Proof Documents: Presentation and Representation

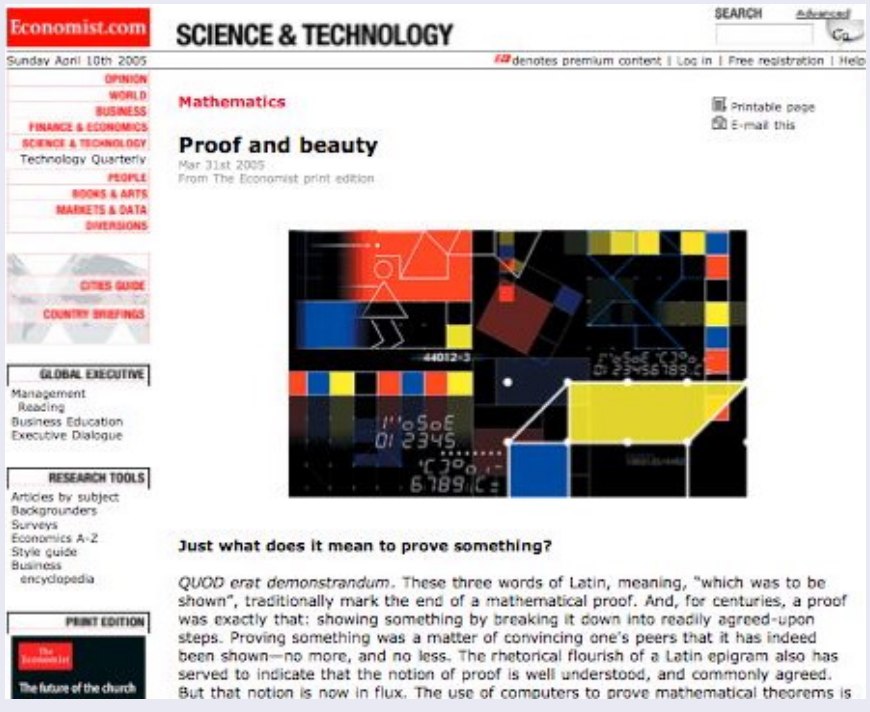
Bengt Nordström

Computer Science, Chalmers and University of Göteborg

National Institute of Advanced Industrial Science and Technology, Senri, Japan, April 2005

Proof editors in the news!

Economist, March 31, 2005



Economist.com

SCIENCE & TECHNOLOGY

SEARCH Advanced

Sunday April 10th 2005

denotes premium content | Log in | Free registration | Help

OPINION
WORLD
BUSINESS
FINANCE & ECONOMICS
SCIENCE & TECHNOLOGY
Technology Quarterly
PEOPLE
BOOKS & ARTS
MARKETS & DATA
DIVERSIONS

CITIES GUIDE
COUNTRY BRIEFINGS

GLOBAL EXECUTIVE
Management
Reading
Business Education
Executive Dialogue

RESEARCH TOOLS
Articles by subject
Backgrounders
Surveys
Economics A-Z
Style guide
Business encyclopedia

