On the Structure
of
the Vernacular Language of Mathematics

Alain Prouté
Université Paris-Diderot France

Notes prepared for a conference held (in French) March 26, 2018
at École Normale Supérieure Paris
`https://www.youtube.com/watch?v=gRbKMOOMDVM&t=5262s`

Last revision of this text: February 17, 2019

Abstract

Contemporary mathematics is a cultural phenomenon that is the result of a long lasting evolution over centuries in pretty much the same way as species have evolved according to Charles Darwin's theory. In other words, our mathematics is the result of a *natural* selection of those concepts that appeared to be the best suited for the purpose of thinking abstractly. As a consequence, the structure of this game that we call *mathematics* is not immediately perceptible, its fundamental mecanisms remain essentially hidden, and it is just a fact that most mathematicians are not conscious of them even if they actually put them at work every day.

This text is not a piece of philosophy. It is the result of several decades of research around the idea of applying elementary topos theory to the design of a proof assistant, in other words to the design of a computer program able to *understand* mathematics at least from a formal point of view. Consequently, my approach is pragmatic and the goal of *implementing* these mecanisms always kept in mind.

Introduction

First of all, the goal I have in mind (as decribed in the abstract above) entails the point of view that mathematics is only an epiphenomenon of the *language of mathematics*. It does not *exist* except as such an *emergence*.([1]) Indeed, whatever we do for giving us the illusion that we manipulate mathematical objects, we are actually manipulating the expressions of the language that represent them, not these objects themselves. And the same is of course true for computers.

The most important thing we must keep in mind when we are doing mathematical logic or programming is the dichotomy between the *signifier* and the *signified*, in other words between expressions of a language and what they represent. This dichotomy was analysed by Gottlob Frege in 1892 in [3]. He answers a question that disturbed me when I was a school boy, which is that if we write down an equality such as $2 = 1+1$, we have at the same time two equal things on both sides of the equal sign and two expressions that are different (not written the same way). It should be clear that resolving this *paradox* can only be made

[1] This idea was already present in Gottlob Frege's work, in particular in his *Begriffsschrift* (*ideography*).

by understanding the above dichotomy. In other words, 2 and $1+1$ are *distinct* expressions representing the *same* mathematical object.

This example suggests that understanding what we are *really* doing when we are doing mathematics is not necessarily easy, because in general, in the way we speak, we do *as if* we were manipulating mathematical objects. Mathematicians live within the illusion (the epiphenomenon) produced by the hidden principles, and this works perfectly well. However, the situation is different if we want to program a computer so that it becomes able to *do mathematics*.

The two fundamental principles that make mathematics work are a notion of *type* (not to be confused with that of *set*) and the principle of *proof interchangeability*.([2]) It is a remarkable phenomenon that the system of types can very well remain hidden without preventing mathematics from functioning. Both principles are unknown to most mathematicians, but I hope that after reading this article they will agree that they are actually at work in the underground (invisible) mathematical world. Furthermore, I show that these two principles are absolutely mandatory. Mathematics could not be what they are without them.

These two principles have tight links with several other fundamental mathematical concepts, and in particular with the notion of *constructivism* and with the *axiom of choice*. I discuss these questions in another paper [12].

I want to thank my collegue and former PhD student Matthieu Herrmann whose collaboration on my proof assistant project led to many clarifications and improvements. This collaboration is still going on.

Contents

1 Is the mathematical vernacular a typed language ?

Mathematicians never use the word “type” for qualifying a mathematical object, at least not in a formal manner.([3]) Nevertheless, the language of mathematics is strongly typed as we shall see now, but it appears that because of the very nature of sets, speaking about

[2] This principle is also known as *proof irrelevance*. The reasons why I prefer to call it *proof interchangeability* are explained below.

[3] Model theorists use this word, but with a meaning that is different from the meaning it has in this article.

types in mathematics is not necessary. On the contrary, using types, even implicitely, is mandatory for several reasons.([4])

In this section, I don't describe a precise type system. I just suggest through several examples that a notion of type must exist.

1.1 Typed declarations

To begin with, consider a sentence such as "*Let G be a group.*", which is of current use in mathematics. This sentence is a *declaration* introducing an arbitrary group whose name is G. If we believe what Zermelo-Fraenkel set theory says, G cannot be anything other than a set, since according to this theory all mathematical objects are sets. Consequently, we could as well write "*Let G.*". However, this is clearly not the case. If a reader of a book of mathematics reads "*Let G.*", he/she will for sure ask "*What is G ?*". Notice that we also often write "*Let X be a set.*", that looks like a pleonasm in view of the assuption that all mathematical objects are sets. The simple fact that we use sentences of the form "*Let* `<symbol>` *be a* `<common name>`.", clearly shows that common names are used for representing *types of mathematical objects*.

