Chapter 1
Checking proofs

Jesse Alama and Reinhard Kahle

1.1 Introduction

Argumentative practice in mathematics evidently takes a number of shapes. An important part of understanding mathematical argumentation, putting aside its special subject matters (numbers, shapes, spaces, sets, functions, etc.), is that mathematical argument often tends toward formality, and it often has superlative epistemic goals: often the aim of a piece of mathematical argumentation is to *prove* that such-and-such a student is *logically true* or *logically valid consequence* of some assumptions; a proved student thus seems to be *indubitable*, *certain*, or *irrefutable*. These aims generally do not depend on the subject matter of what is being argued about; whether one discusses functions, numbers, spaces, shapes, sets, arrangements, flows, figures, or fields, mathematical argumentation, in its final, published, form (and even in ordinary mathematical conversation) tends to be formal and self-consciously explicit about its own argumentative structure. The problem, then, is to better understand the notion of *mathematical proof*. We are interested in this paper in the phenomenon of mathematical proof considered as a species of argumentative practice in mathematics.

For us in this paper the central feature of mathematical argumentation—specifically, mathematical arguments that are put forward with the intention of showing that a certain proposition is a true or valid–is its in-principle formalizability. By the in-principle formalizability of an argument we understand that there exists a formal derivation in some conventionally accepted formalism suited for mathematical reasoning of the proposition that commences from some conventional set of foundational axioms in a gap-free way all the way to the (formalized version of the) proposition.

Jesse Alama
Center for Artificial Intelligence, New University of Lisbon, e-mail: j.alama@fct.unl.pt

Reinhard Kahle
Center for Artificial Intelligence, New University of Lisbon, e-mail: kahle@mat.uc.pt

Even a modest exposure to the practice of producing formal proofs common in introductory courses in logic, mathematics, computer science, law, linguistics, etc., quickly makes clear that the conventional formalisms for reconstructing an argument formally tend to be practically unsuited to the task for which they were designed. Without claiming to offer a complete list, we have

- the method of truth tables,
- Aristotle's syllogisms and variants thereof,
- analytic and semantic tableaux (e.g., Smullyan-style or Jeffrey-style),
- statement-justification tables (as one often sees in elementary courses of geometry),
- Euler/Venn diagrams (for expressing relationships of inclusion and exclusion),
- natural deduction (in various forms: Gentzen, Fitch, Jáskowski, Suppes, . . .)
- Hilbert-style calculi (a linear format preceding from axioms and generally using a handful of rules, e.g., modus ponens),
- Toulmin diagrams (in which there the various roles of statements are represented, so that not all statements are simply bald "premises")
- sequent calculi (á la Gentzen).

The list is quite incomplete; the reader is invited to recall other formats for representing argumentation formally. The point of our list is to suggest to the reader that numerous formalisms available for representing or reconstructing arguments, especially mathematical ones.

It is one thing to formalize a piece of syllogistic reasoning or to use a truth table or tableau method for showing that a short propositional statement such as $p \wedge q \rightarrow p$ is a tautology. But for arguments of any complexity, one sees quickly that reconstructing the argument formally quickly becomes tedious: the formalized argument is often much longer, in an everyday sense, than the argument that it is intended to formalize. One loses the thread of the formalized argument, since most formalisms mandate that one spell out all steps, significant or insignificant. The formal reconstruction takes too long, and the reward at the end (if one has enough patience!) pales to the cost of the formalization.

If one insists on writing mathematical arguments entirely in accordance with the demands of, say, a standard Hilbert-style calculus (where modus ponens is the only rule of inference)[1], then checking a formalized mathematical argument is indeed exceedingly routine, but also exceedingly time-consuming. One wonders what payoff might be had if one were to formalize one's arguments. Lakatos, asking what one can discover in a formalized mathematical theory, gives one answer: "One can discover the solution to problems which a suitably programmed Turing machine could

[1] It is not always the case that modus ponens is the only rule of inference available in a Hilbert style system. In certain systems of modal logic, for example, one typically finds a rule of necessitation as part of a Hilbert-style formalism. But for classical propositional and classical logic, as well as for others, it is known and standard to assume that in a Hilbert-style calculus modus ponens is the only rule of inference. The main feature of Hilbert-style calculi is that they have very few rules, placing the deductive burden of the formalism on its axioms, which are formulas, rather than rules of inference.

solve in a finite time (such as: is a certain alleged proof a proof or not?). No mathematician is interested in following out the dreary mechanical 'method' prescribed by such decision procedures" (Lakatos, 1976, 4). It seems that these formalism in general and the formalisms in which one can reconstruct "informal" arguments, whatever virtues it has (e.g., the soundness and completeness of various systems for expressing derivations), are simply not a practical tool. Those who stress formalisms for writing proofs seem to be overpromising what those formalisms can deliver. It seems that, even if we are interested in the formal side of mathematical reasoning, we need to rest content with its in-principle formalizability. Those who are not so interested in formalization might even suspect that the impracticality of formalizing interesting mathematical arguments constitutes a reductio of the notion of the in-principle formalizability of mathematical proof. If the "in-principle" part of "in-principle formalizability" is so essential, perhaps there is something wrong with the notion of formalizability in the first place.

We do not disagree with the view that formalizing interesting arguments (let alone mathematical proofs) is often tedious. We would like to defend, though, the in-principle formalizability of mathematical proofs as one of their important features by explaining their in-practice formalizability. We are interested, specifically, in the problem of *checking a proof*. Because the ordinary discussion of in-principle formalizability is not taking full account of an important capability that can be brought to bear when formalizing arguments. Instead of formalizing mathematical arguments in our heads or with pencil-and-paper, why not use computers to assist us in the task? Computers are obviously capable of doing symbolic computation at a rate, and with less regard for tedium, than humans have when working only in their own heads or with pencil and paper. We are now in the possession of a wealth of tools that help us to practically reconstruct an informal argument in a formal setting, check whether the formalization is a valid argument and, if not, what defects it has.

In our view these technological developments are important for argumentation theory because, on the one hand, they shed new light on classical topics such as reconstructing and appraising arguments (especially mathematical ones), and, on the other hand, the developments suggest formal treatment of topics that might be thought to be essentially informal.

Our view is closely related to that of Azzouni (Azzouni, 2004), whose so-called derivation-indicator view about mathematical proofs (briefly, that ordinary "informal" mathematical proofs serve as indicators of corresponding formal derivations). The success of computer-assisted (formal) theorem proving projects as discussed here can serve as evidence for Azzouni's view. Derivations for many mainstream mathematical theorems are now available; the days when one can only dream of live derivations for any substantial mathematical theorem are long over. Of course, the empirical success of computer-assisted formal theorem proving projects does not show that Azzouni is right. Our view is compatible with contenders, such as Rav (Rav, 2007). We do not offer formal proof construction and checking as a replacement for ordinary mathematical proof practices, nor are we implicitly suggesting that formal proofs ought to be a central object of interest in argumentation theory in mathematics. Nor, finally, can we recommend formal proofs for everyone; it requires, certainly, a friendly (or at least patient) attitude toward formalization, which not everyone has.

This chapter stands out from several of the others in this volume by its focus on mathematical proof and its connection with formal logic. A fair amount of argumentation theory can be seen as trying to escape from the musty chains of formal reasoning by asking questions about argumentation that are ignored, spurned, or untreatable by the old tools of deductive, valid, formal logic. From this perspective, our contribution might appear to be an unwelcome throwback. Although we focus on such "traditional" matters, we are by no means claiming that in-principle formalizability exhausts the interests of an argumentation theorist looking at mathematics. Further, we do not claim that the subject of *proof* exhausts the study of mathematical argumentation. Proof is clearly but one aspect of a multifarious phenomenon, as the other contributions to this volume can testify. (See, for example, van Bengedem (van Bengedem, 1988).) And although our interests are clearly "traditional" or "foundational", our focus on formalizability stems from the same desire among argumentation theorists looking at mathematics to find out what formal logic does *not* account for. Our modest suggestion is that, thanks to developments in automated reasoning systems (Porteraro (Portoraro, 2008) is a useful survey), new light is shed on the notion of in-principle formalizability and suggests new problems that, we believe, will be of interest to argumentation theorists.

