

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/

February, 1997

I really should be talking to you about more mainstream things like face recognition information retrieval passing the Turing test.

I have some wonderful code that solves these problems completely, and I want to share it with you now.

```
010111010101000000101011101010101010101010010010111010001001
01010101101010100101001000010101010100101010101001011101010
0101010101010101010101110001001010101010101001000101010101110
101001010101001010001000101010101010011010101010101010101010101
0101010100100001101010101010101011101010101010101010101010010
0101010101001010010000010101001101010101111110110010101010101
1110010101010100101010101001010100010101001010101010101100100
1010101010010101010101010101001001010101010101001000101010101
01010100110010101011101011101001010101001010101010101001010101
0101010100101010101010010100101000001001011001101100101010101
010100111010101001100010101010101010101010100101010010101010101
010100111011011011011010101011100000101010101010010101011001010
101001010010
```

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?

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. Perhaps one can do a lot here without going too far with convenience; but more convenience than usual seems appropriate.
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 - largely not by accident. 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 lay the groundwork for the yet more ambitious project of developing a convenient way to prove the correctness of substantial computer programs. There are other issues that need to be addressed in order to accomplish this such as overhauling the present programming languages.

STANDARD FORMAL SET THEORY

The language has the following:

- i) connectives $\neg, \wedge, \vee, \rightarrow, \leftrightarrow$;
- ii) variables x_1, x_2, \dots 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 $\neg A, A \wedge B, A \vee B, A \rightarrow B, A \leftrightarrow B, (\forall x_n)(A)$ and $(\exists x_n)(A)$.

There are the nine usual axioms (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, $\{x: A\}$ 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
INFORMAL DISCUSSION

Work in joint progress with Randy Dougherty.

Again we stick to mathematical text without proofs. We need to shift to class theory. This is well known to be intimately connected with set theory. All objects will be classes. Some classes are "small" and are considered sets. Some classes are too big to be sets, and they are not members of any other classes. We use $M(x)$ to indicate that x is a set.

When a variable is used, one must know its range of possible values (by a well formed formula in the language).

When a constant is introduced, it must be given a definition. The most usual definition completely defines the constant as the unique object obeying some condition. However, more generally, we allow a constant to be defined as any object satisfying some given condition, with the understanding that if no object satisfies the given condition, then the constant is not defined. If the constant c is not defined then we can write this as $c\uparrow$.

A semantic symbol is introduced in exactly one of the following roles:

- i) as a k -ary prefix relation symbol for some $k \geq 1$;
- ii) as a (binary) infix relation symbol;
- iii) as a k -ary prefix function symbol for some $k \geq 1$;
- iv) as a (binary) infix function symbol;
- v) as a (unary) suffix function symbol.

If R is a k -ary prefix relation symbol and x_1, \dots, x_k represent classes, then $R(x_1, \dots, x_k)$ is viewed as either true or false. It is never undefined. If R is an infix relation symbol and x, y represent classes, then $x R y$ is viewed as either true or false. If F is a k -ary prefix function symbol and x_1, \dots, x_k represent classes, then $F(x_1, \dots, x_k)$ is viewed as either a unique class or undefined. If F is an

infix function symbol and x, y represent classes, then $x F y$ is viewed as either a unique class or undefined. Finally, if F is a suffix function symbol and x represents a class, then $x F$ is viewed as either a unique class or undefined.

When a semantic symbol is introduced, it is, optionally, given a definition. This definition is often incomplete. For example, one may introduce the semantic symbol $<$ as a binary relation symbol, and define it only for natural numbers. This does not mean that it is undefined outside the natural numbers (in fact, every actual relation is viewed as being defined everywhere); but rather, the meaning of $<$ for pairs of objects that are not both natural numbers is completely left open. Or for example, one may introduce the semantic symbol $+$ as a binary function symbol, and define it only for natural numbers.

Again, this does not mean that it is actually undefined outside the natural numbers, but rather that its meaning outside the natural numbers is left completely open.

