CHECKING MATHEMATICS WITH THE AID OF A COMPUTER

N.G. de Bruijn

Department of Mathematics and Computing Science,
Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven,
The Netherlands.

In:

The Influence of
Computers and Informatics on
Mathematics and its
Teaching

p. 61 - 68

ICMI Study Series Editors: A. G. Howson and J.-P. Kahane

CAMBRIDGE UNIVERSITY PRESS
Cambridge
London New York New Rochelle
Melbourne Sydney

1986

CHECKING MATHEMATICS WITH THE AID OF A COMPUTER

N.G. de Bruijn

Department of Mathematics and Computing Science,
Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

0. Computers influence mathematics in many ways. This paper is devoted to one of these influences: the fact that we can explain mathematics to a computer. In this process we may learn about how to organize mathematics and how to teach some of its aspects.

At Eindhoven University of Technology, The Netherlands, the project Automath was developed from 1967 onwards, with various kinds of activities at the interfaces of logic, mathematics, computer science, language and mathematical education. Right from the start, it was directed towards the presentation of formalized knowledge to a computer, in a very general language, with quite a strong emphasis on doing things the way humans do. One might say that the project is a modern version of "Leibniz's dream" of making a language for all scientific discussion in such a way that all reasoning can be represented by a kind of algebraic manipulation.

The basic idea of Automath is that the human being presents any kind of discourse, how long it may be, to a machine, and that the machine convinces itself that everything is sound. All this is intended to be effectively carried out on a large scale, and not just "in principle".

This paper does not intend to describe the Automath system in any detail, but rather to explain a number of goals, achievements and characteristics that may have a bearing on the subject of the ICMI discussion on the influence of computers and informatics on mathematics and its teaching.

The paper is definitely not trying to sell Automath as a subject to be taught to all students in standard mathematics curricula. The claim is much more modest: as Automath connects so many aspects of logic, mathematics and informatics, it may be worth-while to investigate whether the teaching of mathematics could somehow profit from ideas that emerged more or less naturally in the Automath enterprise. The idea of Automath is to "explain things to a machine". Students are no machines and should be approached in a different way. But as teachers we should know that if we cannot explain a thing to a machine then we might have difficulties in explaining it to students.

1. In the Automath system the mathematical material is written in the form of a complete book, line by line. A computer can check it line by line, and once that has been done, the book can be considered as mathematically correct.

The interpretation of such a book can be a complete theory, containing all axioms, definitions, theorems and proofs.

2. As a starting point we think of a book written entirely by human beings. Later on we may think of leaving part of the writing to a machine. That part might be simply the tedious routine work, but possibly also the more serious problem solving (i.e., "theorem proving", a branch of artificial intelligence).

In order to be succesful in the hard task of problem solving it might be profitable to temporarily leave the format of the Automath languages. In a way one might say that in this area generality and efficiency are conflicting objectives. The Automath project made a choice here: it never concentrated on automatic theorem proving, but just on checking.

3. We should make a clear distinction between the Automath system and Automath books. The system consists, roughly speaking, of language rules and a computer program that checks whether any given book is written according to those rules.

The system of Automath is mainly involved with the execution of substitution, with evaluation of types of expressions, and comparing such types to one another. It is very essential that everything that is said in a book, is said in a particular context: the context consists of the typed variables that can be handled, but also of the list of assumptions that can be used. The system keeps track of those contexts.

The Automath system does not contain any a priori ideas on what is usually called logic and foundation of mathematics. Any logical system (e.g., an intuitionistic one) can be introduced by the user in his own book, and the same thing holds for the foundation of mathematics. In particular, the user is not tied to the standard 20-th century set theory (Zermelo-Fraenkel). And the user can choose whether to admit or not to admit things like the axiom of choice. From then on, the machine that verifies the user's book will be able to do this according to the user's own standards.

4. In an Automath book, logic and mathematics are treated in exactly the same way. New logical inference rules can be derived from old ones, just like mathematical theorems are derived, and the new inference rules can be applied as logical tools, in the same way as mathematical theorems are applied.