However, of course, the declaration "*Let G be a group.*" does not declare an arbitrary set, but a set of a certain *shape* needed for being considered as a group. Nobody actually really knows what this particular shape looks like, mainly because it depends on particular conventions such as the definition of an ordered pair (a, b), generally given as the set $\{\{a\}, \{a, b\}\}$ (Kuratowski's definition), and similar conventions. A group is generally understood as a tuple $(X, \times, 1)$, where X is the underlying set of G, and where $\times$ and 1 are the multiplication and the unit of the group, and satisfying the axioms of groups. The above tuple can further be understood as the pair $(X, (\times, 1))$ whose second element is itself a pair. Again a convention, since we could as well choose $((X, \times), 1)$ instead. From this, we can *compute* the above tuple, and in order to completely compute our set G, we will have to do similar computations, in particular for understanding the *function* $\times : X \times X \to X$ as a set. Needless to say, the result is not only horrible, but also useless. This kind of remarks were made by numerous mathematicians since decades.

Actually, what matters is only that a group is a mathematical object (which is actually not a set, but which has an underlying set) whose behavior is particular. Consequently, the expression "*be a group*" in the declaration "*Let G be a group.*" does not mean that G is just a set of a particular (complicated) shape, an interpretation which depends on arbitrary and useless conventions. Nevertheless, even if this shape could be described, recognizing that G is actually a group would be quite problematic. It should be clear that the indication "*be a group*" contains more informations than the interpretation of G as a Zermelo-Fraenkel set can provide.

Historically, Zermelo, in his 1908 article [13], presenting the first version of his set theory, did not assume that all mathematical objects were sets. This *restriction* was added only in 1922 by Fraenkel [2]. That all mathematical objects are sets was also very far from the ideas of Cantor in his original 1895 article [1].

We can wonder what were the motivations of Zermelo for set theory. In his 1908 article

[4] It also appears that trying to design a language based on these same principles but within which both types and sets are visible and explicitely manipulated, yields a unnecessarily complicated paradigm. Consequently, keeping the system of types in the shadow is also mandatory.

[13], he proposes seven axioms for the theory, and between axiom III and axiom IV, he proves a theorem saying that *each set M has at least one subset M_0 that is not an element of M*, from which he deduces, not that his theory has no contradiction, but at least that known proofs of Russell's paradox of the set of all sets cannot be carried out in his theory.([5])

Indeed, eliminating Russell's paradox from Cantor's set theory was one of the main mathematical challenges of that time, and it is very likely that this was one of the most important motivations of Zermelo, maybe the first one. Trying to formalize the language of mathematics, in other words trying to establish the rules of the game of writing correct mathematics was another question, and it was not in tune with those times. Nowadays, and in particular because we have computers, the situation is different. Formalizing the language of mathematics requires to retain many more aspects of this language than Zermelo-Fraenkel set theory did. Topos theory is another more sophisticated approach, which provides a richer model, much closer to the vernacular than set theory, even if there is still an important distance between topos theory and the vernacular.([6])

Furthermore, topos theory can also formalize a notion of type. The first important fact is that types should not be confused with sets. Both concepts are present in mathematics, but they are distinct. The link between them was formalized for the first time (as far as I know) by Joachim Lambek in [6], where he exposes his theory of *dogmas*. The picture is that we have two categories at work. The first one is a dogma whose objects are types and the other one is the topos *generated* by this dogma, whose objects are sets. In other words, the concept of set should not be considered as *primitive* as it is in Zermelo-Fraenkel set theory, but as constructed from the more primitive notion of type. As we shall see, a set is essentially a pair made of a type and a predicate on this type.

1.2 Meaningless equalities and clones

There are various arguments for showing that there are types in mathematics. As an example, consider the case of elementary linear algebra. Assume we are working with real vector spaces. Hence, we can have a scalar a and a vector x. What will you say if I write $a = x$? You could perhaps say that this is *false*, but you will more likely say that it is *meaningless*, and you will be right. This is meaningless, so that saying that it is false is itself meaningless. Now, why is it meaningless ? For sure, Zermelo-Fraenkel set theory cannot give any argument in this direction, because according to it, a and x are both sets, hence are comparable. This is of course incoherent with our everyday practice, and notice that it is very desirable that a proof assistant is able to reject such an equality as meaningless.

The reason why the equality $a = x$ is *meaningless* is that a and x do not have the same *type*. All mathematicians feel that very strongly even if they cannot in general explain what a type is. Actually, it is the fact that a is called a *scalar* and x called a *vector* that creates this

[5] Which actually is true only if the theory is consistent, what Zermelo does not tell.

[6] Not all mathematicians think that mathematics is founded on Zermelo-Fraenkel set theory (with axiom of choice), even mathematicians who are known specialists of this theory. As an example, I had around 2007 a conversation with my collegue Jean-Louis Krivine, who is the author of a renowned book on Zermelo-Fraenkel set theory [5]. I asked him the question "*Is ordinary mathematics based on Zermelo-Fraenkel set theory ?*". His answer was "*No ! Usual mathematics is based on naive set theory.*", and he also added "*But naive set theory is not a theory.*". I agree with him, except that I think that topos theory *is* precisely the theory of "naive set theory".

feeling. The same phenomenon can be highlighted with operations other than equality. For example, $a \in X$ is meaningless if a is a polynomial and X a set of matrices. This is again because polynomials and matrices do not have the same type. The reader can easily find many other examples.

