The Automath mathematics checking project and its influence on teaching
de Bruijn, N.G.

Published: 01/01/1985

Document Version
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author’s version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher’s website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Citation for published version (APA):

by

N.G. de Bruijn.

THE AUTOMATH MATHEMATICS CHECKING PROJECT AND ITS INFLUENCE ON TEACHING

University of Technology,
Department of Mathematics,
P.O. Box 513,
5600 MB Eindhoven,
The Netherlands.
The Automath mathematics checking project
and its influence on teaching

by N. G. de Bruijn

1. Automath

Computers influence mathematics in many ways. One of these lies in the fact that we can learn to explain mathematics to a computer, and in this process we may learn about how to organize mathematics and how to teach some of its aspects.

At the Technological University Eindhoven (Eindhoven, 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 knowledge by means of symbolic manipulation, with the possibility to leave much of the work to a computer, with quite a strong emphasis on doing things in a humanly way. One might say that it 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. 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 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.

1.1. A basic idea of Automath is to write 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.

1.2. 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 tedious routine work, but also possibly the more serious problem solving (i.e., "theorem proving", a branch of artificial intelligence).

1.3. 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.

1.4. 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.

1.5. Writing in Automath can be tedious. All details of arguments have to be presented most meticulously. 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 we
claim to be able to handle all usual mathematical discourse,
but the mathematician has more in his mind than he explains.
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.

1.6. 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.

1.7. As the Automath system has no a priori knowledge 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 it 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 take 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 Zermelo-Fraenkel universe, Automath is ready to take it.

1.8. 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.

1.9. 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 day, 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.

1.10. 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 1.8.

1.11. 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", "Theoreml37", "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 1.5).

1.12. 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.

1.13. 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 proven 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.

1.14. Of the more extensive books that were written in Automath
we mention two. The first one is L.S. 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.

1.15. 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 1.13 used the Landau translation (see 1.14) as a data
bank, and that way they added to the bank.

2. Standard mathematical language.

In close connection with Automath a language was studied with the same level of precision, but closer to ordinary language as written by mathematicians, at least when they are very precise. Let us call it MV (for "mathematical vernacular"). MV is the familiar mixture of words and formulas in which some of the letters and formulas play a syntactic role just as if they were ordinary parts of a sentence, like subject, direct object, etc.

2.1. It is possible to formulate logic and the foundation of mathematics in terms of the grammar of such a language. The grammar of MV can be kept quite simple, since all sorts of idiom of natural language can be caught in terms of definitions in the book. This way we do not need to distinguish more then the following four grammatical categories: (i) sentences, (ii) substantives, (iii) names, (iv) adjectives. Each one of these four can occur as a group of words, but also as a mathematical symbol, a formula, or a mixture of words and formulas. The four categories correspond to the four kinds of definitions that mathematicians give. In the definitions of the first kind the new term is a sentence (like:"we say that p divides q if ..."), in the second case it is a substantive ("a square is a ..."), in the third one a name (... is called the n-th Bessel coefficient), in the fourth an adjective (a sequence is called convergent if ...).
2.2. The difference in syntax is not the only difference between Automath and MV. The main difference is that in Automath each line contains exactly all information about how the stated result follows from previous lines: all theorems and inference rules which are used are mentioned, and their role is made absolutely clear. In MV such indications do not belong to the language itself, but can be considered as having been written in the margin. In other words, in Automath they are language, in MV metalanguage.

One can use MV as a stage in the process of writing in Automath. If the steps in MV are small, and if the indications in the margin are sufficiently clear, the translation into Automath is a routine matter.

2.3. Inspecting textbooks in mathematics on school level one finds very little MV. Most of the texts are written in metalanguages of various kinds. Quite often, the intersection of the text with its own representation in MV is little more than the mathematical formulas, i.e. the part that was formalized hundreds of years ago.

3. Effects on mathematical education.

The question was: "How do computers and informatics influence mathematical ideas, values and the advancement of mathematical science?". There will be all sorts of influences, like the taste for constructivity, and, as far as education is concerned, the new possibilities to let students have their own stimulating discoveries with the aid of a computer. But
the influence we get from the fact that we can explain mathematics to a computer, should not be forgotten. We shall look into this in some detail.

3.1. First, there are the philosophic aspects. Is it really mathematics we explain to a computer? Or is it just some piece of code we happen to interpret as mathematics? How arbitrary is our interpretation?

There is no definite answer to such questions. If we have to compare a formal system to something that is partly intuitive, then the comparison cannot be completely formal.

For example, in the partially intuitive mathematical world, the question whether the mathematical objects exist in a platonic reality, might seem to make some philosophical sense. But if we consider a completely formalized version to be explained to a computer, such a question cannot even be formulated. Some people will react by saying that this definitely puts an end to platonism, others will say that it shows that no formalization will ever be complete.

3.2. Having to phrase our mathematics in a very definite language, we have to make clear what part of ordinary mathematics belongs to the language and what part is metalanguage. Many paradoxes arise just by confusing language and metalanguage. Making the distinction will certainly help to understand mathematics better.

3.3. Today, most mathematicians have the idea that the foundation of mathematics is too hard to learn for a non-specialist, and can only be taught to students who know mathematics already. This means that the foundations of the
building of mathematics are laid only after the building is completed, so they can impossibly play the role of the basis of mathematics. The teaching of the foundations at that late stage assumes the students to be acquainted with mathematical ideas (the role of definitions, axioms, theorems) for which one expects the foundations to give explanations. On a lower level, the same thing happens in the boolean propositional calculus: it is a mathematical system which is erected by standard mathematical techniques, and nevertheless it is a popular belief that it can explain what logic is, what proofs are.