Our focus in the present paper is on evaluating formalized mathematical arguments, or, rather, checking proofs. We illustrate how this apparently "dreary mechanical method" that Lakatos was referring to can in fact offer insight into a formalized mathematical argument. Our discussion is based on modern computer implementations of the process of checking formalized proofs, in the guise of so-called *interactive theorem provers* (also known as *proof assistants*). We will focus on the MIZAR interactive theorem prover,[2] which is based on classical first-order logic, set theory, and natural deduction. There are many actively maintained interactive theorem provers now available:

- COQ[3]
- ISABELLE[4]
- HOL[5]) and some variants, such as HOL LIGHT[6] and HOL ZERO[7]

Among these MIZAR is chosen for its relatively straightforward proof syntax, which is most likely to be immediately accessible to an unfamiliar reader. We are not interested in defending a claim about which of the great variety of interactive theorems provers now available is "best". For lack of space, we cannot provide a comprehensive introduction to MIZAR; we refer the reader to (Grabowski et al., 2010). We

[2] http://mizar.org

[3] http://coq.inria.fr

[4] http://www.cl.cam.ac.uk/research/hvg/isabelle/

[5] http://hol.sourceforge.net/

[6] http://www.cl.cam.ac.uk/~jrh13/hol-light/

[7] http://proof-technologies.com/holzero.html

will explain the relevant parts of MIZAR's language for representing mathematical proofs as necessary.

1.2 Computer-assisted proof construction

The problem of formalizing mathematical arguments is rather old. The roots of formalization arguably go back to Euclid, if not earlier (Netz, 2003). For lack of space, we have to ignore a rich history, doing much injustice to many intellectual forbears, and skip ahead past the invention of the modern computer in the 1930s. Some of the earliest research in what we now call artificial intelligence was on the formalization of mathematical arguments, specifically, the task of using computers to search autonomously for formal proofs of mathematical claims (e.g Wang, 1960). Wang's groundbreaking research led to automatically found proofs of many theorems of *Principia Mathematica*. His remarks, made in 1960, have proved to be rather prescient:

> The time is ripe for a new branch of applied logic which may be called "inferential" analysis, which treats proofs as numerical analysis does calculations. This discipline seems capable, in the not too remote future, of leading to machine proofs of difficult new theorems. An easier preparatory task is to use machines to formalize proofs of known theorems.(Wang, 1960, 2)

Wang distinguishes the automated search for genuinely new mathematical results from the formalization of known theorems. We are interested in this second practice. Such work, we urge, provides a fascinating glimpse into the practice of mathematical argumentation.

Early research in the field of *theorem proving*—the search for proofs (or disproofs) of mathematical claims—has led to rather sophisticated techniques and impressive milestones. It is standard to divide automated theorem proving (in which computers search more or less autonomously for proofs, models, refutations, etc.) from interactive theorem proving (in which the emphasis is on the construction of proofs, assisted by a machine). Although there are some precursors of interactive theorem proving going back to the earliest days of modern computers, the field began to pick up steam mainly in the 1960s and 70s: early important projects include AUTOMATH by N. G. de Bruijn in Eindhoven, the Netherlands (de Bruijn, 1980), SAD (System for Automated Deduction) in Kiev, Ukraine (Verchinine et al., 2007), and MIZAR in Bialystok, Poland (Matuszewski and Rudnicki, 2005). (Our account of the early history of interactive theorem proving must, for lack of space, be cut short.) The products of these systems that are most interesting to us are their formal languages for reconstructing the mathematical vernacular, that is, the informal though highly stylized, even slightly rigid, parole used by mathematicians when communicating mathematical proofs. In the ends, these recontructions of mathematical vernacular are themselves formal languages, but they are far from simply being "raw" formalisms such as natural deduction or sequent calculus.

For a more thorough account of the state of the art, one can consult the excellent survey articles by Wiedijk (Wiedijk, 2008), Hales (Hales, 2008), and Harrinson (Harrison, 2008). Here we focus on the aspects of formal mathematical proofs that may be of interest to those working in argumentation theory and mathematics.

The use of computers in mathematical theorem proving is becoming increasingly important. One can distinguish two directions: *proof search* and *proof checking*. Proof search suffers from well-known complexity problems and has so far had only limited success solving general mathematical problems. On the other hand, while it might be very hard to find a proof, to *check* a proof for correctness is, in general, of lower complexity (although it can be rather technical and long).

Wiedijk (Wiedijk, 2006) surveys a corner of the state of the art. He presents seventeen theorem provers and evaluates them in a uniform way: to prove that $\sqrt{2}$ is irrational. Wiedijk's survey gives interesting insight into the state of the art of theorem proving, strongly emphasizing proof check (and less strongly proof search). Wiedijk presents a six line proof of the famous theorem, taken from the classical textbook of Hardy and WriteHardy and Wright (1960, 39 f.):

> The traditional proof ascribed to Pythagoras runs as follows. If $\sqrt{2}$ is rational, then the equation
>
> $$a^2 = 2b^2$$
>
> is soluble in integers a, b with $(a,b) = 1$. Hence a^2 is even, and therefore a is even. If $a = 2c$, then $4c^2 = 2b^2, 2c^2 = b^2$, and b is also even, contrary to the hypothesis that $(a,b) = 1$.

This proof should be understandable by anyone with basic mathematical knowledge. Emphasizing proof checking, Wiedijk writes (2006, 3): "Ideally, a computer should be able to take this text as input and check it for its correctness." Insofar as Wiedijk means that Hardy and Wright's proof *needs* to be checked for correctness, we consider this perspective misleading. Of course, the correctness of a proof is essential, but that doesn't mean that the correctness of a proof is always in doubt, or that mathematical proofs require (repeated) verification. In fact, the proof above, as a textbook proof, has surely another objective besides simply displaying the logical correctness of its conclusion. On the one hand, it should *convince* the reader of the truth of the proven theorem; on the other hand it should provide text which is *memorable*, that can be reproduced whenever needed. For the moment, let us follow the line of proof checking, as this might be an important task when we are in doubt about the truth of a theorem or about a particular proof of a theorem. When Wiedijk presents the results which computer-aided theorem provers provide for the irrationality of $\sqrt{2}$, these results might be very well checkable, but the proofs themselves are far from being acceptable for a human reader.[8] In fact, to guarantee correctness, they have to take into account too many details, details which a mathematician does not like to see exposed in the proof. This was formulated by Scott as follows:

[8] "We can also see clearly from the examples in this collection that the *notations* for input and output have to be made more human readable" (Scott, 2006, *viii* f., in the foreword of Wiedijk, 2006).

> For verification (. . .) *checkable proofs* have to be generated and archived. Computers are so fast now that hundreds of pages of steps of simplifications can be recorded even for simple problems. Hence, we are faced with the questions, 'What really is a proof?' and 'How much detail is needed?' (Scott, 2006, *ix* f.)