Maybe even more disturbing example is the following. Assume that you are teaching calculus. You define the notion of a *sequence of real numbers* as a *function from* $\mathbb{N}$ *to* $\mathbb{R}$ (as is customary). An insightful student could argue that since, according to your definition, a sequence of real numbers is *the same thing as* a function from $\mathbb{N}$ to $\mathbb{R}$, your definition is useless, and that such a sequence could very well be called a "function from $\mathbb{N}$ to $\mathbb{R}$". For sure, you will disagree with this remark, but why ? What argument do you have at hand for contradicting your student ? You can say that you want to give a different name to sequences because you want to do different things with them. This is not a bad answer, but it remains unsatisfactory from a formal point of view, because it does not formally explain the necessity of this new name.

Sequences of real numbers and functions from $\mathbb{N}$ to $\mathbb{R}$ should actually be considered as objects of distinct types. Indeed, the simple fact that we take the trouble of introducing a new name (sequence) for an already known object (function) means that we consider sequences as a *new kind of concept*, which formally means objects of a new type never seen so far. Hence, the actual meaning of your definition is that the set of sequences of real numbers is a *clone* of the set of functions from $\mathbb{N}$ to $\mathbb{R}$, not this set itself. We shall see below why *cloning* sets is mandatory. Notice that category theory is well equipped for formalizing this notion of clone because the original and the clone are nothing other than two canonically isomorphic (but distinct) objects in the appropriate category. This subtlety is sometimes reflected in the vernacular language by the fact that more cautious teachers will add a nuance to the definition of a sequence by writing it as "A sequence of real number is *given by* a function from $\mathbb{N}$ to $\mathbb{R}$." or as "*Formally*, a sequence of real numbers is a function from $\mathbb{N}$ to $\mathbb{R}$."

The next week, you introduce the notion of series of real numbers, and this time again you define them as functions from $\mathbb{N}$ to $\mathbb{R}$ (as is still customary). In other words, you are repeating the same process, that is to say that you create still another clone of the set of functions from $\mathbb{N}$ to $\mathbb{R}$, so that functions, sequences and series are of three distinct types. You have very good reasons to do so. For example, multiplication of sequences and multiplication of series are not defined by the same formula. This clearly shows the necessity to introduce new names, because multiplication has the same name (and is represented by the same symbol, or by juxtaposition) in both cases. From a formal point of view, new names are just creating new types of mathematical objects. This is why in practice (I mean for designing a proof assistant based on topos theory), the category of types (the dogma) should not be considered only up to equivalence of categories.

Before better exploring the problem created by the fact that operations (such as multiplication) have many different interpretations, we end this section by another enlightening example. It is usual practice in mathematics to define the notion of relation between two sets X and Y as a subset of the cartesian product $X \times Y$. Again, doing so we introduce a clone of the power set $\mathscr{P}(X \times Y)$, since we give the name "relation" to a "subset". The proof that we indeed create a clone is that we use expressions such as "*the graph G of the relation r*". We are speaking and writing like this for very good reasons, and for sure we would omit such expressions if experience had shown that they were not necessary. If relations were the same things as their graphs, an expression such as "the graph of a relation" would make mathematics unnecessarily complicated.

1.3 Automatic conversions

Some people will argue that, given a real number x and a complex number z, an equality such as $x = z$ is perfectly acceptable, even if x and z are clearly of different types.

First of all, x and z *must* indeed be of different types for several reasons. One of them is that we first define the set of real numbers before we define the set of complex numbers, and we cannot consider that the definition of the complex numbers constitutes a *redefinition* of the real numbers or even an *extension* of this definition. Complex numbers is clearly a *new concept*. Consequently, when we define complex numbers we introduce a new type of objects, which can also be seen by the fact that we introduce a new common name for naming these new objects.

However, we will immediatly say that a real number *can also be considered as* a complex number. If we have defined complex numbers as pairs of real numbers (written as usual $a + ib$), then we have the map $a \mapsto a + i0$ from the reals to the complexes, and we adopt the *convention* that a is an alternative valid notation for $a + i0$. In practice, this makes no problem because the ambiguity it could introduce can be easily resolved. When the compiler of a proof assistant sees the equality $x = z$, he is able to apply this *automatic conversion* from reals to complexes, and find one and only one type compatible interpretation for this equality.

Mathematics is full of automatic conversions, i.e. operations that changes the type of an object *without changing its notation*. This can give the illusion that mathematics are not as strongly typed as I suggest, or even not typed at all. However, this is just an illusion created by the syntactic facility provided by the possibility to define automatic conversions.

