Research Proposal

Drs. A.J.M. van Gasteren and prof. dr. Edsger W. Dijkstra.

Abstract. Our effort has been inspired by the impression that the effectiveness of the average mathematical argument can be greatly increased; our aim has been to confirm that impression by discovering general techniques for effectuating such improvements.

The impression was correct; even many a not-so-average mathematical argument turns out to leave room for considerable improvement.

Great improvement could be achieved by assigning a greater rôle to calculi of all sorts, the predicate calculus in particular. In view of the intrinsic generality of the predicate calculus —as stratification of all rigorous reasoning— its development into a workable tool was made a major focus of our efforts.

Techniques other than calculational, developed so far, are also effective beyond the environment of mathematics.

We envisage the next phase to be devoted to tasks such as

(i) the development of calculi and the extension of their range of applicability;

(ii) learning how to design suitable notations that cater for one's manipulative needs;

(iii) discovering how to propagate, explain, and teach what we have learnt by trying to propagate, explain, and teach it.

Our plans for the years to come

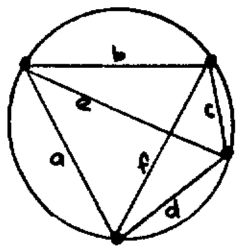
"It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last. The development of this relationship demands a concern for both applications and for mathematical elegance."

John McCarthy, 1967.

Two accidents of history have retarded the widespread recognition of programming as a mathematical activity. The one accident was the development of average programming practice, in which programmers without mathematical inclination are fully absorbed by the intricacies of their tools and the fuzziness of their tasks. The other accident is that the style of mathematics the programming task requires turned out to differ so radically from the style of mathematics developed so far, that most mathematicians (including authoritative ones) failed to recognize programming as a human endeavour amenable to mathematical treatment. (In addition, the existence of computers that can execute the programs has misled people to believe that program correctness is an experimental notion, thus adding to the confusion.)

The mathematical styles in which proofs are conducted are the product of slowly developing traditions. In principle, these traditions are very pragmatic: mathematicians do not like to waste their time on demonstrating the obvious. Each time, however, that something is accepted as "obvious", a certain risk is taken; in mature branches of mathematics there is a well-established consensus about which "risks" can be taken safely, and in that sense established styles are adequate.

Taking an example from plane geometry, we consider four points on a circle. These points determine six chords, a, b, c, d, e, and f, say.



The three possible pairings of the points give rise to three pairs of chords, viz. (a,c), (b,d), and (e,f). An example of an "obvious fact" is the statement that of the three pairs of chords one pair intersects inside the circle - (e,f) in the above - and two pairs intersect outside the circle. Practical reasons fully justify an appeal to this fact without proving it. In contrast with this, most mathematicians agree that the validity of the relation $e \cdot f = a \cdot c + b \cdot d$ between the lengths of the chords requires a proof. As a result, the statement has been given a name (it is the theorem of Ptolemy).

We point out that the existing styles just grew, are rarely the result of well-considered choices, and are by no means necessarily the most effective ones. Moreover, there is no reason to assume that, having grown in response to actual demands, they are adequate in new areas of mathematics.

The intellectual content of the "computer revolution" seems to be that the design of programs we can justifiedly rely upon faces us with mathematical problems that are in a very true sense without precedent in the history of mathematics. It means no more and no less than that we must be willing to question the adequacy of all sorts of time-honoured mathematical traditions.

The reason why the mathematical challenge of programming seems to be so radically new lies in the combination of the following circumstances.
(i) Programming is a problem area without effective "intuitive support". With the latter we mean what the "picture" is for the geometer; for the programmer the "intuitive support" would be the class of possible computational histories that correspond to a program. Experience has shown, however, that this class defies imagination: the combinatorial explosion of operational arguments makes them ineffective, and, therefore, we have to resort to non-operational arguments. As a result, it is an area in which the "intuitive support" is very meagre.

