Proof, message and certificate

CICM 2012

Bremen, Germany, july 2012

Andrea Asperti

Dipartimento di Scienze dell'Informazione
UniversitÃ degli Studi di Bologna

10/07/2012

Abstract

The recent achievements obtained by means of Interactive Theorem Provers in the automatic verification of complex mathematical results have reopened an old and interesting debate about the essence and purpose of proofs, emphasizing the **dichotomy between message and certificate**.

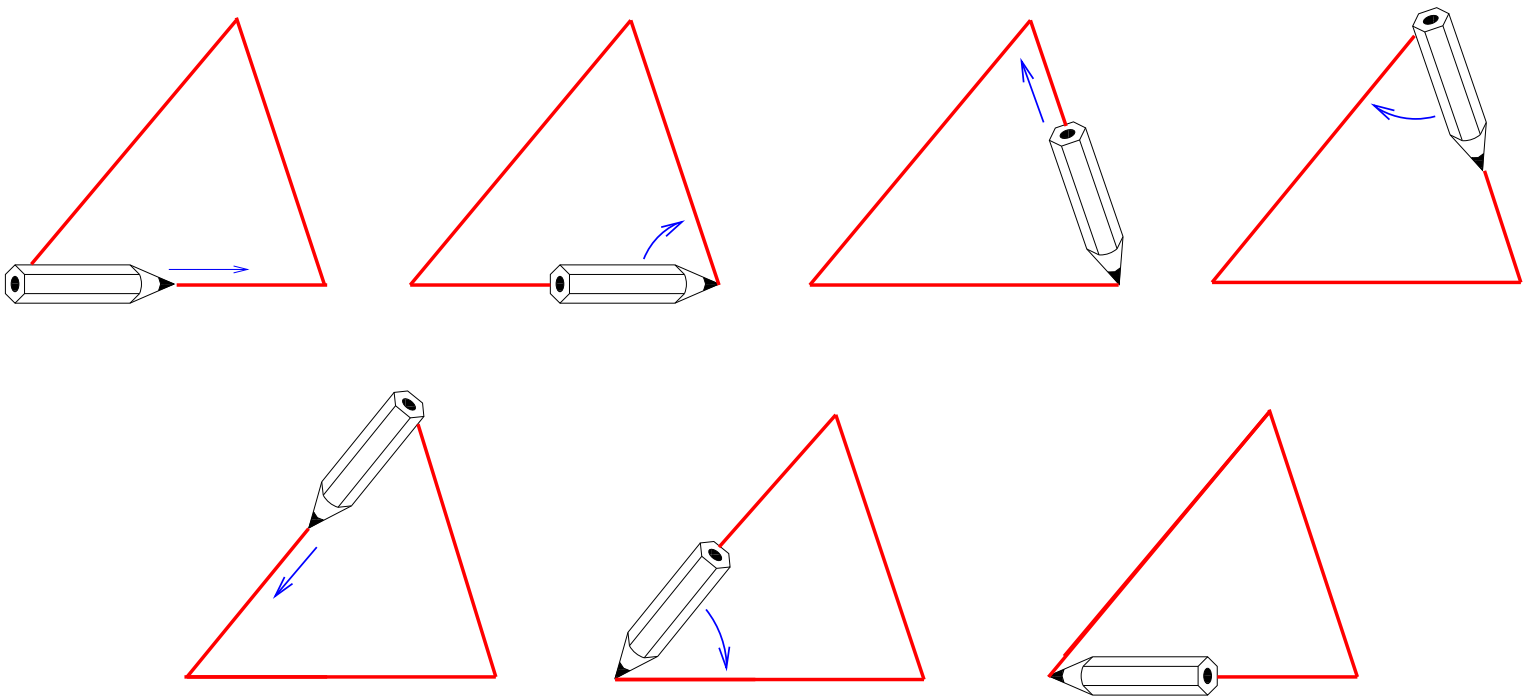
We claim that it is important to **prevent the divorce between these two epistemological functions**, discussing the implications for the field of mathematical knowledge management.

