THE FORMALIZATION OF MATHEMATICS

by
Harvey M. Friedman
Ohio State University
Department of Mathematics
friedman@math.ohio-state.edu
www.math.ohio-state.edu/~friedman/
May 21, 1997

Can mathematics be formalized?

It has been accepted since the early part of the Century that there is no problem formalizing mathematics in standard formal systems of axiomatic set theory. Most people feel that they know as much as they ever want to know about how one can reduce natural numbers, integers, rationals, reals, and complex numbers to sets, and prove all of their basic properties. Furthermore, that this can continue through more and more complicated material, and that there is never a real problem.

They are basically correct. However, the formalization of mathematics is extraordinary inconvenient in any of the current formalisms. But why do we care about inconvenience? Put differently, why would anyone want to formalize mathematics, since everybody thinks anybody who cares can? Let me distinguish two concepts of formalization. The first is what I call syntax and semantics of mathematical text. Here there are no proofs. One is only concerned with a completely precise presentation of mathematical information.

This is already grossly inconvenient in present formalisms. Why do we want to make this convenient?

GENERAL GOALS

1. To obtain detailed information about the logical structure of mathematical concepts. For instance, what are the appropriate measures of the depth or complexity of mathematical concepts? What are the most common forms of assertions? We hope for interesting and surprising information here.

2. To develop a theory of mathematical notation, and notation in general. When how and why do mathematicians break concepts up into simpler ones? What is it about mathematical notation that makes it convenient and readable? These are important matters that have evolved in a certain way; e.g., consider music notation.

3. To maintain a uniformly constructed database of mathematical information. Such a database would benefit from agreement on notation, and would also help facilitate it.

There could be automatic algorithms for changing notation. Also information retrieval of various kinds seem useful and interesting.

The more ambitious concept of formalization includes proofs. These are even much more inconvenient in present formalisms. What is to be gained by making them reasonably convenient?

4. To obtain detailed information about the logical structure of mathematical proofs. For instance, there is a sophisticated area of logic called proof theory, where there is almost no such detailed information.

There is a lot of information in logic about unprovability, but virtually nothing about real proofs. What inference rules are really used frequently? Is there a good classification of the levels of triviality?

5. To maintain a uniformly constructed database of verified mathematical information. Of course, the success of this project depends delicately on how convenient people think it is.

You might be able to consult such a database with intelligent tools and retrieve information about what is known. Uniform presentation of mathematical information is necessary to really get this going.

6. To develop more convenient ways to verify software and hardware. There are other issues that need to be addressed in order to accomplish this such as overhauling the present programming languages.

REVIEW OF STANDARD FORMAL SET THEORY

The language has the following:

i) connectives $\neg\rightarrow\wedge\vee\leftrightarrow$;
ii) variables $x_1,x_2,\ldots$ ranging over sets only;
iii) quantifiers $\forall,\exists$;
iv) membership $\in$;
v) equality $=$.

The terms consist of just the variables. The atomic formulas are equality and membership between terms. Formulas are obtained from the atomic formulas by combining according to the connectives; and by quantification. Thus if A,B are formulas, then so are $\neg A$, $A\rightarrow B$, $A\wedge B$, $A\vee B$, $A\leftrightarrow B$, $(\forall x_n)(A)$ and $(\exists x_n)(A)$.

There are the nine usual axioms and axiom schemes (ZFC):

1. Extensionality. Two sets are equal if and only if they have the same members.
2. Pairing. There is a set consisting of exactly any two (possibly equal) sets.
3. Separation. (Infinitely many axioms). For any formula A in our language, $\{x \in y: A\}$ exists.
4. Union. For any set x, there is a set consisting of exactly the elements of the elements of x.
5. Power set. For any set x, there is a set consisting of exactly the subsets of x.
6. Infinity. There is a set x containing the empty set, and where for all $b \in x$, $b \cup \{b\} \in x$.
7. Replacement. (Infinitely many axioms). For any formula A in our language, if $(\forall x \in u)(\exists! y)(A(x,y))$ then $(\exists z)(\forall x \in u)(\exists! y \in z)(A(x,y))$.
8. Foundation. In every nonempty set x there exists $y \in x$ such that for all $z \in x$, $z \notin y$.
9. Choice. Let x be a set of pairwise disjoint nonempty sets. Then some set has exactly one element in common with each element of x.

One also has some version of predicate calculus at the bottom.

MORE CONVENIENT FORMALISM: GENERAL DISCUSSION OF SYNTAX

Work in joint progress with Randy Dougherty.
Again we stick to mathematical text without proofs.