Statements of claims are also given a name (like LEMMA 5.6 or like FUNDAMENTAL THEOREM OF ALGEBRA). The body of the claim is just a formula (perhaps with an explanatory clause). We allow certain variations that are convenient, such as making multiple claims, and using "Let" clauses to highlight hypotheses.

We now consider the crucial matter of correctness. One way of interpreting mathematical text without proofs is within the theory of classes. One inductively defines the concept of an interpretation of a term or formula based on an appropriate assignment of introduction clauses to signs in that term or formula, as well as the value of that interpretation or truth value of that interpretation (depending on whether it is a term or formula).

Roughly speaking, an interpretation consists of a nonempty domain of objects, together with a binary relation interpreting \in , and assignments of objects (sets or classes as appropriate) to every introduction clause, so that the condition in each introduction clause comes out true, where the quantifiers range over the objects that obey the condition in the governing introduction clause for the variable(s) being quantified.

The text is said to be true if every claim is true (as a statement in the theory of sets, classes, and superclasses) under all assignments.

The main thing we need to discuss is the formation of terms and formulas. We first informally discuss the special symbols:

$$\in = \neg \wedge \vee \rightarrow \leftrightarrow \forall \exists \lambda, () \{ \} \uparrow \downarrow ! M$$

The only symbols here that are not part of the usual set theory formalism are λ and $\neq \emptyset$ and $!$. As we shall see, the λ is used mainly to convert expressions into functions. Thus mathematicians are fond of saying things like "the function $3x + y - 7$." If this is meant to be a function of two variables, then we would write $(\lambda xy)(3x + y - 1)$.

The $\uparrow \downarrow$ are used to indicate that a term is undefined or defined. E.g., $1/0 \uparrow$ and $1/2 \downarrow$.

The braces $\{ \}$ are used not only for, say, the unordered pair $\{x, y\}$, but also for class abstraction in the form $\{x | A\}$. Of course, by Russell's paradox, we have $\neg M\{x | x \notin x\}$.

We use $!$ in connection with quantifiers; i.e., $(\exists !x)(A)$. Also it is used to denote the unique x such that I.e., $(!x)(A)$. If there isn't a unique x , then this is undefined.

The fundamental theorem of true texts asserts the following. Every formula appearing as a claim in a true text that only involves the primitives

$$\in = \neg \wedge \vee \rightarrow \leftrightarrow \forall \exists \lambda, () \{ \} \uparrow \downarrow ! M$$

must be universally true in the standard sense of class theory. A special case is of course that every formula appearing as a claim in a correct text that only involves $\in = \neg \wedge \vee \rightarrow \leftrightarrow \forall \exists () M$ and standard (unrestricted) variables is universally true in the standard sense of class theory.

We remind you that we are for the moment entirely unconcerned with any issues of provability - just truth. Of course we can step back and see what axioms of class theory we need to prove that the text is correct.

ABSTRACT TREATMENT OF FORMULAS AND TERMS

In this treatment of formulas and terms, we start with the symbols

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

and a set V of variables, a set PF of prefix function symbols, a set IF of infix function symbols, a set IR of infix relation symbols, a set PR of prefix relation symbols, and a set SF of suffix function symbols. We assume that V, PF, IF, IR, PR, SF are pairwise disjoint.

We also have 3 precedence relations. A precedence relation on a set is simply a function from that set into the integers (positive or negative or 0). The first precedence relation is on the connectives $\wedge \vee \rightarrow \leftrightarrow$; the second is on IF ; the third is on IR .

We now give the context free grammar for formulas and terms. We need to bring in the additional syntactic categories: bracketed formula and bracketed term.