1.4 Polysemy and the resolution of ambiguities

As remarked above, most mathematical symbols are *polysemic*. In other words, they have several distinct meanings. This is particularly true for the symbol $\times$ (multiplication) that is used for multiplying numbers, matrices, polynomials, and dozens of other types of objects, including sets. Notice that we sometimes introduce distinct notations for operations that we consider as essentially of the same nature, but this is always in a case where a notion of type cannot help. For example, an appropriate set of functions from $\mathbb{R}$ to $\mathbb{R}$ has a product, but also a convolution product. Since no notion of type can discriminate between them (because these operations apply to the same type of objects), they are represented by distinct symbols. There are of course other examples of this fact.

Polysemy is mandatory in practice. It is clear for example that mathematics would become untractable if it was necessary to give distinct names to all sorts of multiplications. However, the problem with polysemy is that it creates ambiguities, and ambiguities must necessarily be resolved. Assume that u and v are either two sequences or two series of real numbers. How do you interpret the product uv ? You cannot do it unless you know if u and v are sequences or series. Hence, the only thing that can let you understand what you are doing, in other words, let you *understand mathematics*, is the presence of types. Modern strongly typed programming languages can allow polysemy, precisely because they are typed.

2 Proof interchangeability

2.1 What is it ?

Consider a mathematical text, for example an article, and assume that it is correct (i.e. that it doesn't contain any error). This text contains *terms* (i.e expressions representing mathematical objects), statements and proofs. Proof themselves can contain other proofs, so that by "proof" I mean not only a proof of a theorem, but any proof ("subproof") which is part of the proof of a theorem. In all cases, a proof is always proving a statement (the *goal* of the proof).

It is clear that if you replace a term by another term representing the same mathematical object (in the same context([7])) within this text, the text remains correct. For example, if you replace 2 by $1 + 1$, nothing bad can happen. On the contrary of course, if you replace 2 by 3, it is likely that the text becomes erroneous, despite the fact that 2 and 3 are of the same type (for example, if they are both natural integers).([8])

Now, if you replace a proof by another proof (correct in the same context) of the *same* statement within the text, the whole text is still correct, even if the replacement proof looks very different from the original one. Hence, all proofs of a given statement have this very particular property that they are *interchangeable*. In some sense, they are all *equal*.

This principle (or phenomenon) is known as *proof irrelevance*. I prefer to call it *proof interchangeability*, because this tells more explicitely what the phenomenon is about, and because this can avoid the well known polemics about the notion of *equality between proofs*. Many logicians and theoretical computer scientists have a point of view which seems to contradict the fact that *any two proofs of the same statement are equal*, since they can spend considerable energy in the analysis of the structure of proofs, which leads them to define various semantics for proofs. For example, proofs can be interpreted as algorithms, as nets, as tangles, as higher arrows in ∞-categories, etc… and of course, in such studies, it is necessary and natural not to consider any two proofs of the same statement as *equal*.([9])

Hence, I retain only the fact that any two proofs of the same statement are *interchangeable*, which just means that any of them can do the job of *proving the goal*. They do that more or less elegantly, but they do it. As we shall see, as far as the structure of the vernacular language of mathematics is concerned, proof interchangeability is very important. It is even a *mandatory* principle for understanding how the vernacular works.

In order to support this thesis, I shall discuss the matter from four different points of view. The first one is an examination of how mathematicians behave, since centuries, regarding mathematical proofs (and mathematical statements). This behaviour is *natural* in the sense that it is part of the long lasting darwinian evolution which led to contemporary mathematics. Furthermore, all the aspects of this behaviour tend to prove that indeed, for a mathematician, proofs of a given statement are interchangeable. The second point of view, is coming from an analysis of the interpretation of mathematics within a topos. It is known that toposes are *mathematical universes* where mathematics can be carried out.

[7] More on this notion of context below.

[8] Notice that decimal notations, just like most symbols, are polysemic. For example, 2 can be a natural integer, a relative integer, a rational, etc… even a polynomial or a matrix, and many other things.

[9] This is anyway what anyone who wants to create a proof assistant needs to do, including myself. However, proofs can at the same time be *equal* in some sense, and be different in another sense, in the same way as 2 and $1 + 1$ are equal and also different from another point of view.

This is not very surprising since a topos is essentially a variant of the *category of sets*. We shall see that this analysis yields the principle of *the uniqueness of the warrantor*, which is even stronger than proof interchangeability. The arguments of the first point of view above can be considered as *symptoms* of the presence of proof interchangeability, and the second point of view relies on the hypothesis that topos theory is the right choice for formalizing mathematics. By contrast, I also give an attempt at *explaining* proof interchangeability by a fundational argument. I'm conscious of the fact that this argument is maybe not perfect, but it is the best I have at hand from the point of view of foundations. The last point of view is that of the programer who is designing a proof assistant, in other words, a computer software for *doing mathematics*, more precisely, a software which is able to check that mathematics written in a language which is supposed to look as much as possible like the mathematical vernacular, is formally correct. We prove in [12] that not assuming proof interchangeability leads to *exotic mathematics* which is unacceptable by mathematicians.

2.2 A natural behaviour

There are many peculiarities in how we write mathematics which can easily be explained by the fact that any two proofs of the same statement have the same logical value (are interchangeable) for the mathematician. We record some of them below.