That formalized proofs are not a good answer to the former question is argued by several authors. For instance, Avigad, working from historical case studies of proofs in elementary number theory, concludes that

> Standard models of deduction currently used in mathematical logic cannot easily support the type of analysis [of proofs] we are after. (Avigad, 2006, 131).[9]

We agree with Scott that formalized or computer-checked proofs challenge our notion of proof with respect to *proof representation*. How do we represent proofs in such a way that a computer could understand them, while still being practical and useful for humans? And can we trust formal proofs in the same way that we can trust "human proofs"? (See Rehmeyer (Rehmeyer, 2008)for a further discussion.) And when is a proof really a proof, anyway?

This already holds "in the small" for the proofs of the irrationality of $\sqrt{2}$ presented in (Wiedijk, 2006). It surely also holds "in the large" when we come to the controversial case of the computer-aided proof of the four color theorem, which suffers from a huge number of case distinctions checked by computer, but which were not (and most likely could never) be checked by a human mathematician. And it also holds for the case of the alleged proof of the Kepler conjecture(Hales, 2005). But, while we said that the verification of Hardy and Wright's proof above is not an issue, the verification of these proofs *is* at issue, and is valuable. For Appel and Haken's original 1976 proof (Appel and Haken, 1977) of the four color theorem this is mentioned by Thomas, who writes in an informal explanation of the second proof:

> We have in fact tried to verify the Appel-Haken proof, but soon gave up. Checking the computer part would not only require a lot of programming, but also inputting the descriptions of 1476 graphs, and that was not even the most controversial part of the proof.Thomas (2007)

So, Robertson, Sanders, Seymour, and Thomas came up with a new proof (Robertson et al., 1997). It is still performed by computer aid, but it reduces the cases from 1476 to 633. While this is still a number which cannot be checked "by hand", no one would deny that it is an improvement, i.e., that this proof is clearly *better* than the original one. But with respect to the verification, now we have the possibility to verify—by computer aid—the programs involved.

The question of verification of large or otherwise controversial proofs was reignited recently by Hales when he launched a program to verify "formally", i.e., by computer aid, his proof of the Kepler conjecture. Hales is engaged in his project because the initial verification attempt *by mathematicians* of his solution to the Kepler conjecture led only to "99% certainty" about the correctness of his proof (Hales,

[9] Avigad actually proposes *methods*—which correspond, for instance, to *tactics* in the ISABELLE theorem prover—as alternatives (cf. Avigad, 2006).

2005). While the situation seems to be in principle similar to the proofs of the four color theorem, the new aspect is that the author indeed tries to convince the mathematical community of the correctness of his proof *entirely* by formal and computer verified means.

What is the difference between proving and checking a proof? When is a proof a proof, anyway? Is there a social aspect to whether a proof is a proof? (We refer the reader to Heintz (Heintz, 2003), and to Löwe, Müller, and Müller-Hill (Löwe et al., 2010) for a discussion of further epistemological issues in formal proof.)

As a consequence of the philosophical discussion of Appel and Haken's original proof of the four color theorem (there are many sources: Tymoczko, Tymoczko (1979), Detlefsen and Luker Detlefsen and Luker (1980), Teller Teller (1980), MacKenzie (MacKenzie, 1999), Arkoudas and Bringsjord (Arkoudas and Bringsjord, 2007), Bassler (Bassler, 2006)), Prawitz stresses—with reference to Teller—the importance of distinguishing proof from its verification:

> That one has verified that a proof is a proof [. . .] is therefore not a part of the proof. That is not to say, of course, that it is not wise to check one's proof; as Hume rightly remarks, the confidence in a proof increases when one runs over it. But the checking does not add anything to the proof itself (Prawitz, 2007, 89).

From this perspective, it is clear that Hales's verification project should concern *only* the confidence one might have in his proof, not the proof itself. But what will be the status of the proof even after it is verified in the lines that Hales proposes?

In fact, this situation is not as new as it might look; it didn't just emerge from the use of computers. There actually is one other instance in which a modularization in numerous cases was carried out by just a large number of mathematicians: the classification of finite groups. In fact, its "proof" (which was, in part, also done by computer aid) has an interesting history with respect to its correctness and its acceptance (Aschbacher, 2004).[10] What distinguishes it most from the computer proofs mentioned above, is the fact, that the different cases might be considered as interesting in their own right, i.e., for the study of a particular group (or class of groups). A study of the single cases in the proofs of the four color theorem does not provide any such mathematical surplus value.

The lesson to take away from the reception of the proof of the four color theorem and the reception of the classification of finite simple groups is that mathematicians still seem to prefer proofs checked by other (human) mathematicians. Even if the proof is so complex that no single mathematician can check it, it is preferred that the mathematical community as a whole cooperates in carrying out the verification. See also Buss, who divides proofs into formal and social (Buss, 1998) In the application of computers to proving properties of computer programs, one can also find controversy; Demillo, Lipton, and Perlis's paper (De Millo et al., 1979) on the so-

[10] "I have described the Classification as a theorem, and at this time I believe that to be true. Twenty years ago I would also have described the Classification as a theorem. On the other hand, ten years ago, while I often referred to the Classification as a theorem, I knew formally that that was not the case, since experts had by then become aware that a significant part of the proof had not been completely worked out and written down" (Aschbacher, 2004, 737 f.).

cial nature of proof is a famous gauntlet thrown down in the discussion about these matters. MacKenzie (MacKenzie, 2004) provides a comprehensive discussion.

1.3 Checking a (formal) proof

The problem of checking a mathematical proof can sometimes be surprisingly complex. It seems that for many ordinary mathematical proofs the process of checking the proof occurs simultaneously with a reconstruction of the proof. Lakatos expresses this thus:

> Often the checking of an *ordinary* proof is a very delicate enterprise, and to hit on a 'mistake' requires as much insight and luck as to hit on a proof.(Lakatos, 1976)

But checking proofs even in a formal setting can also be a delicate enterprise, as well. How?

One can view an argument for a claim as a structure that specifies how claims of the argument are justified by various moves. We begin with an initial thesis, and then make inferential moves from it, making in turn additional claims. Each step we make transforms the thesis to be proved (and possibly introduces new theses) into a different claim. We discharge some obligations and possible introduce others along the way. The argument can be said to be successful if all our steps are the result of sound applications of rules of inference and the thesis to be proved at the end of the argument is acceptable.

Lest we slip into an infinite regress à la Carroll (Carroll, 1895), we need to agree to our rules of inference. Thus, if the thesis is B and we have established A and $A \rightarrow B$, we need to agree that:

- An application of modus ponens to the two claims already established (A and $A \rightarrow B$) yields B,
- Modus ponens is an acceptable rule of inference, and
- The occurrence/utterance of B that is obtained by applying modus ponens is "the same" as the B that we set out to establish.

The first item pins down the premises of an application of modus ponens. The second item is meant to rule out the possibility that the acceptability of modus ponens becomes itself a disputable issue in the argument. The third item, like the first, is meant to pin down the issue under discussion; it won't do if, establishing A and $A \rightarrow B$, we nonetheless reject the conclusion of the application of modus ponens to these two premises because the conclusion B now differs from the B that we set out to establish.[11]

[11] This is not to say that such phenomena are not worth studying. One way of coming to grasp the meaning of a statement is by arguing with it; we may find, for example, that if we have reached an unacceptable conclusion through sound reasoning from premises that we accept, we find ourselves having reached a better understanding of the conclusion. Thus the B we reach at the end is, in some sense, different from the B (in $A \rightarrow B$) from which the argument commenced. Such a phenomenon

To some extent, one could say that there is little at issue when it comes to the problem of checking a formal mathematical proof. We simply choose some target formalism in which to reconstruct the proofs, such as a standard Hilbert-style calculus or a Fitch-style natural deduction calculus. We might prefer calculi that are complete for the notion of logical consequence in which we are interested (often, but not always, classical first-order logic).[12] We then "formalize" the argument in the chosen formalism, producing some kind of figure d (graph, tree) representing the initial argument. The initial argument is then checked if and only if d is a legal figure according to the rules of the proof formalism that we started with. Showing that d is legal is, generally, an entirely mechanical matter.