I think there is a concept of workability for mathematics that is much better than usual formalisms, but which does not support a lot of things mathematicians do in print. The real test is whether the constraints lead to text that is about as good as what is normally written. The acid test is to see what actual formal text looks like.

For example, mathematicians are used to writing

the function sin(x).

They know they really mean

the function of $x \in$ R, sin(x).

They know this is a problem when they write

the function sin(x+y).

They know they really mean

the function of $x \in$ R, sin(x+y).

Now cleaning this all up by l-notation is actually an improvement; it is certainly workable:

λx $\in$ R sin(x+y).

However, say we ban the use of infix operator symbols or ban the use of infix operator variables, or require parentheses be put in. Then we have to write them in prefix, or put a lot of parentheses in. This is not within the realm of being OK, since it seriously affects readability and writability.

Later, with more experience, one can of course add more features.

We will present the theory of formulas and terms in what we call a relational type.

We use the following 22 special characters:

$\in = \neg \wedge \vee \Rightarrow \Leftrightarrow \forall \exists \lambda , ( ) \{ \} [ ] \uparrow \downarrow ! | \sim$

A relational type α consists of the following data:

i) a finite set T which is disjoint from the set of special characters;
ii) an assignment of a unique role to each element of T;
iii) a finite set B of ordered pairs of distinct objects, where the components of B are disjoint from each other and from T and from the set of special characters;
iv) an infix precedence table;
v) we require that there exists an element whose unique role is that of simple variable (see below).

Here are the roles for elements of T:

i) as a constant symbol;
ii) as a sentence symbol;
iii) as a simple variable;
iv) as a normal prefix variable;
v) as a unary prefix variable;
vi) as an infix variable;
vii) as a normal prefix function symbol;
viii) as a unary prefix function symbol;
ix) as an infix function symbol;
x) as a prefix relation symbol;
xi) as an infix relation symbol;
xii) as a prefix relation variable;

xiii) as an infix relation variable.

All elements of B are understood to be in the role of "bracketing function symbol."

The infix precedence table of a consists of the following data:

a) a quasi linear ordering of a subset of the infix function variables and infix function symbols;
b) to each equivalence class there is assigned "left associative," or "right associative".

Each of the 14 roles (including that of "bracketing function symbol") define a set. E.g., the set of all infix function symbols is the set of all elements of T whose role is that of infix function symbol. We call also create sets based on the words "symbol" and "variable"; i.e., the variables consist of the simple variables, the normal prefix variables, the unary prefix variables, the infix variables, and the relation variables.

The associative variables (symbols) are the infix function variables and infix function symbols that are in the precedence table. The nonassociative variables (symbols) are the variables (symbols) that are not in the precedence table.

Let A be the alphabet consisting of the special characters, T, and the components of elements of B. Let A* be the set of all finite strings from A (including the empty string).

Later we will define five subsets of A* by simultaneous recursion. These are the terms, formulas, p-terms, p-formulas, and restricted variables. This definition is really a context free grammar. A nasty unique parsing theorem must be proved. This has been finished.

GENERAL DISCUSSION OF VARIABLES AND SYMBOLS

One of the key semantic concepts is that of an interpretation of a relational type.

This semantics is conducted within the usual theory of classes. We keep this necessary metatheory of classes at an informal level in this section.

Recall the 14 roles of elements of T. Each element of T has exactly one role.

An interpretation assigns objects to the various elements of T. As we shall see, sometimes it is optional that there be an assigned object.

We assign sets to zero or more of the constant symbols. Thus some constant symbols may not be assigned anything.

We assign a truth value (true or false) to each sentence symbol.

We follow Tarski in that assignments to variables are not part of the interpretation, but rather are essential in defining the inductive concept of truth. So we will additionally introduce the concept of variable assignment.

To each normal prefix function symbol there is assigned a partially defined map from the class of all finite sequences of sets into sets. This may be a proper class.

To each unary prefix function symbol there is assigned a partially defined map from the sets into the sets. This may be a proper class.

To each infix function symbol there is assigned a partially defined map from 2-tuples of sets into sets. This may be a proper class.

To each prefix relation symbol there is assigned a class of finite sequences of sets. This may be a proper class.

To each infix relation symbol there is assigned a class of 2-tuples of sets. This may be a proper class.

Now we come to variable assignments. A variable assignment must assign a set to each variable.

To each simple variable there is assigned a set.

To each normal prefix variable there is assigned a partial function from finite sequences of sets into sets. This is required to be a set.