Reading the proof of a theorem is not necessary for using it. This is a well known fact in mathematics. If you need to apply a theorem in order to prove something, you don't need to have a look at a proof of this theorem. Only the statement of the theorem is needed. However, the fact that the theorem was (reliably) proved at least once is also necessary, but how it was proved is pointless. In other words, all proofs of the theorem have the same logical value. Of course, you can, for various reasons, be interested in a new elegant proof of a well known theorem, but it is pointless for using it. You can also be interested in reading the proof of the theorem simply in order to die a little less ignorant. In practice, it is often the case that when we write an article within which we need to use a theorem, we just don't have time to have a look at a proof of it. Many mathematicians have published articles using theorems the proofs of which they have never seen. Of course, they will not shout it from the rooftops.

At this point I want to make a comparison with programming. When programmers need to use a function from a library of programs, they do not only need to read the declaration of this function (which mainly consists in the indication of the types of its arguments and the type of its result). They also need to read a piece of documentation explaining what this function actually does (RTFM !([10])). Mathematicians are very happy compared to programmers.

A theorem is not renamed when a new proof of it is found. If you produce a new proof of an already known theorem, the theorem will not take your name. It keeps the name of the first person who was able to prove it. This is true even if your proof is much better than the original proof. In this case, the new proof can receive your name, not the theorem itself. For example, Cayley-Hamilton's theorem has got dozens of quite different proofs, but never changed its name. Actually, the name of a theorem is neither given to the statement of the theorem nor to a proof of it, but to *the fact that it is proved*([11]), which does not change when

[10] Programmers will understand this.

[11] In the sequel, such a fact is called a *warrantor*.

a new proof is found.

Assumptions are most often anonymous. When we make an assumption, for example when we write “Assume that E”, where E is a statement, we don’t in most cases give a name to this new hypothesis. Nevertheless, we can give it a name, and most often this name is just a number written between a pair of parentheses after the assumption itself. For example, we can write “Assume that E (2.4)”, so that later in the text we can write “by (2.4), we have . . . ”. This name (2.4) is by no way required. This is just an help for the reader, a reminder that indeed the statement E can be taken as an hypothesis. The reason why this name is not mandatory is that the fact that E is true is a unique fact. There are not two distinct facts that E is true.

Proofs may have gaps. Anyone who has done a bit of mathematics has remarked that unlike terms and statements which are written in a formal language (using x’s and y’s), proofs are written in ordinary parlance, and that depending on the targeted audience, more or less details are given. In other words, proofs have *gaps*, and it is up to the reader to fill up these gaps. Gaps in proofs are easily visible. For example, if you read “It is clear that E” (where E is a statement), you know that it is up to you to find a proof of E. In general, this is not difficult (since it is supposed to be *clear*), and you can do it out of one’s head. Of course, the text of the proof has no control on how you prove E in your head. The reason is that it doesn’t matter, since only the fact that E is true is required. There are many other currently used formulations in mathematical proofs that hide such a gap.

Boolean and Heyting algebras. In the literature, a very large part of logic is carried out within the framework of boolean algebras and Heyting algebras. These algebras are nothing other than (small) categories, which are actually ordered sets. What characterizes (pre)ordered sets among categories, is that any two objects X and Y being given, there is *at most* one arrow from X to Y. In the logical interpretation, the order relation is actually the *deducibility relation*. In other words, arrows from X to Y in such categories represent proofs of Y under the hypothesis X, or at least represents what is also represented by proofs of Y under hypothesis X. The fact that so many people have used such categories in which there is at most one “proof” of a statement, is a symptom that proof interchangeability has been widely admitted, even if it was unconsciously.

The very late apparition of proof theory. Proof theory began only arround 1930, with the works of Heyting, Gentzen and some others. Why did mathematicians of the XIX$^{\text{th}}$ century, and the others before them, not think about the structure of proofs ? Probably just because it doesn’t matter so much. Of course, this structure matters very much for me, depite the fact that proofs are interchangeable, because within the source code of a proof assistant, proofs are treated as precise objects, and in particular there are *distinct* proofs for the same statement. For example, such a program contains a routine for normalizing proofs, which clearly implies that all proofs of a given statement are not confused. But here, by *distinct* we clearly mean *syntactically distinct*, not *semantically distinct*.

2.3 What topos theory tells us

Topos theory was initiated by Alexandre Grothendieck in 1963, and was not designed as a tool for logic, but as a tool for algebraic geometry. Nevertheless, William Lawvere realized in 1964 that a topos shares many features with the category of sets (which is just a topos among the others). Around 1970, Lawvere and Tierney gave a simplified and more general definition of a topos, that they called an *elementary topos*. This structure is well suited for studying mathematical logic, even if Grothendieck toposes are also very often used in logical matters.