Content

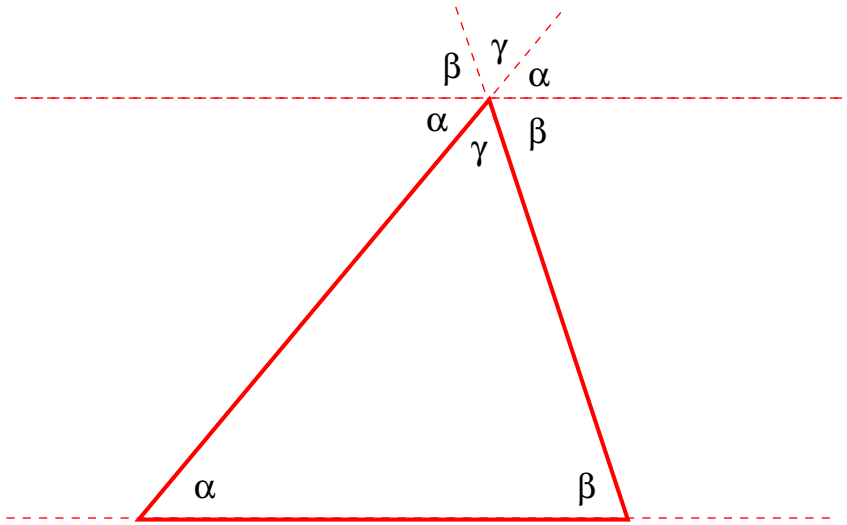
Outline

A proof of Euclid's Theorem

In December 2010, Aaron Sloman posted a message on the MKM mailing list that raised an interesting debate. His message was centered around the following "proof" of Euclid's Theorem,



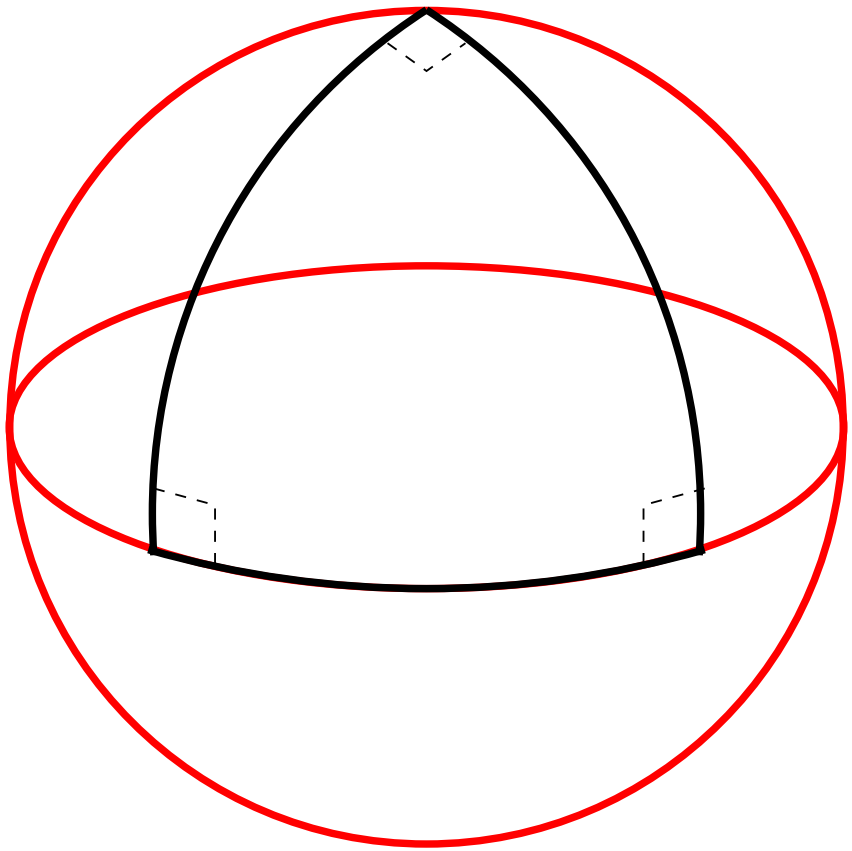
Discussion on the MKM mailing list



Dana Scott

The proof is fine and really is the same as the classical proof. To see this, translate (by parallel translation) all the three angles of the triangle up to the line through the top vertex of the triangle parallel to the lower side. [...]

