project is absolutely unrealizable: the tiniest proof at the beginnings of the Theory of Sets would already require

> several hundreds of signs for its complete formalization. [...] formalized mathematics cannot in practice be written down in full [...].

But with the advent of electronic computers, the practical side of long repetitive tasks appeared in a different light. In 1962, John McCarthy wrote (McCarthy 1962):

> Checking mathematical proofs is potentially one of the most interesting and useful applications of automatic computers. Computers can check not only the proofs of new mathematical theorems but also proofs that complex engineering systems and computer programs meet their specifications. Proofs to be checked by computer may be briefer and easier to write than the informal proofs acceptable to mathematicians. This is because the computer can be asked to do much more work to check each step than a human is willing to do, and this permits longer and fewer steps. [...] The combination of proof-checking techniques with proof-finding heuristics will permit mathematicians to try out ideas for proofs that are still quite vague and may speed up mathematical research.

4 Practical systems for formal mathematics

McCarthy's prediction is being realized in formal mathematics. Since the 1950's there have been a number of formal mathematics systems, differing in purpose, techniques, and scope. Automatic theorem provers are intended to find formal deductions for hypotheses given to the system. There are general purpose automated theorem provers for arbitrary (first-order) statements, and specialized provers optimized for specific areas. It was soon realized that automated theorem provers were hardly able to match the abilities of expert mathematicians in finding successful strategies and constructions for proofs of non-trivial statements. This gave rise to systems where human users provide clues for the proof-finding algorithm, either in advance in some dedicated proof language or interactively.

In this section we briefly describe a selection of important formal mathematics systems which are geared towards wide coverage, ordinary mathematical argumentation, and proving prominent theorems. These systems require expert users to master their idiosyncratic languages and commands, and to understand the underlying logical and software mechanisms.

Automath (Automath) was a pioneering large-scale project in formal mathematics, begun in 1967 by Nicolaas de Bruijn. de Bruijn explained in (de Bruijn 1994, 215):

> [...] the Automath project tries to bring communication with machines in harmony with the usual communication between people.

L. S. van Benthem Jutting (van Benthem Jutting 1977) demonstrated the applicability of Automath to substantial mathematical theories by transcribing the *Grundlagen der Analysis* of Edmund Landau (Landau 1930) into Automath. Automath contained many important ideas and techniques which were taken over by other projects.

Some parts of formal mathematics have developed in parallel with general computer science. So Automath employed a LISP-like input language, which by today's standards would hardly considered to be "readable".

The problem of "readability" in formal mathematics was addressed by the Mizar system (Mizar), which has been developed by Andrzej Trybulec since about 1975. The Mizar language is related to the ALGOL programming language and intends to capture several features of the common mathematical language. Moreover Mizar allows a more natural proof style by bridging "obvious" proof steps with the aid of an integrated automated prover. The system accepts simple transformations and deductions which are common in ordinary proofs without further justification. Most importantly, Mizar comprises a vast library of checked proof texts which can be used as lemmas for further proving. The library contains material from many fields of mathematics, including the Banach Fixed Point Theorem for compact spaces, Fermat's Little Theorem, the Fundamental Theorem of Algebra, the Fundamental Theorem of Arithmetic, the Gödel Completeness Theorem, the Jordan Curve Theorem, and many more.

Whereas Mizar uses a fixed first-order logic and Zermelo-Fraenkel set theory, the Isabelle project (Isabelle) initiated by Larry Paulson only has a minimal inbuilt logic and can be configured to work with different logics and background theories. One of the largest Isabelle formalizations is that of Gödel's theorem of the relative consistency of the axiom of choice by Paulson (Paulson 2003). Many other substantial theorems have been redone in Isabelle like the elementary proof of the Prime Number Theorem by Jeremy Avigad *et al.* (Avigad *et al.* 2007).

The system Coq (Coq) is built on type theory and intuitionistic logic. The most spectacular Coq formalizations are the proof of the Four Colour Theorem by G. Gonthier (Gonthier 2008), and, very recently, the Feit-Thompson theorem (Gonthier 2012) which is an important part of the classification of finite simple groups.

Higher order logic is the basis of the HOL Light system (HOL light) by John Harrison, in which Harrison has proved theorems like the Fundamental

Theorem of Calculus, Brouwer's Fixpoint Theorem, and the Prime Number Theorem, using an analytical proof.