Roughly speaking, an elementary topos is a category which has a product with a neutral element, pullbacks and power objects. In the case of the category of sets, these notions correspond to those of cartesian product, singleton, subset and power set. So, it is not surprising that mathematics can be carried out within this structure, in a similar way as mathematics can be carried out in a model of set theory. Nevertheless there are some differencies between the two approaches. The most visible one is that mathematics is by default intuitionnistic in a topos and classical only in particular toposes, whereas it is always classical in any model of set theory. Another difference is that topos theory can take into account a notion of type, whereas set theory cannot. My opinion is that topos theory is very much better suited for formalizing mathematics than set theory. Another way of saying that, recalling that there are at least two kinds of set theories, namely *naive set theories* and *erudite set theories* (such as Zermelo-Fraenkel set theory and Gödel-Bernays set theory), is that topos theory can actually be seen as a formalization of naive set theory, the theory which is *de facto* used in everyday mathematics.

This article is not the place for explaining topos theory in details. I just recall some facts which are essential for defining what I call a *warrantor*.([12])

In a topos (say $\mathscr{T}$), there is a particular object generally denoted Ω and called the *subobject classifier*. This object is to be understood intuitively as *the set of truth values*. In the case of the topos of sets, Ω identifies to the set $\{\top, \bot\}$ of booleans, but in an arbitrary topos, Ω, which is not necessarily a set, since objects in a category are not necessarily sets, can have a more complex structure. In particular, it can have truth values (arrows from 1 to Ω) distinct from $\top$ and $\bot$. Furthermore, expressions of the language of mathematics have to be interpreted within a *context*. Indeed for example, an expression such as $x > 0$ is meaningless if x is neither declared nor defined.

A context is essentially a (finite and ordered) sequence of *declarations* of the form $x \in X$, where x is a variable and X a set, and *assumptions* that we denote by $\zeta \vdash E$ (read "ζ proves E"), meaning that the statement E is a new hypothesis whose (non mandatory) name is ζ. It is natural to interpret a context as an object within a topos $\mathscr{T}$. This is done by induction as follows. First of all the empty context (no declaration and no assumption) is interpreted as the terminal object $\mathbf{1}$. At any stage of the induction, and if $\overline{\Gamma}$ denotes the object of $\mathscr{T}$ representing the context Γ, sets which are meaningful in the context Γ are interpreted as objects of the slice topos $\mathscr{T}/\overline{\Gamma}$, or if you prefer as arrows of $\mathscr{T}$ targeting $\overline{\Gamma}$. The context obtained by adding the declaration $x \in X$ to Γ is interpreted as the source object (in $\mathscr{T}$) of the arrow which is the object of $\mathscr{T}/\overline{\Gamma}$ representing the set X. Statements which are meaningful in the context Γ are represented as arrow $\overline{\Gamma} \to \Omega$ in $\mathscr{T}$, and the context obtained by assuming the statement E in the context Γ is represented by the source object of the pullback of the arrow $\top : \mathbf{1} \to \Omega$ along the interpretation of E.

Despite its appearent complexity, this definition is easy to understand intuitively because in the case of the topos of sets, it amounts to the fact that $\overline{\Gamma}$ is the set of all *instance* of Γ, in other words, of all tuples of (legal) values that we can assign to the variables declared in Γ, and for which all assumptions in Γ are true.

Summarizing, any statement E which is meaningful in the context Γ, is interpreted as an

[12] "garant" in French. This notion was first introduced in [11].

arrow $\lfloor E \rfloor_\Gamma$ from $\overline{\Gamma}$ to Ω :

$$\begin{tikzcd}\overline{\Gamma} \arrow[r, "\lfloor E \rfloor_\Gamma"] & \Omega\end{tikzcd}$$

Intuitively, the "function" $\lfloor E \rfloor_\Gamma$ maps any instance of the context to the truth value taken by E when the free variables in E are replaced by the values given by the instance. Similarly, sets which are meaningful in the context Γ are interpreted as arrows targeting $\overline{\Gamma}$, in other words, as objects in the slice topos $\mathscr{T}/\overline{\Gamma}$ (which means that sets are dependent on the context).

Now, in any topos $\mathscr{T}$, there is an arrow $\top : \mathbf{1} \to \Omega$ from the terminal object (intuitively, the canonical singleton) to Ω, which represents the truth value $\top$ (i.e. true). It is natural to consider the pullback of $\top$ along $\lfloor E \rfloor_\Gamma$:

$$\begin{tikzcd}\bullet \arrow[r] \arrow[d, "j"'] & \mathbf{1} \arrow[d, "\top"] \\ \overline{\Gamma} \arrow[r, "\lfloor E \rfloor_\Gamma"] & \Omega\end{tikzcd}$$

which gives an arrow j targeting $\overline{\Gamma}$, which is hence candidate to be the interpertation of a set in the context Γ. The question now is *which set has this arrow j as its interpretation* ? In order to answer this question, we can first try to understand the global elements of j seen as an object of $\mathscr{T}/\overline{\Gamma}$, i.e. the *elements* of this mysterious set. Since the terminal object in $\mathscr{T}/\overline{\Gamma}$ is the identity arrow of $\overline{\Gamma}$, such global elements are nothing other than the *sections* of j. It is an elementary exercice of category theory to prove that such a section exists if and only if the arrow $\lfloor E \rfloor_\Gamma$ can be lifted along $\top : \mathbf{1} \to \Omega$. But saying that $\lfloor E \rfloor_\Gamma$ lifts along $\top$ is exactly the definition that E is true in the context Γ in this topos interpretation. As a consequence, *a section of j exists if and only if E is true* ! Hence, it was natural to call such a section a *warrantor* of E, because this arrow (which is a mathematical object) warrants the truth of E. Consequently, the arrow j itself cannot be anything other than the interpertation of *the set of warrantors of* E (in the context Γ), that we denote $W(E)$ (and which is thus context dependent).