For example, in a Hilbert-style calculus, d is a finite sequence $\langle A_1, A_2, \dots, A_n \rangle$ of logical formulas. The question of whether d is a legal derivation of a formula ϕ from assumptions Γ consists of showing that d terminates with ϕ and that for each term A_i of the sequence, that either

- A_i is a member of Γ
- A_i is an axiom (which amounts to simple pattern matching: does A_i have the form $\phi \vee \neg\phi$? Does A_i have the form $\phi \rightarrow (\psi \rightarrow \phi)$?)
- There exist earlier terms A_{i_1} and A_{i_2} of the sequence such that A_{i_2} is $A_{i_1} \rightarrow A_i$. This is just another way of saying that A_i is obtained from A_{i_1} and A_{i_2} by modus ponens.

For other formalisms, e.g., Fitch-style natural deduction or Gentzen-style sequent calculus, checking whether some figure d is legal according to the formalism is likewise quite straightforward.

Yet it often happens that among proof formalisms, there is a balance between the complexity of verifying that a figure purporting to be a legal derivation really is legal, and the length of the proofs. Thus, in a Hilbert-style calculus, it is trivial indeed to check that a given sequence of formulas is a legal Hilbert-style derivation; but the length of the legal sequences, as one considers derivations of increasingly non-trivial mathematical results, grows quite rapidly. (For a thorough systematic discussion of this and related issues of proof complexity, see Orevkov, 1993.) At seems, moreover, that such proofs are unsatisfactory because they diverge significantly from mathematical practice. What is wanted is a formalism that tries to be more faithful to the practice of mathematical argumentation, while still being sufficiently delimited that one can compute with the formalizations.

There are, we submit, such formalisms, balancing ease of use with the practical need that checking whether a figure is a legal derivation is fast. Later we shall see some example proofs written in one of them.

might be understood as argument-based discovery of meaning. Such argumentation—which might be seen as fallacious—is present in mathematics, but we shall not consider it here.

[12] "Classical" means that the law of excluded middle is assumed to be valid: the disjunction $\phi \vee \neg\phi$ is assumed to hold for any formula ϕ.

1.3.1 Formal proofs: Advantages and opportunities

One kind of argument appraisal that is available in the formal setting that is not easily available in the informal setting is the question of what an argument depends on. Essentially any contentful argument takes something for granted. It appeals explicitly to some background knowledge, or perhaps makes certain assumptions implicitly or carries out parts of the argument without any justification. Even certain mathematical definitions might take something for granted. For example, the definition of the real number π as the ratio of a circumference of a circle to its radius, on the surface, simply defines a function from *particular* circles to the set of real numbers. No one doubts that the notions "circumference of a circle" and "diameter of a circle" vary from circle to circle (that is, distinct circles can have distinct circumferences and radii), so without any further analysis all the definition of π gives us is yet another quantity that varies from circle to circle:

$$\forall\gamma(\text{circle}(\gamma) \to (\pi(\gamma) = (\text{circumference}(\gamma)/\text{diameter}(\gamma))))$$

But in fact in standard Euclidean geometry the quantity $\pi(\gamma)$ does *not* vary with γ. Thus, in a fully formal treatment of Euclidean geometry, one would have to establish the theorem

$$\forall\gamma,\gamma'[\pi(\gamma) = \pi(\gamma')].$$

Thus, in a fully formal development of Euclidean geometry, one would expose this implicit dependency of a definition on some of the axioms of Euclidean geometry.

Nonetheless, modern interactive theorem provers generally provide some kind of support for omitted inferences. Which inferences are omitted? We see an interesting formal analogue of the subject taken up by Fallis, concerning intentional gaps in mathematical proofs (Fallis, 2003). In what sense are there gaps in a formal, computer-checked proof?

At the end of formalization, one typically has, at least in principle, an utterly formal proof of a theorem, logically correct down to all details, down to the axioms of whatever background theory one is working with. However, a completely formal proof, for a theorem of any mathematical substance, would be unmanageable. Let us be clear that when working with interactive theorem provers, one does not typically work with "totally" formal proofs that, say, proceed *only* by introduction and elimination rules for quantifiers and connectives. (This is in accordance with the usual notion of *analytic proof*, which proceeds by an analysis of the structure of the claim to be proved and the structures of the assumptions used to prove it. By contrast, a *synthetic proof* brings in some new ingredients that are not formally contained in the statement to be proved.) It is well know that these are simply too big. One uses the computer not to assist in the drudgery of simply storing a large derivation figure in its memory and manipulating it with somewhat greater facility than would be the case were one to just use pencil and paper. Instead, the standard practice is that one works with a formal language that sits above a "totally" formal language. One writes proofs in the intermediate formal language that, in some sense, can be compiled into a totally formal proof. Thus, modern interactive theorem provers typically provide mechanisms for suppressing some inferences. One sees here a computable version of Azzouni's derivation-indicator view. An "informal" or ordinary mathematical proof is said to be an indicator of a formal derivation. Likewise, proofs conducted with modern interactive theorem provers, even though they are rather more formal than informal proofs, can likewise be seen as indicators of totally formal derivations.

This does not mean that a proof constructed with an interactive theorem prover is, in some interesting sense, *informal*. Rather, it is formal, but with some gaps that can be, as it were, computably traversed. That is, gaps generally represent proof search problems; the traversability of a gap means that there is a solution to the proof search problem. Consider, for example, a proof in the MIZAR system for the following elementary set-theoretic fact[13]:

```
for x, X, Y being set holds
  x in X \+\ Y iff not (x in X iff x in Y)
proof
  let x, X, Y be set;
  x in X \+\ Y iff x in X \ Y or x in Y \ X
    by DefDisjointUnion;
  hence thesis
    by DefRelativeComplement;
end;
```

Here the claim is that a set x is in the disjoint union of two sets X and Y (`x in X \+\ Y`) iff it is not the case that x is in X iff x is in Y (`not(x in X iff x in Y)`). The proof uses the definition of disjoint union (`DefDisjointUnion`),

```
definition
  let X, Y be set ;
  func
    X \/ Y -> set
  means :DefDisjointUnion:
  for x being set holds x in it iff (x in X or x in Y);
```

which is defined in terms of relative complement (`X \ Y`), defined as

```
definition
  let X, Y be set ;
  func X \+\ Y -> set
    equals
   (X \ Y) \/ (Y \ X);
```

The disjoint union of X and Y is the union $(X \setminus Y) \cup (Y \setminus X)$ of the relative complement of X from Y and Y from X), and the definition of relative complement itself, which is

```
definition
  let X, Y be set ;
  func X \+\ Y -> set
    equals
   (X \ Y) \/ (Y \ X);
```