5 Enhancing the naturalness of formal mathematics

Although formal mathematics theoretically has a universal potential, it has not yet entered mathematical practice. Freek Wiedijk (Wiedijk 2007) states:

> The other reason that there has not been much progress on the vision from the QED manifesto is that currently *formalized mathematics does not resemble real mathematics at all*. Formal proofs look like computer program source code.

An average mathematician does not use any of the existing formal mathematics systems since they do not go along with the usual, or "natural" mathematical experience.

The naturalness of mathematical texts depends on many factors which are related to human abilities and expectations in various areas. Fields of mathematics have developed their own sublanguages of the mathematical language with specific symbols, methods and implicit background assumptions. A text may be directed at an audience with a specific background knowledge and sophistication. These factors will also be appreciated differently by different individuals. So we can only discuss some general aspects of formal systems which affect naturalness.

5.1 Mathematical aspects

Mathematical theories strongly influence their style of presentation. Obviously a theory is more adequate for a natural formalization if it is highly formal anyway. If a theory is based on intuitively well-understood concepts from, e.g., geometry, physics, or social interaction, then the presentation tends to appeal to those intuitions in plain but linguistically involved natural language which may be difficult to analyze. If a theory is built up axiomatically or algebraically the development is usually more formal. In the course of unfolding a theory new intuitions evolve and are employed. So the beginnings of a theory will be more adequate for natural formalizations than advanced parts.

Mathematical texts combine logical arguments with numerical and symbolic computations. Up to now the techniques of formal mathematics have emphasized logical arguments, so one should prefer "logical" theories. Set theory in some appropriate axiomatization is a powerful system for the general formalization of mathematics, and has been used in several formalization projects, e.g., by Mizar.

5.2 Linguistic aspects

The language of mathematics combines natural language with mathematical formulas. Most natural language words and constructs retain their original meanings, but there are some exceptions and extensions. Through definitions, a word like "ring" may get a new, mathematical semantics, which is completely determined by a formal definition. The word, however, retains its standard grammar as a neuter noun with plural form "rings". Usually the choice of defined words is not completely arbitrary, but takes into account natural language intuitions, systematics, and conventions. Also completely new words, patterns of words, and phrases may be introduced.

Concerning the meanings of grammatical constructs, the standard mathematical language tries to be complete and unambiguous. Whereas the coordination with "or" is in natural language often understood as "either-or", the usual mathematical interpretation is the inclusive "or"; an exclusive "or" has to be made explicit by "either-or" and other means. The tendency to avoid ambiguities facilitates the linguistic analysis of the mathematical language.

Mathematical exactness requires an analysis of *every* sentence of a text. The analysis must be intelligable for a human author so that the author can keep control over the process. This necessitates the use of a grammar-based *deep linguistic analysis* instead of, e.g., stochastic methods.

A mathematical text is a *discourse* in the language of mathematics, i.e., a structured sequence of sentences. *Discourse representation theory* (see Kamp & Reyle 1993) provides means to transform a given discourse into a logical representation which retains important structural elements of the text like the scopes of certain constructs or the interdependencies of sentences through pronouns and other anaphora.

One is lead to the definition of *controlled natural languages* (CNL) which are subsets of the natural language of mathematics with a strict formal grammar and formal semantics. A powerful controlled languages with an associated computer implementation is the language *Attempto Controlled English* (ACE) which combines a rich "natural" language with mechanisms of interest for mathematical applications.

5.3 Internal representations

Attempto Controlled English translates input texts into discourse representation structures as an intermediate layer between natural input and its first-order equivalent. There are, however, aspects of proofs which standard discourse representation theory does not model properly, like the order of statements or the scope of assumptions. This motivates the introduction of *proof representation structures* (PRS) which are enriched discourse representation structures able to represent various argumentative and procedural aspects. PRS seem to be crucial data structures to connect natural and formal proofs.

A PRS should contain information on the visibility of relevant assumptions for every statement in the proof. Immediately preceding statements or distinguished main lemmas or theorems are the most probable and "visible" preconditions for a statement so that these should be attempted with higher priority for the proof of the current statement. A good design of visibility criteria can help the automated prover and make proofs more natural in the sense that "obvious" potential premises are selected by the system in a way similar to the tactics of a human prover.