(ii) In mathematics, the absence of "intuitive support" has always been countered by the application of the formal techniques of calculi. In view of the fact that, by virtue of its mechanical interpretability, each programming notation embodies eo ipso a formal system of some sort, a systematic application of calculi (logical or other) seems, indeed, indicated. This puts programming in principle on mathematically familiar grounds: program specification and program text are treated as formulae and a calculus is applied to demonstrate that the latter satisfies the former. The crux of the matter is, however, the proviso "in principle": viewed as a formula, a program text is several orders of magnitude larger than the formulae mathematicians are wont to work with, and this has a profound consequence. Manipulative techniques that engender an amount of formal labour growing quadratically or exponentially with formula length are quite acceptable in the familiar mathematical environment, in which formulae are short; in our environment they are just no good. Hence, the effectiveness of our manipulative techniques suddenly becomes a novel and major concern.

* * *

Our long-range target is thus the design of much more effective mathematical arguments. When studying the shape of arguments and comparing alternatives, we quickly discovered that in such a study of effectiveness the subject matter of the argument was of secondary importance and that we need not restrict ourselves to examples taken from computer programming. The decision to extend our explorations to mathematical arguments in general provided the added advantage -besides offering more variety in our experiments- of giving us the opportunity to compare our proofs with traditional ones, which, unlike in the field of programming, are amply available in the literature.

In the first stage of our explorations we concentrated on the presentation of arguments. (Topics of study were, for example, the choice of an appropriate size of logical steps in an argument and the problem of whether to introduce names for elements in the argument - such as the points on the circle in the geometrical example given in the preceding.) We did so for the practical reason that comparison of alternative arguments requires a homogeneous style of presentation, and for the more significant reason that we felt -and rightly so!- that there was in general much room for improvement.

Gradually we got involved in exploring ways, other than presentation, of making arguments more concise. Exploitation of symmetries and reduction of the rôle of case analyses were the obvious techniques towards this goal. Since too large logical steps in an argument engender the danger of logical errors and often leave the reader puzzling why the steps are legitimate, we adopted as our ideal a homogeneous degree of rather fine detail. Striving for both more conciseness and fine detail, we found ourselves drawn to the use of formal techniques.

We began our experiments with combinatorial problems, which could be treated effectively by using the calculus of plain arithmetic. The predicate calculus was the next one to be used extensively. Lately, regular expressions were used calculationally in a surprising context and at great calculational advantage *). (For this purpose we needed a "regularity calculus". Parts of it we had to develop ourselves: in the literature we found no traces of such calculational use of regular expressions.)

-----

*) to be presented at a Discussion Meeting on "Mathematical Logic and Programming Languages", to be organized on 15th and 16th February, 1984, under the auspices of the Royal Society of London. The proceedings will be published in the Philosophical Transactions and also as a separate Monograph.

Concerning our use of the predicate calculus to date, several remarks are in order.

(i) In order to forge the predicate calculus into a workable tool, we had to improve on the existing notational conventions. For instance, we designed a uniform notation for so-called "continued" symmetric, associative operators -such as summation-, in order to avoid the unnecessary notational distinction between, for example, $\Sigma_{i=n}^{27}$ and $\max_{n\leq i\leq 27}$ in $\Sigma_{i=n}^{27} i^2$ and $\max_{n\leq i\leq 27} i^2$. Instead, we write $(\Sigma i : n\leq i\leq 27 : i^2)$ and $(\max i : n\leq i\leq 27 : i^2)$. The symbol "i" following Σ and max in the last two formulae indicates another innovation. In so-called "quantified" expressions like the above, we explicitly indicate, syntactically, which of the variables are the so-called "dummies", in order to avoid ambiguities such as $\max_{a\leq x} x+a^2$, which indistinguishably stands for $(\max a : a\leq x : x+a^2)$ or $(\max x : a\leq x : x+a^2)$ -two very different formulae.

During our explorations, the arising need for such notational innovations was an encouraging confirmation of our impression that we were pursuing an as yet unexplored line of research.