5. Writing in Automath can be tedious. All details of arguments have to be presented most meticuously. At first sight this might be very irritating. The questions are (i) whose fault this is, and (ii) what can be done about it.

The questions are related. Part of the negative impression that the length of an Automath book makes, is due to the fact that no attempt was made to "do something about it" at the stage of the design of the general system. This is based on the philosophy that generality comes first, and that adaptability to special situations is a second concern.

The reason why Automath books become so long is that mathematicians have more in their mind than they explain, and nevertheless we want to handle all usual mathematical discourse.

Perhaps we may say that part of mathematical work is done subconsciously. Mathematicians have a vast "experience" in mathematical situations, and such experience may give a strong feeling for how all the little gaps can be filled. Possibly much of the experience is consulted subconsciously "on the spot".

Moreover, mathematical talking and writing are social activities. In every area, people talk and write in a style they know they can get away with. Some poor or incomplete forms of discourse are so wide-spread that it seems silly to bother about improvements; certainly it is not a very rewarding task to try.

The answer to question (ii) is that very much can be done about it indeed. But just like every user can write his own book under the Automath system, he can implement his own attachments to the system. This may involve special abbreviation facilities, but also automatized text writing, producing packages of Automath lines by means of a single command, in cases where there is a clear system behind such a package.

6. Are computers essential for Automath? Not absolutely. The computer sets the standard for what the notion "formalization" means. If we cannot instruct a computer to verify mathematical discourse, we have not properly formalized it yet. In the standard form, the author of an Automath book has to write all the symbols one by one, and since he knows that what he writes is correct, he would also be able to check it by hand.

Nevertheless humans make mistakes. Automath books have been written with a number of characters of the order of a million, all typed by hand. It is hard to guarantee correctness of such a text without the help of a modern computer.

7. As the Automath system has no a priori knowdledge of logic and set theory, it can be used to write in a style that might be more natural than what we see in other formalizations.

There is a wide-spread idea that propositional logic comes down to manipulating formulas in a boolean algebra, a kind of manipulation that is either carried out by handling formulas with the aid of lists of tautologies (in the same way as one used to do in trigonometry), or by a machine that checks all possibilities of zeros and ones as values for the boolean variables. A very much better formalization lies in the system of "natural deduction". This is very easy in Automath. The boolean bit-handling propositional logic can be done in Automath too, but is is much more clumsy than natural deduction.

A second option we get from the liberty of using Automath in the style we prefer, is to give up the 20-th century idea that "everything is a set". There is the magic Zermelo-Fraenkel universe in which every point is a set, and somehow all mathematical objects are to be coded as points in that universe. The particular coding is a matter of free choice: there is no natural way to code.

Zermelo-Fraenkel set theory is quite a heavy machinery to be taken as a basis for mathematics, and not many mathematicians actually know it. An alternative is to take "typed set theory", in which things are collected to sets only if they are of the same type: sets of numbers, sets of letters, sets of triangles, etc. It may takes some trouble to make up one's mind about the question what basic rules for typed set theory

should be taken as primitives, but if we just start talking the way we did mathematics before modern set theory emerged, we see that we need very little. Anyway, in Automath we have no trouble at all to talk mathematics in a sound old-fashioned way.

Yet, if someone still wants to talk in terms of the Zermelo-Fraenkel universe, Automath is ready to take it.

8. One of the advantages of Automath not being tied to any particular system for logic and set theory, is that we can think of formalizing entirely different things too, again in a natural style. As an example we may think of the algorithmic description of geometrical constructions like those with ruler and compass. Although it has not actually been produced, we may think of a single Automath book containing logic, mathematics and the description of ruler and compass constructions, with in particular the description and correctness proof (both due to Gausz) of the construction of the regular 17-gon. This description will be quite different from coding the construction as a point in the Zermelo-Fraenkel universe. We might even think of a robot equipped with ruler, compass, pencil and paper, who reads the details of the construction from the Automath book and carries them out in the way Gausz meant.