3.4. Outsiders would be very surprised to hear that mathematicians are so vague about their own foundations, even now, towards the end of the 20-th century, that great century for logic.

If one really takes the task seriously to write (like it can be done in Automath) the foundations of mathematics up to a level such that the working mathematician would be able to build on it, one will see that it is not at all that hard. A sound basis can easily be given at the age of 17 to 19. For many questions about the relation between mathematics and computers (questions like program correctness) it is very essential to have such a basis.

Of course, the basis need not be given itself in a formal language. It can be quite informal, but the teacher should know the formal background.

The method of natural deduction is a very good candidate for explaining the foundation of mathematics. It opens the possibility to treat the introduction and elimination rules of the propositional calculus in exactly the same style as those of the predicate
calculus. Moreover, it can be pointed out to the student, by means of an informal metalanguage, what is a proof, an axiom, a definition, an assumption, a theorem. And it opens the way to understanding notions that cannot be properly explained at all on an informal basis. In this connection we mention the notion of existence, which has remained a mystery to many generations of mathematicians.

3.5. A foundations course at an early stage should be recommended. This is not only because of the computer; another important reason is the disintegration of the teaching of geometry.

Traditionally, school geometry used to give the initiation into mathematical reasoning. Other mathematical subjects used to train the art of calculation, not the art of proof. But geometry had its drawbacks: it was hard and unattractive to keep the reasoning pure, i.e., to remove every appeal to what we learn by observation of the physical world. In particular this refers to the matter of order on the line and in the plane. Another drawback was that quite often the arguments failed in some exceptional, often trivial, situations, and that these had to be treated separately. And a satisfactory treatment of the axiomatic basis was too difficult to be treated at school. And, lastly, the logical content was so limited: no predicate calculus, no quantifiers, apart from a few cases where sets played a role (the geometric loci). On the other hand, geometry showed a wonderful interplay between intuition and argumentation.

Possibly because of the drawbacks mentioned here, traditional school geometry was almost entirely discarded in most countries, and replaced by the study of "structures", called "new math". In these new subjects there was hardly a chance to train the art of
proof, and now we are left with the sad situation that upon entry
of the university the students, even mathematics and computer science
majors, are very weak in this respect.

3.6. In many parts of the new math, in particular in algebraic
areas, it is quite hard to draw the borderline between mathematics
and metamathematics (cf. 2.3). And reasoning about sets, with or
without Venn diagrams, is often on a low logical level. In particular,
it gives hardly any opportunity for handling variables. It has to be
admitted that the innovations in mathematical education have given us
quite some progress, both in insights as in practical applicability,
but the price we paid by neglecting the art of proof may have been
too high.

3.7. Mathematics majors on the university level usually
learn to handle predicate calculus in courses on the
foundation of analysis. At least they learn it implicitly,
on a practical basis, and directly tied to the formalization
of notions with an intuitive background, like uniform
convergence.

Needless to say this kind of material will become gradually
harder now that the students enter the university with such a
poor preparation in the art of proof.

Another matter is that it is no longer clear whether
informatics students should take courses in the foundation of
analysis. There is a danger that in the near future the only
intersection of the curricula for mathematics and informatics
will be some kind of simple calculus.

3.8. As to teaching the art of proof, it may be a good
idea not to tie it to geometry, and not to any new subject like combinatorics, set theory or algebra, but to take it as a subject in its own right, in the form of an elementary logics course.

As a kind of experiment such a course was tried for computer science students, right from school, at the Technological University Eindhoven since 1982. It seems to have been successful in teaching the structure of proof by means of explaining the rules of the game of propositional and predicate calculus. The basis was natural deduction (cf. 3.4). Only after the building of logic was erected, it was shown how the notion of valuation gives the link with the boolean algebra aspect.

The course started with a chapter on syntax, involving the study of parentheses, representation of formulas as trees, infix notation, bound variables, lambda calculus notation, substitution, etc. It turned out to be illuminating to take the trees as the central theme, in particular in connection with substitutions in formulas with bound variables.

In the treatment of predicate logic, predicates were taken to be defined on sets, and in that respect the course took a naive point of view. It was not attempted to develop the language of mathematics in all its glory: that would probably have taken twice as much time as could reasonably be devoted to the course.

This introductory course on logic took not more than 18 hours teaching, with about 24 hours added for excercises.

In a sequel of this course (again 18 hours teaching plus excercises), applications were made to mathematical fundamentals (treatment of sets and mappings, the system of natural numbers, the method of induction, recursion and definition by recursion), but also to a number of subjects on the borderline of mathematics.
and informatics. These were mainly: the terminology of the free
monoid and its relation to language, context-free grammars
in a mathematical setting (with terminals and non-terminals),
and the relation of this with the Backus-Naur form.

3.9. A course like the one described in 3.8 might be
recommended as the body of the intersection of the curricula
of mathematics and informatics.

What might be added to the intersection is a mathematical
description of what is a computer, a program, input, output,
program specification and program correctness. At that stage
it is better not to go into details of a programming language,
apart from the description how such languages can be defined
by recursion.

3.10. Parts of the logics course, like syntax and
propositional calculus in natural deduction, might be
shifted to the school age (16-18 years). The natural
deduction would be very appropriate for showing what a
proof is, and it would raise the teaching of logic above
the "trigonometry level" (cf. 1.7). And lambda calculus
might really help to make school mathematics easier.

3.11. Some of the material mentioned in section 2 was taught
at Eindhoven since about 1977 in a course called "Language
and structure of mathematics", for those mathematics
majors who wanted a teachers certificate in mathematics.
Much of it would be fit for all mathematics majors at an
eyear stage of their university career.
References on Automath:

N.G. de Bruijn, A survey of the project Automath.