Now, the interesting fact as far as proof interchangeability is concerned is the fact that a statement E (in a given context) cannot have more than one warrantor, since j, which is a monomorphism, cannot have more than one section. I call this the *principle of the uniqueness of the warrantor*. Since a proof of E cannot be anything other than a notation for a warrantor of E (this can be taken as a definition of the notion of *proof* if $\mathscr{T}$ is the free topos), it is immediate that the principle of the uniqueness of the warrantor entails proof interchangeability.

Hence, anybody who trusts topos theory as a foundation for mathematics should consider proof interchangeability as a mandatory principle.

2.4 A fundational argument

Mathematics is a four levels system, where each level has a syntactical and a semantical aspect, and depends on previous levels. The table below gives a picture of this situation, where the most fundamental level is at the bottom of the table.

signifiers	*signified*
proofs	warrantors
statements	truth values
terms	mathematical objects
types	types

The most fundamental level is that of types. Everything is constructed above it, and this level has the very important property from the fundational point of view that equality between types is nothing other than syntactical identity. Notice that this is made possible by the fact that the type system we have in mind has no so-called *dependent types*. The type system designed by Lambek has this property and is very well suited for formalizing mathematics, except that it lacks a notion of clone. Details on Lambek's system can be found in [7].

That equality between types is algorithmically decidable is important from a fundational point of view, because if a notion of proof is needed for testing equality between types, a still more fundamental level would be required, and we avoid an infinite regression only if at least one of the levels has an algorithmically decidable equality. In our view, this is the level of types. Notice that if we had choosen a type theory with dependent types, we could still have a well founded system because it would still be possible to consider that two types are equal if and only if they have the same normal forms, a concept which is still algorithmically decidable.([13])

The second level is that of terms and mathematical objects. It is dependent on the level of types because each mathematical object has one and only one type, which implies that each term has a unique type and that any two terms which are *equal* have the same type. It is an important fact that the type of a term is algorithmically decidable, possibly using particular techniques due to the presence of polysemy.([14])

The third level is that of statements and truth values. It is dependent on the previous level in the sense that statements are just *stating* properties of mathematical objects. In particular, statements are expressions which can contain terms as subexpressions.

The last level is that of proofs and warrantors. It depends on the previous level in the sense that each proof proves a statement (that we call the *goal* of this proof). Warrantors are the semantics of proofs, and as explained above, the principle of the uniqueness of the warrantor (for a given statement) entails proof interchangeability.

It is a fact that the last two levels (statements,proofs) have some common behavior with the first two levels (types,terms). This is known as the *Curry-Howard correspondance*. However, this correspondance is far from being an *isomorphism* as it was originally presented, in the presence of proof interchangeability, because in this case we have instead a kind of duality in the fact that types have *at least* one element, whereas statements have *at most* one warrantor. This is actually neither a *correspondance*, but it is still true that the two pairs of levels share a common structure.([15])

Now, we arrive at our fundational argument, which is quite simple. If it is the case that two warrantors of the same statement can be distinct, it must be the case that this distinction

[13] This is the position adopted for the type system W sketched in [11].

[14] This is an allusion to the Hindley-Milner algorithm also known as *type-checking algorithm*. The heart of this algorithm is the notion of *unification*.

[15] Essentially, that of a bicartesian closed category.

is used for something, whatever it is. Otherwise, there is no practical reason for considering it. But since proofs can contain statements and terms, and since equality between statements and terms is not algorithmically decidable, it is likely that equality between proofs would also not be algorithmically decidable. This implies that we need a notion of *meta-proof* in order to ensure that two proofs are equal. It is clear that we are beginning another infinite regression, which is unacceptable from a fundational point of view. To my opinion, this is the true reason why any two proofs of a given statement are *equal*.

2.5 A design decision for any proof assistant

The notion of *set* is central in contemporary mathematics but should not necessarily be understood as the notion defined by Zermelo-Fraenkel set theory. Actually, there is a naive notion of set, first introduced by Bolzano at the end of the XIX$^{\text{th}}$ century and made universally known by Cantor (and the word *set* was for sure already used before Bolzano), and also a notion of *type* in everyday mathematics. A proof assistant can either have both notions (types and sets) or only one of them, most often that of type, which seems to be a consequence of the fact that in such a system, types often play the role played by sets in mathematics.