5.4 Logical aspects

In principle all mathematical statements can be translated into first-order statements about sets and the membership-relation. Standard set-theoretic formalizations of mathematical notions like the coding of integers by von Neumann ordinals introduce exponential growth and may not be practically feasable. Therefore intermediate logics should be used which are close to the "natural logic" of mathematical input texts. This requires an efficient (weak) type system so that complex objects or notions can be atomic at some higher level of the type system. This was already described by Bourbaki (Bourbaki 2004, 10):

> [...] it is imperative to condense the formalized text by the introduction of a fairly large number of new words (called *abbreviating symbols*) and additional rules of syntax (called *deductive criteria*). By doing this we obtain languages which are much more manageable than the formalized language in its strict sense. Any mathematician will agree that these condensed languages can be considered as merely shorthand transcriptions of the original formalized language.

5.5 Automated theorem proving

Proofs come with a certain step size or *granularity* depending on the style of proof. Proof checking amounts to the justification of each proof step, either by the argumentative abilities of a human (expert) reader, or by interpolating proof steps by a formal derivation in case of automated proof checking. Ideally automated theorem provers (ATP) like Otter or Vampire should be able to interpolate proof steps of a natural granularity. Experiments with existing formal mathematics systems indicate that this is possible at least in certain contexts.

5.6 Typesetting

Mathematical texts stand out by the elaborate typography for formulas. Systems like TeX and LaTeX enable mathematicians to do mathematical typesetting without expert help. These systems have become *de facto* standards in mathematical publishing and can be considered “natural” formats for communicating mathematics. Natural formal mathematics should accept those formats.

6 Examples of natural formal mathematics systems

6.1 *System for Automated Deduction*: The SAD project

The SAD project (SAD) is based on a controlled natural language for mathematics called ForTheL (Formula Theory Language), which goes back to the 1960’s and was further developed by Alexander Lyaletski, Andrei Paskevich, and Konstantin Verchinine (Verchinine *et al.* 2007). SAD is designed to approximate parts of common mathematical language and argumentation. Several frequent and useful phrases and methods of proof have been implemented with appropriate first-order semantics. The language includes a soft type system which is akin to the naive typing often found in mathematical texts. The proof checking process is devided into two layers: a *reasoner* attempts to identify inferences which to humans appear immediate or trivial; if the reasoner fails, the proof search is delegated to some automated theorem prover. Although SAD is only a small prototypical system, it allows for surprisingly natural mathematical texts. The following is an excerpt from a proof that the square root of a prime number is irrational:

```
Theorem Main.

For all nonzero natural numbers n,m,p if p * (m * m)
= (n * n) then p is compound.

Proof by induction.  Let n,m,p be nonzero natural
numbers.
Assume that p * (m * m) = (n * n).  Assume that p is
prime.  Hence p divides n * n and p divides n.  Take
q = n / p.
Then m * m = p * (q * q).  Indeed p * (m * m) = p *
(p * (q * q)).  m < n.  Indeed n <= m => n * n <= m
* m.

Hence p is compound.

qed.
```

The frugal ASCII appearance of ForTheL texts can easily be improved by putting a LaTeX layer on top of the language. Here is an original excerpt from an SAD + LaTeX proof of the infinitude of prime numbers which comes rather close to textbook versions:

> **Theorem 1.** *The set of prime numbers is infinite.*
>
> **Proof.** Let A be a finite set of prime numbers. Take a function p and a number r such that p lists A in r steps. ran$p \subseteq \ {}^{+}$. $\prod_{i=1}^{r} p_i \neq 0$. Take $n = \prod_{i=1}^{r} p_i + 1$. n is nontrivial. Take a prime divisor q of n.
>
> Let us show that q is not an element of A. Assume the contrary. Take i such that $(1 \leq i \leq r$ and $q = p_i)$. p_i divides $\prod_{i=1}^{r} p_i$ (by MultProd). Then q divides 1 (by DivMin). Contradiction. qed.
>
> Hence A is not the set of prime numbers. ■

6.2 *Natural Proof Checking*: The Naproche project

Whereas SAD achieves an impressive but limited degree of linguistic naturalness with a carefully crafted small controlled language, the Naproche project (Naproche) aims at an analysis and formal approximation of extensive parts of the full natural language of mathematics. The project set out by analysing mathematical texts using annotations, formal grammars and discourse representations (see Koepke & Schröder 2002; 2003; Cramer & Schöder 2012; Cramer *et al.* 2011). The fact that formal semantics in linguistics usually leads to representations in first-order logic is advantageous for mathematical texts (see Cramer *et al.* 2009). In the Naproche software, first-order representations are transformed into queries to automatic theorem provers (ATP) in order to check whether statements in mathematical texts are logical consequences of previously established facts (see Cramer *et al.* 2010*a*;*b*).