[13] http://mizar.org/version/current/html/xboole_0.html#T1

The proof is three steps long: the initial “let” statement, the inference from the definition of disjoint union, and the final inference (`thesis`) from the definition of relative complement. One might wonder what the trouble is all about; see the discussion of what counts as an “obvious” inference in Section 1.4 for an explanation of why it’s necessary, at least for the case of MIZAR, to spell out an argument. Taking for granted the need to articulate these steps in the proof, one might wonder whether this is *really* a formal proof. Indeed, it may fail to adhere to, say, the requirements of a Fitch-style natural deduction system. The “let” of the first three steps introduces three variables (`x`, `X`, and `Y`) all at once, in one step, rather than one at a time. Another way in which the MIZAR proof could fail to strictly adhere to the requirements of a totally formal proof is that it fails to start with the definition of disjoint union, considered as the universal claim

$$\forall X \forall Y [X \oplus Y = (X \setminus Y) \cup (Y \setminus X)]$$

and instantiate it for the terms of interest, and then apply a rule of equality to transform the given claim in terms of `\+\` into one involving union and relative complement. A complete Fitch-style natural deduction proof of the claim would proceed as in Fig. 1.1. This Fitch-style deduction takes dozens of steps. We cannot claim that there is no shorter proof. Still, it is implausible to us that there is a Fitch-style deduction of the same theorem from the same premises that is much shorter than this one. The Fitch proof even economizes in some ways: we took as an axiom an instance of the propositional tautology

$$\neg(p \leftrightarrow q) \rightarrow [(p \wedge \neg q) \vee (\neg p \wedge q)]$$

in step 1 and an instance of

$$(p \wedge q) \rightarrow (q \wedge p)$$

in step 2 as premises. The principle assumptions are of course 3, 4, and 5, which are definitions of three mathematical concepts. In a Fitch-style natural deduction formalism in which (instances of) this formula are axioms, then no further work is needed. If this formula is not an axiom, then of course a proof of it must be given, so the proof would need to be even longer. We have also economized by allowing multiple-variable instantiations for universal formulas, that is, permitting the inference of

$$\phi[x_1, x_2, \ldots, x_n := t_1, t_2, \ldots, t_n]$$

from

$$\forall x_1 \forall x_2 \cdots \forall x_n \phi,$$

where $n \geq 1$, in a single step (where $\phi[x_1, \ldots, x_n := t_1, \ldots, t_n]$ denotes the simultaneous substitution of the term t_k for the variable x_k, $1 \leq k \leq n$). Thus, depending on precisely how restricted the natural deduction system is, the deduction easily approaches 50 steps.

Contrast the 3-step formal proof earlier, from which the Fitch-style derivation came, with the totally formal Fitch-style derivation. The point is that there is no

1	$\neg(x \in X \leftrightarrow x \in Y) \to [(x \in X \wedge x \notin Y) \vee (x \notin X \wedge x \in Y)]$	Tautology
2	$(x \notin X \wedge x \in Y) \to (x \in Y \wedge x \notin X)$	Tautology
3	$\forall X \forall Y \forall x[x \in X \cup Y \leftrightarrow x \in X \vee x \in Y]$	Definition of union ($\cup$)
4	$\forall X \forall Y \forall x[x \in X \setminus Y \leftrightarrow (x \in X \wedge x \notin Y)]$	Definition of relative complement ($\setminus$)
5	$\forall X \forall Y [X \oplus Y = (X \setminus Y) \cup (Y \setminus X)]$	Definition of disjoint union ($\oplus$)
6	$X \oplus Y = (X \setminus Y) \cup (Y \setminus X)$	$\forall$E, 5
7	$x \in X \setminus Y \leftrightarrow (x \in X \wedge x \notin Y)$	$\forall$E, 4
8	$x \in Y \setminus X \leftrightarrow (x \in Y \wedge x \notin X)$	$\forall$E, 4
9	$x \in X \cup Y \leftrightarrow x \in X \vee x \in Y$	$\forall$E, 3
10	$x \in (X \setminus Y) \cup (Y \setminus X) \leftrightarrow x \in (X \setminus Y) \vee x \in (Y \setminus X)$	$\forall$E, 3
11	$x \quad X \quad Y \qquad x \in X \oplus Y$	
12	$x \in X \leftrightarrow x \in Y$	
13	$x \in (X \setminus Y) \cup (Y \setminus X)$	=-E, 11, 6
14	$x \in (X \setminus Y) \vee x \in (Y \setminus X)$	$\leftrightarrow$-E, 13, 10
15	$x \in X \setminus Y$	
16	$x \in X \wedge x \notin Y$	$\leftrightarrow$-E, 15, 13
17	$x \in X$	$\wedge$E, 16
18	$x \in Y$	$\leftrightarrow$-E, 16, 12
19	$x \notin Y$	$\wedge$E, 16
20	$\perp$	$\perp$-I, 18, 19
21	$x \in Y \setminus X$	
22	$x \in Y \wedge x \notin X$	$\leftrightarrow$-E, 21, 8
23	$x \in Y$	$\wedge$E, 22
24	$x \in X$	$\leftrightarrow$-E, 23, 16
25	$x \notin X$	$\wedge$E, 24
26	$\perp$	$\perp$-I, 24, 25
27	$\perp$	$\vee$E, 14, 15–20, 21–26
28	$\neg(x \in X \leftrightarrow x \in Y)$	$\neg$I, 12–27
29	$x \in X \oplus Y \to \neg(x \in X \leftrightarrow x \in Y)$	$\Rightarrow$I, 11–28
30	$\neg(x \in X \leftrightarrow x \in Y)$	
31	$(x \in X \wedge x \notin Y) \vee (x \notin X \wedge x \in Y)$	$\Rightarrow$E, 1, 30
32	$x \in X \wedge x \notin Y$	
33	$x \in X \setminus Y$	$\leftrightarrow$-E, 32, 7
34	$x \in X \setminus Y \vee x \in Y \setminus X$	$\vee$I, 33
35	$x \in X \oplus Y$	=-E, 34, 6
36	$x \notin X \wedge x \in Y$	
37	$x \in Y \wedge x \notin X$	$\Rightarrow$E, 2, 36
38	$x \in Y \setminus X$	$\leftrightarrow$-E, 37, 8
39	$x \in X \setminus Y \vee x \in Y \setminus X$	$\vee$I, 38
40	$x \in X \oplus Y$	=-E, 39, 6
41	$x \in X \oplus Y$	$\vee$E, 31, 32–35, 36–40
42	$\neg(x \in X \leftrightarrow x \in Y) \to x \in X \oplus Y$	$\Rightarrow$I, 30, 41
43	$x \in X \oplus Y \leftrightarrow \neg(x \in X \leftrightarrow x \in Y)$	$\leftrightarrow$-I, 29, 42
44	$\forall x \forall X \forall Y [x \in X \oplus Y \vee x \in Y \setminus X \leftrightarrow \neg(x \in X \leftrightarrow x \in Y)]$	$\forall$I, 11–43

Fig. 1.1 Fitch-style deduction of $\forall x \forall X \forall Y [x \in X \oplus Y \vee x \in Y \setminus X \leftrightarrow \neg(x \in X \leftrightarrow x \in Y)]$

need to write down all those 40+ steps of the Fitch-style derivation, because most of them can be *computed*. The above MIZAR proof might even be just what we want, since, apart from the first step of instantiating the variables, the two steps are, essentially:

1. Apply the definition of disjoint union, then
2. Apply the definition of relative complement.

This seems to be is the heart of the matter, and the proof does not diverge from that. By contrast, it is not apparent what the "heart" of the corresponding Fitch-style proof is, since we had to carry out various instantiations of universal premises and even take a detour through a propositional logic.

1.4 What inferences are "obvious"?

When giving a mathematical proof, one has to decide what to say and what can go without saying. Learning the norms for communicating proofs is an important part of acquiring knowledge of mathematics. (One might even see mathematics as an instance of Toulmin's fields (Toulmin, 2003).) It is a recurring question that teachers of mathematics face: "Which steps should be included? What steps can I omit?" Rarely (if ever) do we see all steps of a mathematical proof, if by "step" we understand a single application of a rule of inference in some conventionally accepted formalism akin to the Fitch-style derivation given in the previous section. Indeed, if a student were to give a totally formal derivation as a solution to a mathematical problem, we would rightly feel that the student has, in some sense, failed, despite the inarguable validity of his solution. In the opposite extreme, teachers of mathematics can surely recount occasions where a student simply asserts a complicated statement that, by reasonable lights, cannot be simply asserted, since it needs justification, at least in the context of instruction.