To each unary prefix variable there is assigned a partial function from sets into sets. This is required to be a set.

To each infix variable there is assigned a partial function from 2-tuples of sets into sets. This is required to be a set.

To each prefix relation variable there is assigned a set of finite sequences of sets.

To each infix relation variable there is assigned a set of 2-tuples of sets.

In the full blown formal treatment, one defines the value (if any) of terms and truth values of formulas, under an interpretation and a variable assignment. In mathematics

truth values of formulas always exist, yet values of terms may not.

GLOBAL STRUCTURE OF TEXT

The main finished product that the theory covers is that of a valid text. Here a text does not contain any proofs. Rather it contains assertions only, and these assertions are or are not valid.

We don't quite say that these assertions are or are not true. This is because in mathematics, we introduce, say, a prefix relation symbol, and define its meaning only a certain objects that we care about. It isn't that the prefix relation symbol is officially "undefined" at other arguments. Rather it's truth value is "undetermined," which is quite different.

So in text, not only are there claims, but also there are various kinds of incomplete definitions and conventions of various sorts.

Definitions and conventions must take on a certain form. The definitions are usually incomplete, and they are augmented or verified as the text proceeds without necessarily changing the symbols. This still creates no ambiguity if looked at in the right way.

The conventions are of two main kinds. The first kind states the range of objects associated with variables. E.g., one might introduce G as a variable ranging over groups.

The second kind are what we call control conventions. For example, these assert that certain definitions or conventions apply only to certain regions of the text. E.g., a change of notation, or a definition that is in force only for "section 6."

Several big inductive definitions are to be made. First of all, the conventions determine the "current relational type" at any "point" in the text.

Secondly, one must define the concept "total definitional commitment" at a point in the text. This takes into account all of the definitions and conventions in force at that point in the text, and how they interact to form a total effect.

Thirdly, one defines the concept of a claim being "valid relative to the total definitional commitment" at a point in the text. This uses the interpretations treated in the previous section.

A text is then said to be valid if and only if every claim is valid relative to the total definitional commitment at the point of that claim in the text.

One of the fundamental facts that must be proved is the following. Here is a very special case: In a valid text, every formula in pure set theory that appears as a claim must be true in the standard sense of set theory. That is, all of the complicated interlocking conventions and definitions that can be made cannot allow you to write down a false assertion in pure set theory.

INFORMAL DISCUSSION OF THE SEMANTICS OF $\in = \neg\wedge\vee\Rightarrow\Leftrightarrow\forall\exists\lambda,(\,)\{\}[\,]\uparrow\downarrow!|\sim$

All terms denote sets or nothing (undefined). $s \in t$ is always true or false. If s or t is undefined, then $s \in t$ is false. The same holds of $s = t$.

Now $s \sim t$ means that either s,t are both defined and identical, or are both undefined.

$s\downarrow$ means that s is defined.
$s\uparrow$ means that s is undefined.

The logical connectives $\neg\wedge\vee\Rightarrow\Leftrightarrow$ are self explanatory. Every formula has a truth value, regardless of whether or not a lot of its subterms are undefined; there are no undefined formulas.

$s(t_1,\ldots,t_n)$ means the result of applying s to the arguments $t_1,\ldots,t_n$. In this context, any term s can be treated as a (partial) function of any number of arguments. If s or any t_i is undefined then $s(t_1,\ldots,t_n)$ is automatically undefined.

$s[t_1,\ldots,t_n]$ means that the value of s holds at the values of $t_1,\ldots,t_n$. Thus in this context, any term s can be viewed as denoting a relation of any number of arguments. $s[t_1,\ldots,t_n]$ is always defined, and has a truth value. If any of s, $t_1,\ldots,t_n$ are undefined, then $s[t_1,\ldots,t_n]$ is automatically considered to be false.

$(!x)(\varphi)$ means the unique object x such that φ holds. If there is no such unique object (either more than one exists, or none exists), then this is undefined.

We use $\{t_1,\ldots,t_n\}$ for the set consisting of $t_1,\ldots,t_n$. If any t_i is undefined, then the whole term is undefined.

Braces are also used in $\{x|\varphi\}$. Here this denotes the set consisting of all sets x such that φ. There may not be such a set, in which case $\{x|\varphi\}$ is undefined.

We finally come to λ. $(\lambda x)(s)$ denotes the set function f such that for all sets x, either f(x) is the value of s, or both f(x) and s are undefined. There may not be such a set function. If there is, it obviously is unique. If there is no such set function, then $(\lambda x)(s)$ is undefined.