The grammars and formats of the linguistic analysis define a controlled language of accepted sentences, the Naproche language. Like Automath, the Naproche project also takes Landau's *Grundlagen* (Landau 1930) as a benchmark text to be reformulated and checked. This has been done for the first two chapters of the book, and we give a sample of a representative theorem and the beginning of its proof, taken from the translation (Landau 1966):

> **Theorem 4,** and at the same time **Definition 1:**
>
> *To every pair of numbers x, y, we may assign in exactly one way a natural number, called $x + y$ ($+$ to be read ”plus”), such that*

1. $x + 1 = x'$ *for every* x,

2. $x + y' = (x + y)'$ *for every* x *and every* y.

$x + y$ *is called the sum of* x *and* y, *or the number obtained by the addition of* y *to* x.

Proof: A) First we will show that for each fixed x there is at most one possibility of defining $x+y$ for all y in such a way that

$$x + 1 = x'$$

and

$$x + y' = (x + y)' \quad \text{for every} y.$$

Let a_y and b_y be defined for all y and be such that

$$a_1 = x', \quad b_1 = x',$$

$$a_{y'} = (a_y)', \quad b_{y'} = (b_y)' \quad \text{for every} y.$$

Let $\mathfrak{M}$ be the set of all y for which

$$a_y = b_y.$$

I)

$$a_1 = x' = b_1;$$

Hence 1 belongs to $\mathfrak{M}$.

II) If y belongs to $\mathfrak{M}$ then

$$a_y = b_y,$$

hence by Axiom 2

$$(a_y)' = (b_y)',$$

therefore

$$a_{y'} = (a_y)' = (b_y)' = b_{y'},$$

so that y' belongs to $\mathfrak{M}$.

Hence $\mathfrak{M}$ is the set of all natural numbers; i.e., for every y we have

$$a_y = b_y.$$

The argument, proving the uniqueness of an addition function on the natural numbers, is rather subtle since it uses higher-order arithmetic. This requires some (background) theory of sets and functions, which is not made explicit in the Landau text. The Naproche system includes such a background theory (Cramer 2012) so that the proofs get cleaner and don't have to appeal to "the possibility to define" certain terms. Here is a checked rendering of the Landau argument in the current version of the Naproche system:

> Theorem 4: There is precisely one function $x, y \mapsto x + y$ such that for all x, y, $x + y$ is a natural number and $x + 1 = x'$ and $x + y' = (x + y)'$.
> Proof:
> A) Fix x. Suppose that there are functions $y \mapsto a_y$ and $y \mapsto b_y$ such that $a_1 = x'$ and $b_1 = x'$ and for all y, $a_{y'} = (a_y)'$ and $b_{y'} = (b_y)'$.
> Let $\mathfrak{M}$ be the set of y such that $a_y = b_y$.
> $a_1 = x' = b_1$, so 1 belongs to $\mathfrak{M}$.
> If y belongs to $\mathfrak{M}$, then $a_y = b_y$, i.e., by axiom 2 $(a_y)' = (b_y)'$, i.e., $a_{y'} = (a_y)' = (b_y)' = b_{y'}$, i.e., y' belongs to $\mathfrak{M}$. So $\mathfrak{M}$ contains all natural numbers. Thus for all y, $a_y = b_y$.
> Thus there is at most one function $y \mapsto x+y$ such that $x+1 = x'$ and for all y, $x + y' = (x + y)'$.

Note that this text can be seen as a stricter version of Landau's argument. Due to the natural language features of Naproche and the built-in function theory the reformulated text is as short and readable as the original.