At issue is the problem of distinguishing, in a context, which mathematical claims need to be justified from those that can be simply granted. (A related topic is the problem of characterizing persuasiveness of certain mathematical moves. For more on the subject, see the contribution by Inglis and Mejía-Ramos in this volume.) A full answer evidently requires a classification of the various contexts in which mathematical claims are made. A mathematics teacher might reject a student's unjustified claim of an equation, while accepting, only five minutes later, the very same equation put forward by a colleague in the mathematics department. The teacher is not being duplicitous because the contexts of mathematical acceptability in the two situations are different.

What about in the formal context? The choice of formalism determines what claims count as justified and those that require justification. The result is extreme: *anything that is not an axiom requires justification*. It is not so bad that non-trivial propositions require proof. What is worse is that it often happens that even "trivial" statements, so long as they are not axioms, require proof. And sometimes trivial statements turn out to be not so trivial after all, in that they apparently require unexpectedly long formal proofs. Moreover, the definition of formal derivation requires that every step in the proof that is not an axiom be the result of an application of a rule of inference.

If we are working cooperatively with a computer, though, the answer to the question of which inferences need justification can be deferred somewhat to the computer. The strength of the mechanisms for doing automated reasoning will have a powerful influence the proofs. Rudnicki presents the situation thus:

> The core of an automatic proof-checking system is a decision procedure for accepting/rejecting presented inferences. A rejection does not mean that an inference is logically invalid, it simply mirrors the fact that the proof-checker was unable to certify the inference's correctness. Certainly, an invalid inference has to be rejected. Using a proof-checker is similar to a discussion between humans. One admits that one does not see why a conclusion follows from premises (even if it does in fact), but one agrees quickly that the adversary is right after being given additional explanation. The criterion for acceptance/rejection of valid logical inferences in a proof-checking system is said to define a class of 'obvious' inferences in the system.(Rudnicki, 1987, p. 383).

We shall follow the terminology used by Rudnicki (who is reusing the term introduced by Davis) and discuss now the problem of *what inferences are obvious?* The problem is to try to give a formal or computable account of an informal notion. We will, of course, not succeed in full, but we believe that we can learn about the notion of obviousness and attendant features of mathematical argumentation by studying it through a formal lens.

As Rudnicki suggests, we are interested in *sound* obvious inference. (To precisely specify what we mean by "sound", we should specify a logic, and there seems to be room for non-classical logics. For our purposes, let us stick to classical first-order logic.) Just because something is judged to be non-obvious does not mean that it is false or invalid. Moreover, we are all too aware of claims put forward as "obvious" but which turn out to be false or otherwise unacceptable. Calling something "obvious" can sometimes run the risk of being nothing more than a thinly-veiled appeal to authority, argument by intellectual boasting or belittlement (it's obvious to me—why isn't it obvious to you?), or premature abandonment ("it's obvious" might just amount to saying that we don't have time to argue or that the arguer isn't willing to argue).

The extreme solution of stipulating that all (valid) inferences are obvious seems unacceptable. With this understanding of obviousness, any mathematical inference whatsoever would be acceptable without justification. This extreme solution might not even be well-defined. After all, what notion of validity should we use? Should we use classical logic or non-classical logic? What is the characterization of validity? Should the notion of validity be syntactic or semantic? If we were worried about consistency (is it possible that incompatible claims are both obvious?), we might try to restrict obviousness so that only true conclusions are accepted—but what claims are true? Evidently not all true claims are obvious and at least some require proof; indeed, often we don't even *know* that a mathematical claim is false until we try to prove it, tentatively accepting it as true, and realize only in our failure to prove it that it is not even true.

We have already discussed the other extreme, in which nothing is obvious except what is formally an axiom. The result here is likewise unacceptable, because we then have that only the most brutally obvious claims are accepted, and we would give up quickly. One could take as axioms an extremely large set of principles; but then we collapse to the other extreme in which essentially everything is obvious.

What is wanted is a notion of obvious inference that allows us to omit *some* (valid) reasoning, but not too much. The notion should also be practical: if we are interested in doing much mathematical reasoning formally, checking whether a step in the proof is an obvious inference should be done quickly. The notion of obviousness should therefore have a low computational complexity.

Davis has taken on the problem of characterizing, proof-theoretically, the notion of an "obvious" inference (Davis, 1981). The context from which Davis devised his characterization of obvious inference was a project of natural deduction at Stanford. Natural deduction arguments à la Gentzen, Fitch, or Suppes (as with any serious proof formalism) can be rather tedious. The heart of an informal argument is often obscured or diffused if one adheres strictly to the requirements of the formalism. In formal contexts one wants a rule of inference that would allow one to dispense with certain tedious details.

Here is a precise reformulation of Davis's proof-theoretic definition of "obvious inference" in first-order classical logic:

Definition 1. A logical formula ϕ is an **obvious logical consequence** of assumptions Γ if there is a Herbrand proof of ϕ from Γ in which each quantified formula of Γ is instantiated at most once.

The precise definition of Herbrand proof will not be given here; see (Davis, 1981) or (Harrison, 2009). The idea is that in drawing an obvious logical inference, it is ruled out to use multiple instances of quantified formulas in Γ. Once one has chosen, for each quantified formula α in Γ, an instance α^* (one may elect not to instantiate α at all), then one has to formally derive ϕ from Γ and the instances $\alpha_n^*, \ldots$ using only a "light" form of first-order reasoning and propositional calculus.

The notion of obvious logical inference, as defined by Davis, clearly does not characterize how we use the term "obvious" in the context of ordinary reasoning. A consequence of the proposal is that quite a lot of propositional inferences get classified as "obvious". This seems to accord with our intuitions in cases such as

q because p and $p \to q$

but fail for cases such as

$((p \to q) \to p) \to p$

(a famous classical validity known as Peirce's formula), or

p because X is unsatisfiable,

where X is a large, complex unsatisfiable set of propositional formulas. Davis's proposal also fails to account for such first-order inferences as:

$\phi(f(f(a))$ because $\phi(a)$ and $\forall x[\phi(x) \to \phi(f(x))]$,

because we need to instantiate the universal premise $\forall x[\phi(x) \to \phi(f(x))]$ twice (once with a, and again with $f(a)$).[14]

Still, it is a valuable proposal because of its conceptual simplicity and practical efficiency. In general, if we have some quantified formulas Γ and a conclusion ϕ, deciding which instances of formulas in Γ to choose can quickly lead to a vast number of possibilities. Davis's proposal limits the search for instances: *choose at most one instance each time*. When a conclusion ϕ is not an obvious logical consequence of background assumptions Γ, then the inference is "too complicated" to be checked by a machine, and the human formalizer needs to supply more information.

Following on Davis's work, Rudnicki (Rudnicki, 1987) has outlined some problems with Davis's proposal and has offered a second mathematical characterization of the notion of obvious inference.

As mentioned earlier, the problem faced by the designer of an interactive theorem proving system is to sail between an extremely dense but extremely fast notion of acceptable inference, as opposed to trying to make as much as possible obvious by devoting arbitrary computational resources to determining whether an inference is obvious. The effect of this decision is that the class of proofs will tend to be extremely long on the one hand, and maximally concise on the other.

One might say that the preference here is clear: one should go for the most concise proofs possible! The problem is that computer proof search for interesting logics is a rather difficult computational task. The search spaces for theorem proving problems tend to be extremely large, even intractable, so that devoting increasing amounts of computational power often has surprisingly little effect. It often happens, because of the sheer size of the search spaces involved, that if a theorem proving problem cannot be solved in 5 minutes, then it cannot be solved in one hour either.[15] We are thus forced by complexity to keep our aims modest.