Non euclidean geometries



Discussion on the MKM mailing list

Arnon Avron

If this "proof" is taught to students as a full, valid proof, then I do not see how the teacher will be able to explain to those students where the hell Euclid's fifth postulate (or the parallels axiom) is used here, or even what is the connection between the theorem and parallel lines.

Dana Scott

I should have commented in my explanation of the proof that if you translate the line on which the base of the triangle sits along each of the sides up to the vertex, then both actions result in the same line – the unique parallel.

Thought experiments

According to Lakatos, "proofs" proves nothing: they are just "thought experiments" not eventually leading to the expected result.

Lakatos - 1976

After Columbus one should not be surprised if one does not solve the problem one has set out to solve.

Message and certificate

It is usually acknowledged that proofs have a double epistemological function:

- **Message**: emphasis on *communication*: the proof is supposed to explain – by providing intuitions – the reasons for believing in the validity of a given statement, and convey information about the line of thought used to conjecture and to approach the result.
- **Certificate**: emphasis on *verification*: the proof is supposed to provide a precise line of reasoning that can be verified in an objective and essentially mechanical way.

Extreme positions

negate any deductive validity to proofs:

Hardy - 1928 [7]

There is strictly speaking no such thing as a mathematical proof; we can, in the last analysis, do nothing but point; [...] proofs are what Littlewood and I call *gas*, rhetorical flourishes designed to affect psychology, pictures on the board in the lecture, devices to stimulate the imagination of pupils.

negate the possibility of *communication* without a clear, objective and verifiable assessment of its actual content:

N.G.De Brujin - 2004 [5]

If you can't explain your mathematics to a machine it is an illusion to think you can explain it to a student.

Outline

The problem of complexity

The dichotomy between message and certificate has been furtherly stressed by the recent achievements in the field on interactive theorem proving, controverting the **general disbelief** about the actual feasibility of complete formalization:

R.A. De Millo, R.J. Lipton, A.J. Perlis -1979 [2]

[...] A formal demonstration of one of Ramanujan's conjectures assuming set theory and elementary analysis would take about two thousand pages.

Bourbaki - 1950 [4]

The tiniest proof at the beginning of the Theory of Sets would already require several hundreds of signs for its complete formalization.

The role of provers

Maurer - 1979 [11]

We can make an analogy here with compiling a higher level language program into a machine language. Originally this was done by hand [...], then compilers came along and started to do the job automatically.

Harrison - 2008 [8]

The arrival of the computer changes the situation dramatically. [...] checking conformance to formal rules is one of the things computers are very good at. [...] the Bourbaki claim that the transition to a completely formal text is routine seems almost an open invitation to give the task to computers.

Formal Proofs

The collection $\mathcal{V}$ of true arithmetical formulas is a productive set.

A formal system $\mathcal{T}$ may provide a (forcedly uncomplete) r.e. approximation of $\mathcal{V}$ (usually expressed as an existential projection of a recursive set of proof-statement pairs).

In extenso, a formal proof for a given formal system $\mathcal{T}$ is any additional information permitting to decide if a formula φ belongs to $\mathcal{T}$ (for instance, to the dimension of the proof).

Space and time

- the dimension of the formal proof can be made arbitrarily small at the expense of the time required for its checking
- the formal proof can be arbitrarily distant from *any* message.

The divorce between message and certificate

D. Mackenzie [10]

Ever since Euclid, mathematical proofs have served a dual purpose: certifying that a statement is true, and explaining why it is true. Now those two epistemological functions may be divorced. In the future, the computer assistant may take care of the certification and leave the mathematician to look for an explanation that humans can understand.

Should we really attribute (following Mackenzie) a positive value to this eventuality?

Anonymous reviewer

In the end the point of a (ideal) proof is to show, without doubt, that some claim is true. If I do not even have to read the proof itself (because it was formalized in a proof assistant) it is even better.