PRINT EDITION
The future of the church

Mathematics

Printable page
E-mail this

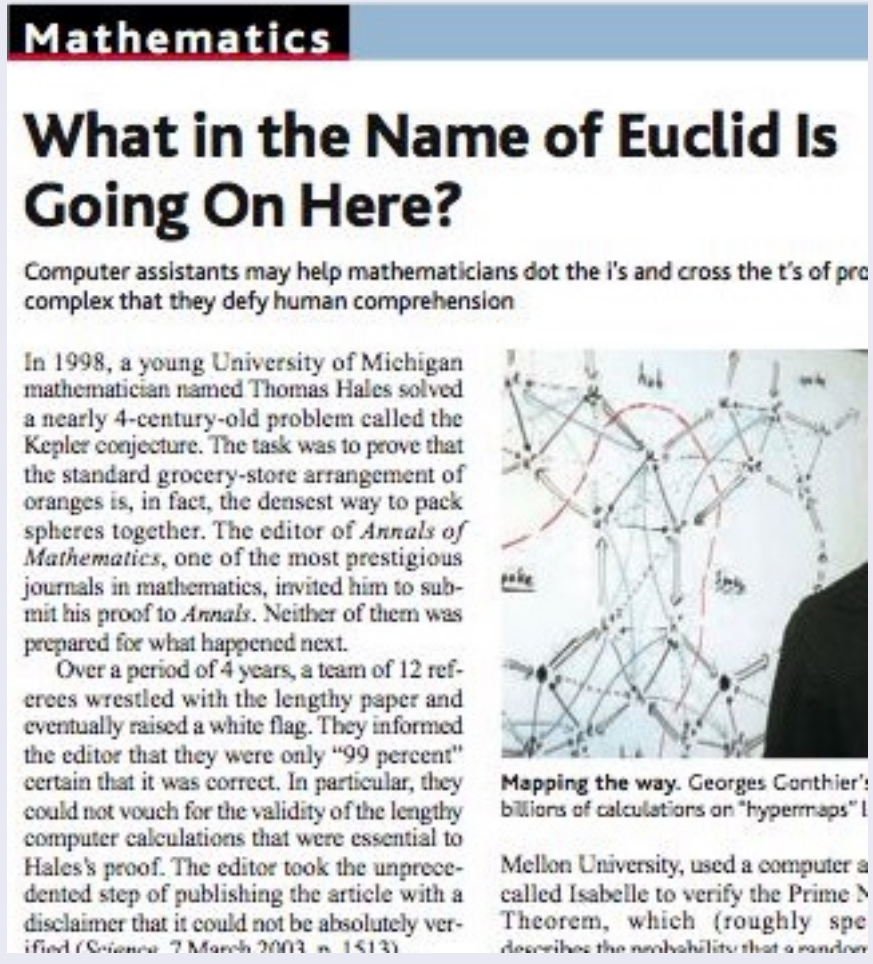
Proof and beauty

Mar 31st 2005
From The Economist print edition

Just what does it mean to prove something?

QUOD erat demonstrandum. These three words of Latin, meaning, "which was to be shown", traditionally mark the end of a mathematical proof. And, for centuries, a proof was exactly that: showing something by breaking it down into readily agreed-upon steps. Proving something was a matter of convincing one's peers that it has indeed been shown—no more, and no less. The rhetorical flourish of a Latin epigram also has served to indicate that the notion of proof is well understood, and commonly agreed. But that notion is now in flux. The use of computers to prove mathematical theorems is

Science, March 4, 2005



Mathematics

What in the Name of Euclid Is Going On Here?

Computer assistants may help mathematicians dot the i's and cross the t's of pro complex that they defy human comprehension

In 1998, a young University of Michigan mathematician named Thomas Hales solved a nearly 4-century-old problem called the Kepler conjecture. The task was to prove that the standard grocery-store arrangement of oranges is, in fact, the densest way to pack spheres together. The editor of *Annals of Mathematics*, one of the most prestigious journals in mathematics, invited him to submit his proof to *Annals*. Neither of them was prepared for what happened next.

Over a period of 4 years, a team of 12 referees wrestled with the lengthy paper and eventually raised a white flag. They informed the editor that they were only "99 percent" certain that it was correct. In particular, they could not vouch for the validity of the lengthy computer calculations that were essential to Hales's proof. The editor took the unprecedented step of publishing the article with a disclaimer that it could not be absolutely ver-

Mapping the way. Georges Gonthier's billions of calculations on "hypermaps" l

Mellon University, used a computer a called Isabelle to verify the Prime N Theorem, which (roughly spe

How can proof editors become a tool for programmers and mathematicians?

We must rely on the proof:

- Small and understood proof checker.
- Well understood relationship between the proof checker and the logic.
- Well understood logic.

But this is not enough!

- Proofs readable by humans.
- The final product should be a document consisting of formal and informal parts. (Matematica).
- The editor is important.

The editor is important

Two choices:

- Structure oriented WYSIWYG editor. (Mathematica, Word, Alfa)
- Powerful text editor like emacs.

In both cases, the concrete syntax of the user language becomes important.