7 Perspectives of formal mathematics

Against the background of the state of formal mathematics as sketched above I propose a sequence of theses, leading from safe ones already substantiated to more speculative ones. In section 4 we saw:

1. Formal mathematics has become an established and active research area.

2. Formal mathematics is already covering a wide range of substantial mathematical results.

There are singular points where current mathematical research uses formal mathematics, e.g., the flyspeck project (flyspeck) of Thomas Hales to construct a formal proof of the Kepler conjecture, or the work of Vladimir Voevodsky in homotopy theory, using the Coq proof assistant. Thus:

3. Formal mathematics is beginning to interact with research mathematics.

4. Formal mathematics could become part of mathematical practice.

In line with Wiedijk's analysis of the current role of formal mathematics we hold that:

5. The acceptance of formal mathematics in mathematical practice will depend on the naturalness of its application.

Section 5 identified areas and proposed methods for the improvement of naturalness. This will involve the combination of best methods from various, already existing systems:

6. The naturalness of strong formal mathematics can be increased considerably.

Therefore:

7. Formal mathematics will become part of mathematical practice.

But it seems too early to make predictions on the degree of coverage and acceptance of formal mathematics tools in the day to day work of future mathematicians. Some practioners of formal mathematics like Jeremy Avigad, Kevin Donnelly, David Gray, and Paul Raff hold (Avigad *et al.* 2007):

> On a personal note, we are entirely convinced that, although there is a long road ahead, formal verification of mathematics will inevitably become commonplace. Getting to that point will require both theoretical and practical ingenuity, but we do not see any conceptual hurdles.

On the other hand one can expect resistance by mathematicians who feel that they would lose the traditional freedom of mathematical presentation, which can be very sloppy and even formally false in "inessential" or "trivial" places. To allay the reservations of traditional mathematicians, formal mathematics systems have to offer rich and natural interfaces, and there has to be reasonable added value for the user.

7.1 A scenario: Formal mathematics and textbook mathematics

Many attempts in formal mathematics are directed towards a register of mathematical discourse described as *textbook mathematics*. This involves extensive texts, a systematic development of some limited area of mathematics, and a rather detailed renderings of proofs. The prerequisites of such texts should be simple, and everything else is introduced within the text, preferably in a Definition–Theorem–Proof style.

Let us assume that formal mathematics is able, within the next decade, to handle some such texts: experts which understand the mathematics and the formal mathematics system reformulate chapters of textbooks into texts which are very similar in typesetting, language, and logical structure to the original text, but which are also checked for correctness by the system. The feasibility of this scenario will depend on the kind of mathematics to be handled (see 5.1).

What are the consequences of such developments? Obviously one could then have textbooks, which are readable like standard textbooks, but which are completely correct (we don't want to discuss the remote possibilities of computer and software faults at this place). This may be a relief to authors and referees. A referee could concentrate on main ideas instead of checking tedious details. On the other hand the demand for formalization may force some mathematically unnatural or superfluous issues into the presentation. Computer proof checking will provide possibilities to explore logical dependencies within the text which are not explicitely mentioned: the automated checker can produce a log of its proof (a "proof object" of some kind) which can be searched for information generated during checking. So the checkable textbook text is like a surveyable surface, under which one could explore different layers of logical detail.

Most mathematical research articles combine some high level reasoning with extensive low level arguments, often of some "combinatorial" kind. Although the high level reasoning may be far above the abilities of formal mathematics systems, combinatorial arguments sometimes have a textbook style as described above. One might consider writing "textbook arguments" with the help of formal mathematics systems to assist authors, referees, and readers. Often the high level reasoning is familiar to experts and proceeds along established intuitions of the field. By contrast, combinatorial arguments are sometimes difficult to grasp and intuite, so that a validity check may be welcomed by everybody involved. In this way, formal mathematics designed for the textbook level might also enter research mathematics.

The introduction of such techniques will depend on decisions and trends within the wider mathematical community. As an example, the systems TEX and LATEX could manifest themselves since they gave authors support and control of a process that previously could only be managed by a longwinded iterative process of approximations to the desired typeset result. Further benefits were given by the small footprint of the data files, the openness of the formats, the quality of the software, and other factors. Within a few years TEX and LATEX have become a *de facto* standard which is now made essentially mandatory by publishers.