1. For all $x \in V$, $y \in PF$, $z \in PR$, x is a bracketed term, y is a bracketed term, and z is a bracketed formula;
2. For all bracketed formulas B , $\neg B$ is a bracketed formula;
3. For all bracketed formulas B_1, \dots, B_n , $B_1 \text{ op } \dots \text{ op } B_n$ is a formula, where $n \geq 2$ and the op's are among $\wedge \vee \rightarrow \leftrightarrow$;
4. For all terms s, t , $M(s)$, $s \uparrow$, $s \downarrow$, $s \in t$, $s = t$, and $s \sim t$ are bracketed formulas;
5. For all terms t , (t) is a bracketed term;
6. For all formulas B , (B) is a bracketed formula;
7. For all terms t_1, \dots, t_n , $n \geq 1$, $\{t_1, \dots, t_n\}$ is a bracketed term;
8. For all terms s, t_1, \dots, t_n , $n \geq 1$, $s[t_1, \dots, t_n]$ is a bracketed formula and $s(t_1, \dots, t_n)$ is a bracketed term;
9. For all bracketed terms s , and $x \in PF$, $y \in SF$, $z \in PR$, xs , sy are bracketed terms and zs is a bracketed formula;
10. For all terms t_1, \dots, t_n and $x \in PR$ and $y \in PF$, $x[t_1, \dots, t_n]$ is a bracketed formula and $y(t_1, \dots, t_n)$ is a bracketed term;
11. For all $n \geq 2$, bracketed terms t_1, \dots, t_n , $x_1, \dots, x_n \in IF$, $y_1, \dots, y_n \in IR$, t_1, \dots, t_n is a term, and $t_1 y_1 t_2 \dots t_n$ is a formula;
12. For all terms t , $x \in V$, and formulas B , $\{x|B\}$ and $\{x \in t|B\}$ are bracketed terms;

13. Let B be a bracketed formula, $n \geq 1$, $x_1, \dots, x_n \in V$, t be a term. Then $\forall x_1, \dots, x_n B$, $\exists x_1, \dots, x_n B$, $(\forall x_1, \dots, x_n) B$, $(\exists x_1, \dots, x_n) B$, $\exists! x_1, \dots, x_n B$, $(\exists! x_1, \dots, x_n) B$, $\forall x_1, \dots, x_n \in t B$, $\exists x_1, \dots, x_n \in t B$, $(\forall x_1, \dots, x_n \in t) B$, $(\exists x_1, \dots, x_n \in t) B$, $\exists! x_1, \dots, x_n \in t B$, $(\exists! x_1, \dots, x_n \in t) B$, are bracketed formulas;
14. Let t be a bracketed term, $n \geq 1$, $x_1, \dots, x_n \in V$, s be a term. Then $\lambda x_1, \dots, x_n t$, $(\lambda x_1, \dots, x_n) t$, $\lambda x_1, \dots, x_n \in s t$, $(\lambda x_1, \dots, x_n \in s) t$ are bracketed terms;
15. Let t be a bracketed term, $n \geq 1$, $x_1, \dots, x_n \in V$, s be a term. Then $!x_1, \dots, x_n t$, $(!x_1, \dots, x_n) t$ are bracketed terms.

This grammar has unique parsing. Parsing is very efficient.

CONCRETE TREATMENT OF FORMULAS AND TERMS

The formalism is based on a fixed finite alphabet A . Although there probably is some wisdom in making this expandable, we will simplify matters by taking A to be fixed. These characters are identified with bytes (ASCII codes).

A is divided into characters, the blank, and the carriage return.

The characters are divided into visible and invisible characters only for the purpose of what is displayed on the screen in normal mode, and what the printout looks like. It is only when the screen is in special mode that the blanks, carriage returns, and invisible characters can be seen.

A must have at least the visible characters

$\epsilon = \neg \wedge \vee \rightarrow \leftrightarrow \forall \exists \lambda, () \{ \} [] \uparrow \downarrow | \sim M$

as well as the visible characters (unformatted) remaining on standard computer keyboards.

A name is a nonempty string from A that has no carriage returns and does not begin or end with a blank, comma, or period.

The formulas (terms) are the elements x of A^* which are uniquely (up to isomorphism) a substitution instance of an abstract formula (abstract term) subject to the following conditions:

- i) the substitutions are made by names;

ii) in the substitution, the names that are replaced by variables are exactly the names used in the substitution that start with a lower case English alphabetic character.