(ii) We have used the logical connective of equivalence at great calculational advantage in preference to the logical connective of implication, often leading to a reduction in size of the argument by a factor 2 or more. This experience could well be of great methodological significance since, on the one hand, it is this century's tradition to base mathematics on set theory, whereas, on the other hand, set theory lacks the analogue of the logical connective of equivalence.

Recent information (from J Strother Moore) casts a completely new light on the rôle of equivalence, which suggests a different explanation for the fact that equivalence is used so rarely in traditional mathematics. The use of the equivalence is tantamount to so-called "symmetric rewrite rules", which are shunned in automatic theorem proving because they allow the manipulative process to get hooked up in an endless loop —e.g. replacing A by B, then B by A, and repetatur ad infinitum—. A fear for similar dead ends may subconsciously underly the underexploitation of the equivalence in traditional mathematics.

(iii) Our experiences indicate that the predicate calculus in the form and notation that we developed for it over the last years is quite generally applicable. This, we propose to demonstrate in the next phase.

Among the recent results of our work, we mention the following two incidents:
(i) the efficiency of a computing machine (still on the drawing board) could be improved at a glance by more than 10% at the price of a very minor modification: the design violated one of our principles for a well-shaped mathematical argument;
(ii) our stress on the fact that 0 and 1 are very natural numbers, led to a significant simplification in a work on VLSI design; the simplification was obtained by identifying characters from an alphabet with one-character strings, and the empty string with the 0-element of the concatenation operator; as a result, two originally distinct functions — one on characters and one on strings — merged into one.

* * *

We expect that in the next phase design, use, and propagation of calculi will form an important part of our activities. Our main goal remains what it was, viz. the streamlining of the mathematical argument — "streamline: [...] simplify, make more efficient or better organized", Concise Oxford Dictionary —; over the last years we have learnt to appreciate calculational arguments — when available! — as effective means of streamlining, and extending their range of applicability has, therefore, become a significant subgoal.

In its wake, we hope and expect to come to grips with the problem of the design of suitable notations. (Most mathematicians agree that for many mathematical developments the availability of a good notation has been crucial; in this light it is surprising that the design of notations is not a topic explicitly addressed.) The systematic application of calculational techniques is expected to provide the necessary guidance: by virtue of their rather rigid format they will reveal more clearly the manipulative needs to which the notation should be geared.

Another thing we propose to do is to try to establish (by experimental exploration) the more general validity of a number of still tentative observations. For instance, in a number of cases we have observed that the explicit statement of what is considered "obvious" often allows the rest of the proof to be carried out in a more calculational manner. We suspect this to be the case in general.

In the mean time more macroscopic concerns are emerging. The clear delineation of the scope of all sorts of conventions is one of them ("scope" is the technical term for that part of a text to which a definition, convention, or assumption pertains); though important, it does not seem to raise profound problems. Tougher seem more structural problems such as the optimal size of scopes and the best interface between logical modules of the argument. (We seem to have collected evidence that striving for small-size scopes results in more disentangled arguments, and that the "usefulness" of theorems depends more on the simplicity of their formulations than on, say, the complexity of their proofs.)

In a next phase, when we expect to have obtained a clearer and convincing picture of "true elegance", we shall address ourselves more fundamentally to heuristics, i.e. the problem of how to construct a truly elegant argument.

* * *

Concerning the ways in which our efforts can contribute to BP as industrial enterprise we see three aspects.

(i) More effective ways of doing mathematics are of immediate relevance for BP in its capacity of employer of mathematicians and computing scientists.

(ii) Our methodological techniques —such as "separation of concerns"—, though primarily developed in the context of intricate mathematical arguments, are quite general and of wide applicability; the mathematical arguments have served as "laboratory environment" in which these techniques could be explored in vitro.

(iii) Our efforts should assist BP in its continuing transition into a company more and more involved in high technology: "elegance" —in the technical sense in which we use the term—, formerly only nice, is now absolutely essential.

For all this to bear fruit, an effective dialogue with BP staff is necessary; our first experiences in this respect have been very encouraging.