8 Philosophical perspectives

The development of formal mathematics may be viewed as a strengthening of the formalist position. The above proof of the infinitude of primes is not only a text that communicates number theoretic ideas to a fellow mathematician, and which **could** be fully formalized. In a rich formal system, including automated theorem proving, the text **is** already a formal text. Does this indicate some analogy with Richard Montague's *English as a Formal Language* (Montague 1974)?

In the discussion of informal versus formal proofs their seemingly huge dissimilarity is a decisive aspect. Hannes Leitgeb (Leitgeb 2009) writes:

> why not think of "formally provable(-in-T)" (for some instantiation of "T") as a Carnapian explication of "informally provable"? The answer is simple: because it is not. According to Carnap, whatever explicates an explicandum must be as similar as possible to the latter, but as our comparison from above has shown, formal provability and informal provability are just too dissimilar to satisfy this criterion.

But if in the case of the infinitude of primes T is taken to be the abovementioned system SAD + LATEX informal and formal proof may coincide so that at least in certain situations "formally provable(-in-T)" might be a Carnapian explication of "informally provable"!

Strengthening formalism will affect the balance between the main positions in the philosophy of mathematics and may have far-reaching consequences. In his MSc thesis (Tanswell 2012) Fenner Tanswell has argued that Naproche could be a tool for overcoming the philosophical objections to formalism and develop a new type of formalism.

On the other hand it may be too early to start this discussion in detail. So let me just mention one issue with respect to the Philosophy of Mathematical Practice: The current way of checking mathematical correctness, rather than being meticulous logical checking, has been described by philosophers of mathematical practice as a complicated process based on a network of trust in intuitions, published papers, authorities, refereeing processes, etc. This system will change once formal certificates are available for parts of the mathematical research and dissemination process. Initially certificates will be seen as a welcome extra justification, until they will become mandatory, at least for certain kinds of arguments. Does this mean that certain observations of the Philosophy of Mathematical Practice concerning the shakyness of the present network of trust will become outdated in the long run?

Bibliography

ACE. Attempto Controlled English. http://attempto.ifi.uzh.ch/.

Automath. www.cs.ru.nl/˜freek/aut/.

Avigad, J., Donnelly, K., *et al.* (2007). A formally verified proof of the prime number theorem. *ACM Transactions on Computational Logic (TOCL)*, *9*(1), 1–23.

Bourbaki, N. (2004). *Theory of Sets.* Berlin: Springer.

Coq. http://coq.inria.fr.

Cramer, M. (2012). Implicit dynamic function introduction and its connections to the foundations of mathematics. In *Philosophy, Mathematics, Linguistics: Aspects of Interaction*, St. Petersburg.

Cramer, M., Fisseni, B., *et al.* (2010*a*). The Naproche Project – Controlled natural language proof checking of mathematical texts. In *Controlled Natural Language 2009*, *Lecture Notes in Computer Science*, vol. 5972, Fuchs, N., ed., Berlin; Heidelberg: Springer, 170–186, doi:10.1007/978-3-642-14418-9_11.

Cramer, M., Koepke, P., & Schröder, B. (2011). Parsing and disambiguation of symbolic mathematics in the naproche system. In *Intelligent Computer Mathematics*, *Lecture Notes in Computer Science*, vol. 6824, Davenport, J. H., Farmer, W. M., *et al.*, eds., Berlin; Heidelberg: Springer, 180–195, doi:10.1007/978-3-642-22673-1_13.

Cramer, M., Koepke, P., *et al.* (2009). From proof texts to logic. Discourse representation structures for proof texts in mathematics. In *From Form to Meaning: Processing Texts Automatically*, Chiarcos, C. et al., ed., Tübingen: Narr.

Cramer, M., Koepke, P., *et al.* (2010*b*). Premise selection in the Naproche System. In *Automated Reasoning, IJCAR 2010*, *Lecture Notes in Computer Science*, vol. 6173, Giesl, J. & Hähnle, R., eds., Berlin; Heidelberg: Springer, 434–440, doi: 10.1007/978-3-642-14203-1_37.

Cramer, M. & Schöder, B. (2012). Interpreting plurals in the Naproche CNL. In *Controlled Natural Language*, *Lecture Notes in Computer Science*, vol. 7175, Rosner, M. & Fuchs, N. E., eds., Berlin Heidelberg: Springer, 43–52, doi: 10.1007/978-3-642-31175-8_3.

de Bruijn, N. G. (1994). Reflections on Automath. In *Selected Papers on Automath*, *Studies in Logic and the Foundations of Mathematics*, vol. 133, R.P. Nederpelt, J. G. & de Vrijer, R., eds., Elsevier, 201–228, doi: 10.1016/S0049-237X(08)70205-2.

flyspeck. http://code.google.com/p/flyspeck.