9. Many parts of science are patchwork consisting of pieces of theory, connected by rather vague intuitive ideas. Ever since the last part of the 19-th century it has been one of the ideas of the mathematical community that mathematics should be integrated: all parts of mathematics are to become sub-domains of one single big theory. The patchwork picture still applies to most physical sciences, but also to several parts of the mathematical sciences. One such part is informatics.

It seems to be a good idea to integrate informatics into mathematics, at least in principle. And, as in the case of geometrical constructions, Automath is a good candidate for describing this. It is possible to write an Automath book containing: logic, mathematics, description of syntax and semantics of a programming language, and particular programs with proofs that the execution achieves the solution of particular mathematical problems. One might even think of going further: description of the computer hardware with proof that it guarantees the realization of the programming language semantics. Or directly, without the intervention of a programming language, that a given piece of hardware produces a result with a given mathematical specification.

Needless to say, this kind of integrated theory will always contain a number of primitives we have no proof for, but it will be absolutely clear in the Automath book what these primitives are.

10. One thing people like in Automath, and other people strongly dislike, is the way Automath treats proofs as if they were mathematical objects. This is called "propositions as types". As the type of a proof we have something that is immediately related to the proposition established by that proof.

One should not be worried about this. Automath does not say that proofs are objects, but just treats them syntactically in the same way as objects are treated. This turns out to be very profitable: it simplifies the system, as well as its language theory and the computer

verification of books. A third case where things are treated as objects is the one of the geometrical constructions we mentioned in section 8.

11. In standard mathematics, most identifiers are letters of various kinds, possibly provided with indices, asterisks and the like. And then there are the numerals, of course. We have learned from programming languages, however, to use arbitrary combinations of letters and numerals as identifiers, (with restrictions like not to begin with a numeral). We do the same thing in Automath, thus having the possibility to choose identifiers with a mnemonic value, like "Bessel", "Theorem137", "commutative". This certainly helps to keep books readable.

In contrast to programming languages, the Automath system does not have the numerals 0,1,...,9. One can introduce them as identifiers in a book containing the elements of natural number theory, taking "0" and "succ" (for "successor") as primitive, and defining 1:=succ(0), 2:=succ(1),..., 9:=succ(8), ten:=succ(9). After having introduced addition and multiplication, we can define things like : thirtyseven:=sum(prod(3,ten),7), but the Automath system has no facilities to write this as 37. This decimal notation might be added as an extra (it is one of the possible "attachments" mentioned in section 5).

12. One of the basic aims of the Automath enterprise was to keep it feasible. This has been achieved indeed: considerable portions of mathematics of various kinds have been "translated" into Automath, and the effort needed for this remained within reasonable limits. If we start from a piece of mathematics that is sound and well understood, it can be translated. It may always take some time to decide how to start, but in the long run the translation is a matter of routine. As a rule of thumb we may say there is a loss factor of the order of 10: it takes about ten times as much space and ten times as much time as writing mathematics the ordinary way. But it is not overimportant how big this loss factor is (it would not be hard to reduce it by means of suitable attachments, adapted to the nature of the subject matter). What really matters is that it does not tend to infinity, which happens in many other systems of formalizing mathematics. The main reason for the loss factor being constant is that Automath has the same facilities for using definitions (which are, essentially, abbreviations) as one has in standard mathematics. The fact that the system of references is superior to what we have in standard mathematics, makes it possible that the loss factor even decreases on the long run when dealing with a large book.

13. Another feature that makes Automath feasible is that we need not always start at the beginning: we can start somewhere in the middle, and if we need something that we have not defined, or have not proved, we just take it as a primitive (primitive notion or axiom) and we go on. We can leave it to later activity to replace all these primitives by defined objects and provem theorems.

This kind of tactics was often (about 30 cases) applied at Eindhoven by students (mathematics majors). It usually took the student not much more than 100 hours work to learn about the system, to translate a given piece of mathematics, to use the conversational facilities at a computer terminal, and to finish with a completely verified Automath book containing

the result. In order to give an idea of the subjects that had to be translated we mention a few: (i) The Weierstrasz theorem that says that the trigonometric polynomials lie dense in the space of continuous periodic functions, (ii) The Banach-Steinhaus theorem, (iii) The first elements of group theory.