A very important concept in mathematics is that of *subset*. In a system where types play the role of sets, we will have a notion of *subtype* (should it be primitive or defined). The most usual way of producing an element in a subset A of a set X is to start with an element a of X, and prove that $a \in A$. In a proof assistant, nothing should be lost, and in particular proofs must be kept. This is why it is tempting to define an element of a subset (or subtype) A of a set (or type) X, as a pair (a, p) made of an element a of X and a proof p that $a \in A$. This is actually for example what Martin-Löf does in his type theory [8]. However, in everyday mathematics, there is a *canonical inclusion* $i : A \to X$, and this map is supposed to be *injective*. What happens if we have two non equal proofs p and q of the fact that a belongs to A ? We get two elements (a, p) and (a, q) of A which seem to be *a priori* non equal if p is not equal to q. As a consequence, the canonical inclusion $i : A \to X$ is not injective since (a, p) and (a, q) are both mapped to a.

It is just a matter of fact that non injective canonical inclusions are unacceptable by mathematicians. This is one of the reasons why Martin-Löf-like systems implement what we could call *exotic mathematics*. This was the case of the proof assistant Coq, until it offered the possibility to state proof irrelevance as an axiom. If we accept that the above two proofs p and q are equal, then the *projection* $(a, p) \mapsto a$, which represents the canonical inclusion, becomes injective. Martin-Löf has changed his point of view in 2006 [9].

This problem has nothing to do with the design decision of using types instead of sets, or only sets or both notions. It is only a question of equality between proofs. The above discussion could let you think that the problem can still be solved in one way or another in a system willing to consider that a statement can have non equal proofs. This is not the case as we shall see below. Indeed, it is known since Martin-Löf's work on type theory that the axiom of choice can be proved in a constructive system allowing non equality between proofs. On the other hand, there is a theorem (Diaconescu's theorem) which asserts that the principle of the excluded middle is a constructive consequence of the axiom of choice. If this theorem was provable in Martin-Löf's type theory (which is constructive), the principle of the excluded middle would also be constructively provable. But it is an easy consequence of Gödel's incompleteness theorem that this cannot be the case (see [12]). As a consequence, there is at least one step in the proof of Diaconescu's theorem which cannot be carried out in Martin-Löf's system. If we try to formalize the proof in this system, we find that at some point, a canonical inclusion cannot be proved injective. This was proved in [10].

As a consequence, there is no hope that a proof assistant allowing non equal proofs of the same statement can ever be accepted as a working tool by mathematicians. It is to my opinion necessary, not only to allow proof interchangeability, but even to put it as an unquestionable (hard coded) principle.

A consequence of taking the design decision that any two proofs of the same statement are equal is that there is no more need to represent elements of a subset A of a set X as pairs (a, p) as we saw above. Proofs of statement can now be kept outside the representation of mathematical objects, into a so-called *knowledge base*. When something has to be proved, the system should just use this knowledge base to get (or reconstruct) a proof. That way, descriptions (implementations) of mathematical objects are separated from the properties they can have, which are all recorded into the knowledge base. This will not only simplify the design of the proof assistant, but also ensure that the mathematics formalized by the system is actually concordant with *the mathematics of everybody*, which is of course much more important.

References

[1] **G. Cantor** : *Beiträge zur Begründung der transfiniten Mengenlehre.* Mathematische Annalen 46, p. 481-512.

[2] **A. Fraenkel** : *Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre.* Mathematische Annalen 86, p. 230-237.

[3] **G. Frege** : *Sinn und Bedeutung.* Zeitschrift für Philosophie und philosophische Kritik (100), 1892.

[4] **M. Herrmann, A. Prouté** : *On dependent conjunction and implication.* (2016) arXiv:1606.06005 [math.LO]

[5] **J.-L. Krivine** : *Théorie des Ensembles.* Cassini, Paris, 2007.

[6] **J. Lambek** : *From types to sets.* Advances in Mathematics 36, 1980, p. 113-164.

[7] **J. Lambek, P. J. Scott** : *Introduction to higher order categorical logic.* Cambridge University Press, Cambridge 1986.

[8] **P. Martin-Löf** : *Intuitionnistic type theory.* Studies in Proof Theory, Bibliopolis, Naples, 1984.

[9] **P. Martin-Löf** : *100 Years of Zermelo's Axiom of Choice : What was the Problem with It ?* The Computer Journal Volume 49 Issue 3, May 2006. Pages 345-350

[10] **A. Prouté** : *Expressions indéterminées, constructivisme et axiome du choix.* Cahiers de Topologie et Géométrie Différentielle Catégoriques (1992) Volume: 33, Issue: 3, page 279-288.

[11] **A. Prouté** : *Sur quelques liens entre théorie des topos et théorie de la démonstration.* `http://www.logique.jussieu.fr/~alp/luminy_05_2007.pdf`

[12] **A. Prouté** : *Demystifying the axiom of choice, or what do we mean by "chosing" ?* `http://www.logique.jussieu.fr/~alp/demystifying_choice.pdf` (currently in preparation)

[13] **E. Zermelo** : *Untersuchungen über die Grundlagen der Mengenlehre. I* Math. Annalen, vol. 65, 1908 p. 261-281.