We also use $(\lambda x|\varphi)(s)$, which denotes the set function f whose domain consists of all x such that φ holds and s is defined.

FORMAL TREATMENT OF THE SYNTAX OF FORMULAS AND TERMS IN A RELATIONAL TYPE

We now define five subsets of A* by simultaneous recursion. These are the terms, formulas, p-terms, p-formulas, and extended variables. This definition is really a context free grammar.

1. Every constant symbol and every variable is a p-term.
2. If s is a term, then (s) is a p-term.
3. If x is a unary function variable or unary function symbol and s is a p-term, then xs is a p-term.
4. If x is a normal prefix function variable or normal prefix function symbol, and $s_1,...,s_n$ are terms, $n \geq 0$, then $x(s_1,...,s_n)$ is a p-term.
5. If $s,t_1...,t_n$ are terms, $n \geq 0$, then $(s)(t_1...,t_n)$ is a p-term.
6. If x y is a bracketing function symbol and $s_1,...,s_n$ are terms, $n \geq 0$, then $xs_1,...,s_ny$ is a p-term.
7. If $s_1,...,s_n$ are terms, $n \geq 0$, then $\{s_1,...,s_n\}$ is a p-term.
8. If x is a variable or restricted variable, and φ is a formula, then $\{x|\varphi\}$ is a p-term.
9. If $x_1,...,x_k$ are variables, $k \geq 1$, φ is a formula, and s is a p-term, then $\lambda x_1,...,x_k$ s and $(\lambda x_1,...,x_k)s$ and $(\lambda x_1,...,x_k|\varphi)s$ are p-terms.
10. If $x_1,...,x_k$ are restricted variables, $k \geq 1$, φ is a formula, and s is a p-term, then $(\lambda x_1,...,x_k)s$ and $(\lambda x_1,...,x_k|\varphi)s$ are p-terms.
11. If x is a variable or restricted variable and φ is a formula, then $(!x)(\varphi)$ is a p-term.
12. Every p-term is a term.
13. If $x_1,...,x_k$ are associative variables or associative symbols, and $s_1,...,s_{n+1}$ are p-terms, $n \geq 1$, then $s_1x_1...s_nx_ns_{n+1}$ is a term.

14. If x is a nonassociative infix function variable or infix function symbol and s,t are p-terms, then sxt is a term.
15. If x is a variable, y is an infix relation variable or infix relation symbol or = or ~ or $\in$, and s is a term, then xys is a restricted variable.
16. Every sentence symbol is a p-formula.
17. If s,t are terms, then s$\uparrow$ and s$\downarrow$ are p-formulas.
18. If x is a prefix relation symbol and $s_1,...,s_n$ are terms, $n \geq 0$, then $x[s_1,...,s_n]$ is a p-formula.
19. If $t,s_1,...,s_n$ are terms, $n \geq 0$, then $t[s_1,...,s_n]$ is a p-formula.
20. If $x_1,...,x_n$ are infix relation variables or infix relation symbols or = or ~ or $\in$, and $s_1,...,s_{n+1}$ are terms, $n \geq 1$, then $s_1x_1...s_nx_ns_{n+1}$ is a p-formula.
21. If φ is a formula, then (φ) is a p-formula.
22. If φ is a p-formula, then $\neg\varphi$ is a p-formula.
23. If $x_1,...,x_k$ are variables, $k \geq 1$, and φ is a p-formula, then $\forall x_1,...,x_k\,\varphi$ and $(\forall x_1,...,x_k)\varphi$ and $\exists x_1,...,x_k\,\varphi$ and $(\exists x_1,...,x_k)\varphi$ and $\exists! x_1,...,x_k\,\varphi$ and $(\exists! x_1,...,x_k)\varphi$ are p-formulas.
24. If $x_1,...,x_k$ are restricted variables, $k \geq 1$, and φ is a p-formula, then $(\forall x_1,...,x_k)\varphi$ and $(\exists x_1,...,x_k)\varphi$ and $(\exists! x_1,...,x_k)\varphi$ are p-formulas.
25. Every p-formula is a formula.
26. If $\varphi_1,...,\varphi_n$ are p-formulas, $n \geq 2$, then $\varphi_1\wedge...\wedge\varphi_n$ and $\varphi_1\vee...\vee\varphi_n$ are formulas.
27. If φ,ψ are p-formulas, then $\varphi\Rightarrow\psi$ and $\varphi\Leftrightarrow\psi$ are formulas.