Gödel, K. (1929). Über die Vollständigkeit des Logikkalküls. In *Kurt Gödel,* Collected Works – *Vol. I: Publications 1929–1936*, Feferman, S., ed., New York: Oxford University Press, 60–101, 1986.

Gödel, K. (1931). Über formal unentscheidbare Sätze der *Principia mathematica* und verwandter Systeme I. In *Kurt Gödel,* Collected Works – *Vol. I: Publications 1929–1936*, Feferman, S., ed., New York: Oxford University Press, 144–195, 1986.

Gonthier, G. (2008). Formal proof – the four-color theorem. *Notices of the AMS*, *55*, 1382–1393.

Gonthier, G. (2012). Public email.

HOL light. www.cl.cam.ac.uk/ jrh13/hol-light/.

Isabelle. www.cl.cam.ac.uk/research/hvg/isabelle.

Kamp, H. & Reyle, U. (1993). *From Discourse to Logic*. Dordrecht: Kluwer.

Koepke, P. & Schröder, B. (2002). Natürlich formal. In *Computational Linguistics – Achievements and Perspectives*, Willée, G. et al., eds., Sankt Augustin: Gardez!-Verlag, 184–189.

Koepke, P. & Schröder, B. (2003). ProofML – eine Annotationssprache für natürliche Beweise. *LDV Forum*, *18*, 428–441.

Landau, E. (1930). *Grundlagen der Analysis*. Leipzig: Akademische Verlagsgesellschaft.

Landau, E. (1966). *Foundations of Analysis*. New York: Chelsea Pub., 3rd edn., translated into English by F. Steinhardt.

Leitgeb, H. (2009). On formal and informal provability. In *New Waves in Philosophy of Mathematics*, Bueno, O. & Linnebo, Ø., eds., Basingstoke; New York: Palgrave Macmillan, 263–299.

McCarthy, J. (1962). Computer programs for checking mathematical proofs. In *Recursive Function Theory: Proceedings of the Fifth Symposium in Pure Mathematics*, Decker, J. C. E., ed., American Mathematical Society, 219–227.

Mizar. http://mizar.uwb.edu.pl.

Montague, R. (1974). English as a formal language. In *Formal Philosophy: Selected Papers of Richard Montague*, Thomason, R. H., ed., New Haven: Yale University Press, 247–270.

Naproche. http://naproche.net.

Paulson, L. C. (2003). The relative consistency of the axiom of choice mechanized using Isabelle/ZF. *LMS Journal of Computation and Mathematics*, *6*, 198–248, doi:10.1112/S1461157000000449.

Russell, B. (1938). *The Principles of Mathematics.* W. W. Norton, 2nd edn., 1st ed., Cambridge: University Press, 1903.

Russell, B. (1998). *Autobiography.* Hoboken: Taylor & Francis.

SAD. http://nevidal.org.

Tanswell, F. (2012). *Proof and Prejudice: Why Formalising doesn't make you a Formalist.* Msc thesis, Universiteit van Amsterdam, ILLC Publications MoL-2012-07.

van Benthem Jutting, B. (1977). *Checking Landau's "Grundlagen" in the Automath system.* Ph.D. thesis, Eindhoven University of Technology.

Verchinine, K., Lyaletski, A., & Paskevich, A. (2007). System for automated deduction (SAD): A tool for proof verification. In *Automated Deduction – CADE-21*, *Lecture Notes in Computer Science*, vol. 4603, Pfenning, F., ed., Berlin Heidelberg: Springer, 398–403, doi:10.1007/978-3-540-73595-3_29.

Whitehead, A. N. & Russell, B. (1910-1913). *Principia Mathematica.* Cambridge: Cambridge University Press.

Wiedijk, F. (2007). The QED Manifesto revisited. *Studies in Logic, Grammar and Rhetoric*, *10*(23), 121–133.

Peter Koepke
Mathematical Institute
University of Bonn
Germany
koepke@math.uni-bonn.de