Underlying assumptions

A type theoretic view on document structure:

- A document is represented by an abstract syntax tree (type theoretic object) which gives all important structure of it. This object expresses how a document is built up from parts.
- The type of the abstract syntax tree expresses the syntactic structure of it.
- Dependent types will be used to express syntactic restrictions.
- The concrete syntax of a document is a mapping from its abstract syntax tree to a string (this will be generalized).
- The same abstract syntax tree can have many concrete syntaxes, multilingual documents.

Syntax

abstract syntax

the type of the abstract syntax tree

concrete syntax

a mapping from the abstract syntax tree to a string

There are usually many concrete syntaxes for the same abstract syntax tree:

- user syntax (language represented as text, must be parsable)
- with mathematical symbols
- advanced layout (not parsable in general)

User syntax

Requirements:

- Must be parsable.
- Unstructured text should be simple to input. (Programming languages are not good: needs special quoting)
- Mathematical expressions like infix, postfix, mixfix operators should be simple to input. (XML, S-expressions are not good).

Our proof languages are not good for this.

- The user should have a view that she is writing a document containing mathematical expression and structural information.
- Not that she is writing a mathematical expression containing informal text.

Distinction between user syntax and concrete syntax

Some mathematical syntax requires a lot of mathematical knowledge to be parsed. If we distinguish the user syntax from the presentation syntax things become much simpler.

- There is no need to require that the concrete syntax is parsable (it is enough that a human reader can parse it).
- Hiding of subexpressions from the concrete syntax is completely free as long a human can read it. Replacing a proof with the word trivial (or FOL) is fine if it is obvious for a human.

How are mathematical documents represented today?

XML

- XML is a language for inductively defined objects.
- A concrete syntax can be obtained by using a style sheet, a function which maps an XML document to an HTML document (which can be printed).
- The type system for the abstract syntax is a simple (but complicated) type system like in programming languages.

LaTeX

- The abstract syntax tree of a document can be expressed by functional constants. There is no type system.
- The concrete syntax can be specified by giving macro definitions to the functions.

Why typed documents?

- Errors are found early.
- Meaningful help from the editor (e.g. better completions).
- Increases the possibilities for computing with documents as data.
- Interesting application of type theory.

Example of dependent types in documents

The abstract syntax of an adress

```
Adress = [country: Countries,
          zipcode: Zip (country),
             city: City(country, zipcode)]
where
Countries: Set
         ={Sweden, Japan, UK, ...}
Zip: Countries -> Set;
Zip(Sweden)=Digits5;
Zip(Japan) = ...;
Zip(UK)    = ...;
```

In this way we can express that the syntax of the zip code depends on the country, and make sure that the zip code and the city matches each other.

Related work I

Grammatical Framework (Aarne Ranta)

- Grammar for natural languages.
- Abstract and concrete syntax.
- Type system: Martin-Löf's type theory.
- Concrete syntax expressed in a functional language.
- Multilinguality.

Related work II

WebALT Consortium

- use existing standards for representing mathematics on the web and existing linguistic technologies to produce language-independent mathematical didactical material.
- will create software and sample content for an XML database of mathematical problems to be used in undergraduate university courses in mathematics.

Scenario

- A document editor which can be used to produce a mechanically checked proof.
- Proofs readable by humans.
- Small type checker.
- Relationship between type checker and formal system is spelled out.
- No manual!

References

- Types consortium. `www.cs.chalmers.se/Cs/Research/Logic/Types/`
- Agda homepage. `http://www.cs.chalmers.se/ catarina/agda`
- Grammatical framework homepage. `www.cs.chalmers.se/ aarne/GF/`,
- T. Coquand, R. Pollack, and M. Takeyama. A logical framework with dependently typed records. In *Typed lambda calculi and applications (2003), Lecture Notes in Comput. Sci., 2701*.
- Alfa homepage. `www.cs.chalmers.se/ hallgren/Alfa/`
- *Int. Conf. on Mathematical Knowledge Management*, LNCS 3119, Springer, 2004.
- Coq homepage. `http://pauillac.inria.fr/coq/`, 1999.
- WebALT consortium. `http://webalt.math.helsinki.fi`