A risk to be avoided

- the message (informative content) is the proof, not the statement (for a proof theorist, the semantics of a statement is its proof; the statement is just a decoration). Proofs (not statements) embody the techniques of mathematics and shape the actual organization of this discipline into a structured collection of interconnected notions and theories.
- if the certificate is divorced from the message, it is enough (up to the adequacy of the encoding) to certify the correctness of the statement, but it says nothing about the correctness of its supposed "explanation". If the explanation is not tightly related to the actual proof we have no evidence of its validity, hence we do not know if we can trust the message.

Outline

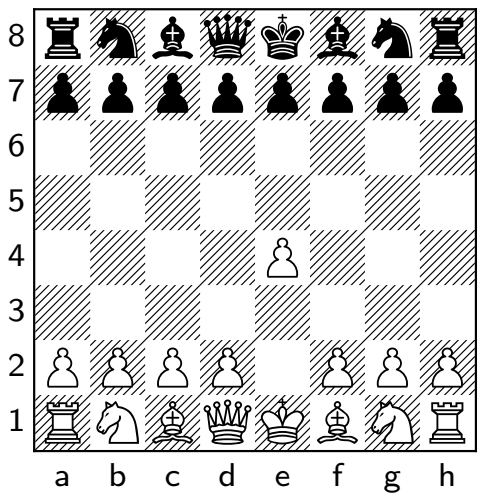
Declarative vs. procedural

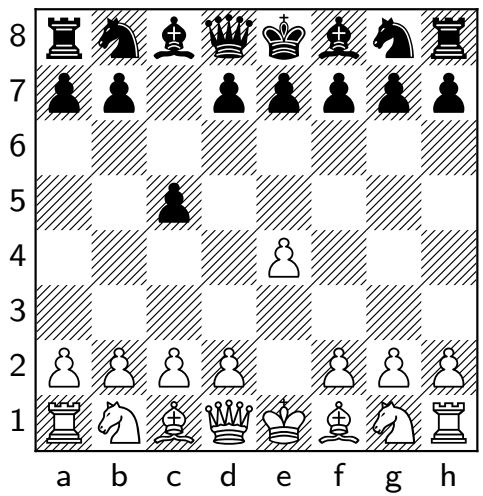
A chess game can be described in essentially two ways: as a sequence of *moves* or as a sequence of *positions*.

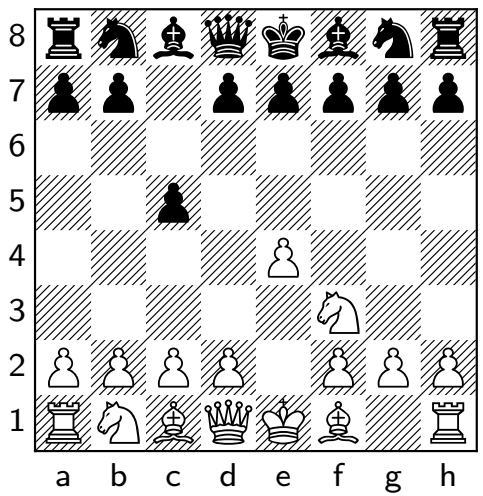
procedural

1 e4 c5; **2 ♘f3 d6**; **3 d4 c×d4**; . . .

declarative







. . .

Pros and Cons

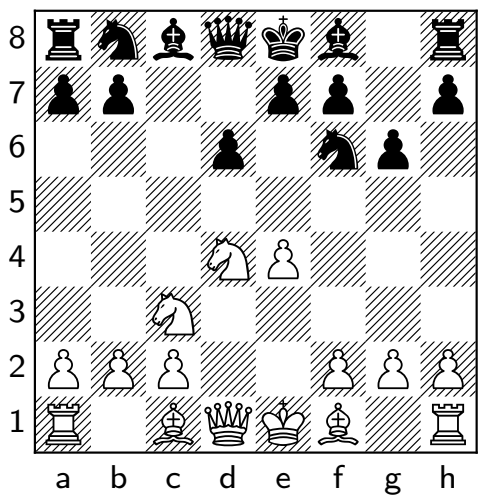
- **Procedural**: very compact but quite unreadable: each move refers to a state implicitly defined by the previous steps. Not meant to be read but to be interactively re-executed.
- **Declarative** declarative languages provide, at each instant, an explicit description of the current state: since the evolution does not depend on the past, you do not need to remember or rebuild any information and may entirely focus on the given state. Declarative descriptions are hence immediately readable, but typically much more *verbose*.