If we vary the strength of the checker for obvious inferences, can we detect a dividing line between what is obvious, in the everyday sense of the term, from what is not obvious? Surely there must come some point (though we concede that it seems likely that there could be Sorites-type paradoxes here). For at the extreme end we could have complicated theorems of mathematics that are known to be valid consequences of some recursively enumerable set of axioms, such as those of Zermelo-Fraenkel set theory (ZF); given arbitrary computational resources to check whether an inference is "obvious", we would find that, say, the fundamental theorem of calculus gets deemed as an obvious inference from some finite set of axioms of ZF, though we would surely balk at judgment.[16]

[14] This example appears in (Rudnicki, 1987).

[15] There are many counterexamples to this general outlook. The solution, by an automated theorem prover, of the long-outstanding Robbins problem required 8 days of continuous computation (McCune, 1997).

[16] If one is not satisfied with this example, we could replace the fundamental theorem of calculus by some other significant mathematical fact, perhaps even one that has not yet been discovered.

The result is that by varying the strength of the mechanism that certifies obviousness has an effect on the intuitive *explanatory value* of the proofs. (For a further discussion of explanation in mathematics, see (Mancosu, 2000).) This is intuitively clear. Earlier we presented "$\phi(a)$ and $\forall x[\phi(x) \to \phi(f(x))]$ therefore $\phi(f(f(a)))$" as a case where Davis's notion of obvious inference fails. This case would be counted as obvious if we strengthened Davis's notion and permitted two instances of universal premises to be selected.

Here is a slightly more mathematical example of the same phenomenon, showing the effect that such a strengthening has on live mathematical proofs coming from the Mizar Mathematical Library, the curated body of formalized mathematical knowledge that has been reconstructed in the Mizar interactive theorem prover. [17] The proofs in the Mizar Mathematical Library are governed by a notion of obviousness similar to Davis's notion. (A precise definition of the class of Mizar-obvious inferences is not needed.)

Consider the following formal theorem:

```
reserve X, Y, Z for set;

Lemma1: X c= Y & Y c= Z implies X c= Z;

Lemma2: X c= X \/ Y;

Lemma3: X c= Y implies X \/ Z c= Y \/ Z;

X c= Y implies X c= Z \/ Y
proof
 assume X c= Y;
 then A1: Z \/ X c= Z \/ Y by Lemma3;
 X c= Z \/ X by Lemma2;
 hence X c= Z \/ Y by A1,Lemma1;
end;
```

The first line says that in what follows, the variables `X`, `Y`, and `Z` are sets (more precisely, they are assigned the type `set`). The next three statements are background lemmas (assigned the labels `Lemma1`, `Lemma2`, and `Lemma3`).

Lemma 1 (`Lemma1`) expresses the transitivity of the subset relation: if $X \subseteq Y$ (`X c= Y`) and $Y \subseteq Z$ (`Y c= Z`), then $X \subseteq Z$ (`X c= Z`).

Lemma 2 (`Lemma2`) expresses the fact that X is always a subset `c=` of the union of X with any other set Y (`X \/ Y`).

Lemma 3 (`Lemma3`) expresses the fact that if X is a subset of Y (`X c= Y`), then the union $X \cup Z$ is a subset of the union $Y \cup Z$ (`X \/ Z c= Y \/ Z`).

Lemmas 1, 2 and 3 will be taken for granted, for the sake of discussion. That is, the text above is not acceptable to Mizar as written, because all three lemmas are not Mizar-obvious and there require Mizar-proof. Our interest is the final result (the `theorem`) of the Mizar text fragment, which expresses the simple result that if X is a subset of Y (`X c= Y`), then X is a subset of the union $Z \cup Y$ for any set Z (`X`

[17] We thank Artur Korniłowicz for this example

`c= Z \/ Y`). Unlike the case for the three lemmas, a proof of this fact is provided. Let us proceed through it.

We are carrying out a natural deduction-style proof of an implication (`X c= Y implies X c= Z \/ Y`); the first step, naturally, is to assume the antecedent (`assume X c= Y`). From this assumption we get, using Lemma 3, the include `Z \/ X c= Z \/ Y`. (MIZAR is also implicitly using the commutativity of the binary union operation—note that Lemma 3 puts `Z` on the right-hand side of the union, but in the conclusion just drawn, it appears on the left-hand side of the union). The notation `A1:` is simply assigning a label to the statement just concluded; we will use the formula later in the argument by appealing to its label. The third step in the argument simply applies Lemma 2; we have from it that $X \subseteq Y \cup X$. We do not use the hypothesis of the theorem nor the previously concluded statement (whose label is `A1`) to infer the result; this follows immediately from Lemma 2 alone. The final step of the proof (followed by `hence`) is the desired conclusion: we have that $X \subset Z \cup Y$ because of

- $X \subseteq Z \cup X$ (from the previous line)
- the formula labeled `A1`, i.e., $Z \cup X \subseteq Z \cup Y$
- the formula labeled `Lemma1`, i.e., $X \subseteq Y \wedge Y \subseteq Z \to X \subseteq Z$.

This argument is optimal in the sense that no step can be removed. The MIZAR proof checker rejects all possible compressions the argument. In the maximal compression, one justifies the final theorem by simply declaring, without proof, that it follows from the lemmas, i.e.,

```
X c= Y implies X c= Z \/ Y by Lemma1, Lemma2, Lemma3;
```

one finds that the MIZAR proof checker rejects the inference. Other kinds of attempted compression, such as removing the intermediate statement `A1`, viz.

```
X c= Y implies X c= Z \/ Y
proof
 assume X c= Y;
 hence X c= Z \/ Y by Lemma1, Lemma2, Lemma3;
end;
```

or dropping the application of Lemma 2, viz.

```
X c= Y implies X c= Z \/ Y
proof
 assume X c= Y;
 then A1: Z \/ X c= Z \/ Y by Lemma3;
 hence X c= Z \/ Y by A1, Lemma1, Lemma2;
end;
```

are all rejected by the MIZAR proof checker. They are rejected because they require that multiple instances of background universal premises be taken, and this is precisely what is ruled out by the notion of (1-)obvious inference.

But what if instead of using the notion of 1-obviousness, we use 2-obvious? That is, what if we permitted the proof checker to pick two universal premises, rather than one? The result is that the above proof can indeed be compressed:

```
X c= Y implies X c= Z \/ Y by Lemma1, Lemma2;
```

This is a fascinating compression. We don't even need to articulate a proof any longer; for the MIZAR proof checker, our claim is a 2-obvious consequence of Lemmas 1 and 2 alone. We don't even need the help of Lemma 3, which was essential before when we were operating under the constraint that all inferences must be 1-obvious.

To sum up, we argue that mathematical proofs should be evaluated with respect to their ability to provide answers—answers in a "digestible" form. Computer proofs are often just "indigestible", not only due to excessive information, but also because they can't be addressed by our questions.

1.5 Appraising and improving formal proofs

After completing a formal proof, one can return to it and improve it in various ways. We again see some advantages of the formal approach to mathematical proof and its implementation in an interactive theorem prover: we can evaluate an argument in ways that might be more difficult were we to insist on working with informal proofs.

For example, when constructing a formal argument, it can happen that parts of the argument play no role in the final conclusion. This can be automatically detected, thus providing a kind of mechanical detection of irrelevant reasoning.