The recognition and parsing problems can be solved very efficiently.

THE SYNTAX OF TEXT

A name is a nonempty string from A that has no carriage returns and does not begin or end with a blank, comma, or period.

A (well formed) text is an element x of A^* satisfying certain conditions. It is required that x consist of a series of entries which contain no carriage returns, separated by at least one carriage return. It is understood that any number of blanks can be inserted anywhere in a text and the result will still be a text.

There are three types of entries. When reading each entry, one ignores blanks.

To determine the kind of entry, look for the first period. This must be one of the following:

CONVENTION x .

DEFINITION x .

x .

Here x must be a name. In the third case, x does not start with CONVENTION or DEFINITION. Typically, in the third case, x will be THEOREM, LEMMA, CLAIM, PROPOSITION, FACT, SUBLEMMA, LEMMATA, COROLLARY, etcetera.

The actual conventions (there may be more than one) are found after the first period in the first case. They are of the following forms:

x .

u has precedence k .

Precedence k is left associative.

Precedence k is right associative.

Here in the first case, x is a formula. The idea is that x asserts that the free variables in x are to range over those choices which make x true. The most normal case is

where x has exactly one free variable, say y , and we are simply declaring the range of the variable y ; e.g., x is just $M(y)$. This is like declaring that y is a variable ranging over all sets.

In the second case, we are designating the precedence of u . Here u is any name (including a single binary connective). This is referred to if and when u is used in a formula as a single binary connective or as an infix symbol, for the purposes of fixing the ultimate parsing.

The actual definitions are found after the first period in the second case. They are of the following forms:

```
Define  $R[x_1, \dots, x_n]$  as  $Q$ ;
If  $P$  then define  $R[x_1, \dots, x_n]$  as  $Q$ ;
Define  $F(x_1, \dots, x_n)$  as  $t$ ;
If  $P$  then define  $F(x_1, \dots, x_n)$  as  $t$ .
```

Here R, x_1, \dots, x_n, F are names and P is a formula and t is a term; x_1, \dots, x_n begins with a lower case alphabetic letter. The free variables of Q as well as t must be among x_1, \dots, x_n .

Conflicts in definitions are resolved by latest updates. This is a little tricky in full generality.

The claim entries are of the form

P .

where P is a formula. In interpreting P , one uses the earlier definitions and conventions.

SEMANTIC ASPECTS

From the semantic point of view, every text is a presentation of mathematical information that sits inside the theory of classes, as represented by a version of the von Neumann Bernays class theory with the global axiom of choice - VBGC - based on classes of sets. The truth definition is carried out in an appropriate superclass theory, based on classes of classes of sets.

Concentrating on the present concept of text finesses the issue of treating mathematical text that refers to various other mathematical texts, which may not have consistent

notation with each other. This is an issue my coauthor especially likes - Randy Dougherty at the Ohio State mathematics department.

In the appropriate version of VBGC, every object is a class. If the class X is a set then we write $M(X)$. The sets are just the classes that are elements of some class. Hence every element of a class is a set. Classes that are not sets are called proper classes. The unordered pair, union, and power set of any set is a set. There is an infinite set. Two classes are equal if and only if they have the same elements. Every nonempty class has an element which has no element in the class. From these axioms, we have ordered pairing for sets. Therefore, we know what we mean by a class being a function. The image of every function on a set is a set. There is a function which produces an element of any nonempty set to which it is applied. Finally, we have separation for classes. This asserts that we can form the class of all sets satisfying any first order formula in our language, provided all quantifiers in the formula are restricted to sets.

VBGC is well known to be a conservative extension of ZFC = Zermelo Frankel set theory with the axiom of choice, in which all variables range over sets only (no proper classes). I.e., every sentence provable in VBGC in which all quantifiers range over sets is provable in ZFC (with the relativizations of the quantifiers removed). Furthermore any sentence provable in ZFC is provable in VBGC if the quantifiers are relativized to sets.