Logical Cuts

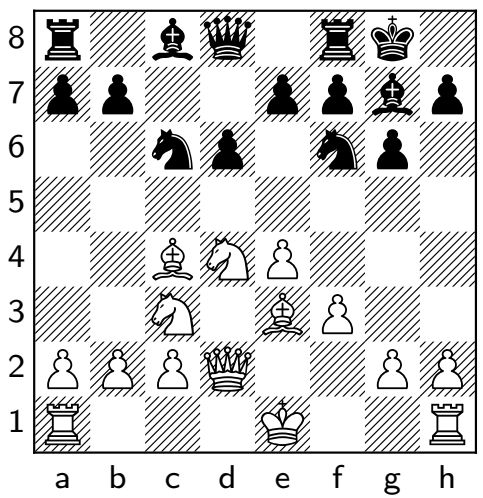
Procedural and declarative languages are not orthogonal: they **complement each other and intergate together well**.

Fisher vs Larsen, Portoroz 1958, Sicilian Defense, Yugoslavian Attack at the Dragon Variation

1 e4 c5
2 ♘f3 d6
3 d4 c×d4
4 ♘×d4 ♘f6
5 ♘c3 g6



6 ♗e3 ♗g7
7 f3 O-O
8 ♕d2 ♘c6
9 ♗c4



Logical Cuts

Inserting a cut (explicit state description) in a procedural script is not a problem.

Reducing the verbosity of a declarative description is less obvious: your should either insert **fragments of procedural languages**, or **rely on the intelligence of the machine** to automatically fill the missing gaps.

In the latter case, the level of granularity is **machine-oriented, and not human oriented**.

Outline

An analogy with software

We have a similar situation in software development: writing a program requires understanding and solving a problem, but it is extremely difficult to extract such a knowledge (the *message*) from the final code (playing the role of *certification*).

The major investment, in programming as well as in formalization, is not the actual writing up of the program, but the preliminary phase of analysis, planning and design; it is a real pity that this information gets essentially lost in the resulting encoding.

Conclusion

- ITP technologies exacerbate the tension between the roles of message and certificate in formal proofs
- it is important to prevent their divorce, improving readibility and documentation of formal scripts
- contrarily to what is currently believed, it is not evident that declarative languages are in a better position than procedural ones to carry out this task
- simple documentation generators are likely to be more rapidly adopted by users of interactive provers than sophisticated authoring interfaces

Bibliografia


A.Asperti and J.Avigad. Zen and the art of formalization.
Mathematical Structures in Computer Science, 21(4), pp.679-682, 2011.


A.Asperti, H.Geuvers and R.Natarajan.
Social processes, program verification and all that.
Mathematical Structures in Computer Science, 19(5), pp.877-896, 2009.


A.Asperti and C.Sacerdoti Coen.
Some Considerations on the Usability of Interactive Provers.
Proc. of CICM 2010, LNCS 6167, pp. 147-156. 2010.


N.Bourbaki. The architecture of mathematics. *Monthly*, 57:221–232, 1950.


N.G.De Bruijn. Memories of the automath project.
Invited Lecture at the Mathematics Knowledge Management Symposium, 25-29 November 2003, Heriot-Watt University, Edinburgh, Scotland.


R. A. DeMillo, R. J. Lipton, and Alan J. Perlis.
Social processes and proofs of theorems and programs. *Commun. ACM*, 22(5):271–280, 1979.


G. H. Hardy. Mathematical proof. *Mind*, 38:1–25, 1928.


J.Harrison. Formal proof - theory and practice.
Notices of the American Mathematical Society, 55:1395–1406, 2008.


I. Lakatos. *Proofs and Refutations: The Logic of Mathematical Discovery*.
Cambridge University Press, 1976.


D.MacKenzie. What in the name of euclid is going on here?
Science, 207(5714):1402–1403, 2005.


W. D. Maurer. Letter to the editor. *Communications of the ACM*, 22:625–629, 1979.