14. Of the more extensive books that were written in Automath we mention two. The first one is L.S. van Benthem Jutting's complete translation of E. Landau's Grundlagen der Analysis. In order to test the feasibility of the system, the translator kept himself strictly to Landau's text, rather than inventing some of the many possible shortcuts and improvements that would make the translation easier and shorter. The second one we mention here was by J.T. Udding, who wrote a new text with about the same results, much better suited to the Automath system, both in its general outline and in its details. The gain over Landau's text, in space as well as in time, was roughly 2.5.

15. One of the ideas of the Automath enterprise was to get eventually to a big mathematical encyclopaedia, a data bank, containing a vast portion of mathematics in absolutely dependable form. This is a thing that would take many hundreds of man years (thus far the Automath project took something like 40). But the idea is feasible. Most of the students mentioned in section 13 used the Landau translation (see section 14) as a data bank, and that way they added to the bank.

16. It is not the purpose of this paper to enter into details of the Automath language, but the reader might want a general orientation.

There are several dialects of Automath in use, but here we only look into basic things they have in common.

The expressions used are always lambda-typed lambda calculus expressions. This means that we have lambda expressions where every variable has a type, and that type is again a lambda expression.
In lambda calculus the basic expression-forming devices are "abstraction" and "application", but in Automath we have a further device, called "instantiation". Instantiation is the operation that leads from an n-ary prefix operator f to an expression f(E1,...,En), where E1,...,En are expressions.

Having both "instantiation" and "application", Automath has two different devices for expressing functionality, and both can be linked to standard mathematical practice.

In Automath we write mathematics in the form of books, line after line. There are three kinds of lines: (i) context lines, (ii) definitional lines, and (iii) primitive lines.

A context line sets the context for the sequence of non-context lines between that context line and the next one. A context is a sequence of variables provided with types. We denote typing by a colon, and describe a typical context of length 3:

$$x : A, \qquad y : B(x), \qquad z : C(x,y)$$

(here A, B(x), C(x,y) denote expressions; C(x,y) is an expression containing the variables x and y).

A definitional line describes an abbreviation. It takes an expression E (of type F), and abbreviates that E by a new identifier, c, say. The line looks like

c := E : F.

A primitive line introduces some new identifier as a primitive notion, and attaches a type to it. That is, it is not defined, but declared to be available for further use. So it looks just like the definitional line above, but without the E. In order to stress that the defining expression E is omitted, we write a fixed symbol (like PN, or 'prim') in its place:

d := 'prim' : F.

These scanty remarks might give an idea about what the languages look like; for detailed description we refer to de Bruijn (1970), and for an informal introduction into the use of the language also to de Bruijn (1973).

We refer to de Bruijn (1980) for a survey of the whole project, more extensive than the one given here. And also van Benthem Jutting (1979) will give a good idea about the project and the languages, but on top of that it is a report of all the experience obtained in the Landau translation mentioned in section 14.

REFERENCES

N.G. de Bruijn (1970). The mathematical language AUTOMATH, its usage and some of its extensions. Symposium on automatic demonstration. IRIA, Versailles, December 1968. Lecture Notes in Mathematics, Vol. 125, Springer Verlag.

N.G. de Bruijn (1973). AUTOMATH, a language for mathematics. A series of lectures at the Séminaire de mathématiques supérieures, Université de Montréal, June 1971. Lecture notes by B. Fawcett. Les Presses de l'Université de Montréal.

N.G. de Bruijn (1980). A survey of the project AUTOMATH. In: To H.B. Curry: Essays in combinatory logic, lambda calculus and formalism, ed. J.P. Seldin and J.R. Hindley. Academic Press.

L.S. van Benthem Jutting (1979). Checking Landau's "Grundlagen" in the AUTOMATH system. Mathematical Centre Tracts nr. 83, Amsterdam.