It can also happen that in the inferences appearing in a proof, some can be safely omitted. One might say that this is an appraisal of *redundant information*. That is, a proof is valid, but it also remains valid if one takes away parts of some of the justifications. Moreover, a formal mathematical proof can also be more verbose than necessary for a proof checker. Consider: step:

```
Step1: A by Lemma1;
MiddleStep: B by Theorem1, Step1;
Step2: C by Lemma2, Theorem2, MiddleStep;
```

It can happen that we can eliminate the middle step and combine the justifications:

```
Step1: A by Lemma1;
Step2: C by Theorem1, Lemma2, Theorem2, Step1;
```

The first step (`Step1`) is unchanged. We moved justification of the middle step (`MiddleStep`) to the second step (`Step2`). The proof is now compressed by one step.

Interestingly, even when such compressions are possible, we may not always wish to carry them out, because intermediate steps might be important for our understanding of the proof; deleting such intermediate steps may decrease the comprehensibility of the proof.

1.6 Conclusion

We intended to explain how the formal view of mathematical proof, far from being a stale, old subject, gets new life breathed into it thanks to technological progress coming from automated reasoning. The problem of what inferences count as acceptable without any further justification—which we have called "obvious inferences"—might be thought to be inherently informal, but which appears quite naturally in the setting of automated reasoning. Those interested in mathematical practice and argumentation can, we hope, see that a formal approach to mathematical proof is not at odds with other analyses but complements them.

Acknowledgment

Both authors were partially supported by the ESF research project *Dialogical Foundations of Semantics* within the ESF Eurocores program *LogICCC*, LogICCC/0001/2007, and the second author was also supported by the project *Hilbert's Legacy in the Philosophy of Mathematics*, PTDC/FIL-FCI/109991/2009, both funded by the Portuguese Science Foundation (FCT).

References

Appel, K. and Haken, W. (1977). Every planar map is four-colorable. *Illinois Journal of Mathematics*, 21:439–567.

Arkoudas, K. and Bringsjord, S. (2007). Computers, justification, and mathematical knowledge. *Minds and Machines*, 17(2):185–202.

Aschbacher, M. (2004). The status of the classification of the finite simple groups. *Notices of the American Mathematical Society*, 51(7):736–740.

Avigad, J. (2006). Mathematical method and proof. *Synthese*, 153:105–149.

Azzouni, J. (2004). The derivation-indicator view of mathematical practice. *Philosophia Mathematica*, 12:81–106.

Bassler, O. B. (2006). The surveyability of mathematical proof: a historical perspective. *Synthese*, 148:99–133.

Buss, S. (1998). *An Introduction to Proof Theory*, volume 137 of *Studies in Logic and the Foundations of Mathematics*, chapter 1. Elsevier, Amsterdam.

Carroll, L. (1895). What the tortoise said to achilles. *Mind*, 4(14):278–280.

Davis, M. (1981). Obvious logical inferences. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 530–531.

de Bruijn, N. (1980). A survey of the project AUTOMATH. In Hindley, J. R. and Seldin, J. P., editors, *To H. B. Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism*. Academic Press.

De Millo, R., Lipton, R. J., and Perlis, A. J. (1979). Social processes and proofs of theorems and programs. *Communications of the ACM*, 22(5):271–280.

Detlefsen, M. and Luker, M. (1980). The four-color theorem and mathematical proof. *Journal of Philosophy*, 77(12):803–820.

Fallis, D. (2003). Intentional gaps in mathematical proofs. *Synthese*, 134:45–69.

Grabowski, A., Korniłowicz, A., and Naumowicz, A. (2010). Mizar in a nutshell. *Journal of Formalized Reasoning*, 3(2):153–245.

Hales, T. (2005). A proof of the Kepler conjecture. *Annals of Mathematics*, 162(3):1063–1185.

Hales, T. C. (2008). Formal proof. *Notices of the American Mathematical Society*, 55(11):1370–1380.

Hardy, G. H. and Wright, E. M. (1960). *An Introduction to the Theory of Numbers*. Oxford, 4th edition.

Harrison, J. (2008). Formal proof—Theory and practice. *Notices of the American Mathematical Society*, 55(11):1395–1406.

Harrison, J. (2009). *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press.

Heintz, B. (2003). When is a proof a proof? *Social Studies of Science*, 33(6):929–943.

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

Löwe, B., Müller, T., and Müller-Hill, E. (2010). Mathematical knowledge as a case study in empirical philosophy of mathematics. In van Kerkhove, B., de Vuyst, J., and van Bendegem, J. P., editors, *Philosophical Perspectives on Mathematical Practice*, number 12 in Texts in Philosophy.

MacKenzie, D. (1999). Slaying the Kraken: The sociohistory of a mathematical proof. *Social Studies of Science*, 29(1):7–60.

MacKenzie, D. (2004). *Mechanizing Proof: Computing, Risk, and Trust*. MIT Press.

Mancosu, P. (2000). On mathematical explanation. In Grosholz, E. and Berger, H., editors, *Growth of Mathematical Knowledge*, pages 103–119. Kluwer.

Matuszewski, R. and Rudnicki, P. (2005). MIZAR: The first 30 years. *Mechanized Mathematics and Its Applications*, 4:3–24.

McCune, W. (1997). Solution of the Robbins Problem. *Journal of Automated Reasoning*, 19(3):263–276.

Netz, R. (2003). *The Shaping of Deduction in Greek Mathematics: A Study in Cognitive History*. Cambridge University Press.

Orevkov, V. P. (1993). *Complexity of proofs and their transformations in axiomatic theories*, volume 128 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI. Translated by Alexander Bochman from the original Russian manuscript, Translation edited by David Louvish.

Portoraro, F. (Fall 2008). Automated reasoning. In Zalta, E. N., editor, *Stanford Encyclopedia of Philosophy*.

Prawitz, D. (2007). Proof verifying programs and programs producing proofs—a conceptual analysis. Presented at ”Workshop on Deduction, Computation, Experiment. Exploring the Effectiveness of Proofs”, in Bologna, Italy, 3–4 April,2007.

Rav, Y. (2007). A critique of a formalist-mechanist version of the justification of arguments in mathematicians' proof practices. *Philosophia Mathematica*, 15(3):291–320.

Rehmeyer, J. (2008). How to (really) trust a mathematical proof. *Science News*.

Robertson, N., Sanders, D. P., Seymour, P. D., and Thomas, R. (1997). The four colour theorem. *Journal of Combinatorial Theory. Series B*, 70:2–44.

Rudnicki, P. (1987). Obvious inferences. *Journal of Automated Reasoning*, 3(4):383–393.

Scott, D. (2006). Foreword. In Wiedijk, F., editor, *The Seventeen Provers of the World*, volume 3600 of *Lecture Notes in Computer Science*, pages vii–xii. Springer.

Teller, P. (1980). Computer proof. *Journal of Philosophy*, 77(12):797–803.

Thomas, R. (2007). The four color theorem.

Toulmin, S. E. (2003). *The Uses of Argument*. Cambridge University Press, updated edition.

Tymoczko, T. (1979). The four-color problem and its philosphical significance. *Journal of Philosophy*, 76(2):57–83.

van Bengedem, J. P. (1988). Non-formal properties of real mathematical proofs. In *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*, volume 1: Contributed Papers, pages 249–254.

Verchinine, K., Lyaletski, A. V., and Paskevich, A. (2007). System for automated deduction (SAD): A tool for proof verification. In Pfenning, F., editor, *CADE*, volume 4603 of *Lecture Notes in Computer Science*, pages 398–403. Springer.

Wang, H. (1960). Toward mechanical mathematics. *IBM Journal of Research and Development*, 4(1):2–22.

Wiedijk, F., editor (2006). *The Seventeen Provers of the World*, volume 3600 of *Lecture Notes in Computer Science*. Springer.

Wiedijk, F. (2008). Formal proof—Getting started. *Notices of the American Mathematical Society*, 55(11):1408–1414.

Index