

Some questions.

Some questions are posed, that by the very fact of being put forward, reveal deep insight. Other questions display, by the very same fact of being put forward, only ignorance. In all probability, my questions fall in the latter category.

It begins with Gödel's Theorem, and here I start already with a display of ignorance. Someone --who was it?-- borrowed my copy of Kleene's "Introduction to Metamathematics." and, as a result, I have only my memory to rely upon. That would not be too bad if I had studied the book thoroughly, but I have not. That I did not study it, but only read (parts of) it, was caused by the fact that the reading of it did not give me the feeling that its contents really concerned me. But if I remember correctly it states --pardon my loose terminology!-- that no consistent set of axioms can serve as a basis for the demonstration of its own completeness. The above loose formulation may cause a professional logician to shudder --I am sorry: my memory is not any better-- what I do remember more clearly was that the result did perhaps startle me, but most certainly did not bother me. Is it, because Gödel's Theorem denied the possibility of a form of perfect understanding that had never been my ideal in the first place?

The above is brought up because over the past years I have discovered that many people are very suspicious about either the legitimacy or the adequacy of axiomatic methods. And recently I came to the conclusion that I could not understand why.

In a recent letter Tony Hoare asked me, whether I knew Bertrand Russell's verdict: "The advantages of the method of postulation are great; they are the same as the advantages of theft over honest toil." (Somehow I knew it; perhaps it was Tony himself who had thrown the quotation to me at an earlier occasion!) An (intellectual?) dishonesty characteristic for the method of postulation is more than mildly suggested. I allow for a difference in criteria for "honesty" between Russell and me --and, with due respect: such a difference would not necessarily bother me!-- ; yet the verdict intrigues me enough to raise the question "Why?".

Russell's verdict would not interest me, were it not for the fact that I hear it echoed today with respect to proposals for defining the semantics of programming languages. What is now known as "the axiomatic method" --would "postulational method" have given rise to fewer misunderstanding?-- defines the semantics of a programming language in close parallelism to its syntax: for each new way of defining the semantics of a composite object in terms of the semantics of its components a new syntactic construction is chosen. I was greatly attracted by that approach: for the type of programs that I would like to be better able to design in a controlled fashion, such postulates seemed to be most readily forged into a tool for practical program composition. I found it, for instance, much more attractive than mechanizing --or at least: developing a calculus for-- the translation of recursive programs into repetitive ones.

The usual defense for that two-step approach was that it was a way of reconciling the potentially greater efficiency of repetitive programs with the fact that "recursive solutions come most naturally". But do they? Is this a Law of Human Nature? Or are people that feel so, confusing "convenient" with "conventional" and is the feeling no more than the symptom of a mathematical

inability to cope with repetition --and "variables" as they occur in programming languages-- in its own right?

I found a confirmation in the temptation --I myself yielded to it once, in a moment of weakness!-- to define the semantics of

"while B do S od"

as that of the call "while(B, S)"

of the recursive procedure

"proc while(B, S): if B then S; while(B, S) fi corp" ,

but is not that cracking an egg with a sledgehammer? (I know that for the purpose of an analogy, it may be illuminating to regard all scalar constants as functions, defined on the same one-point domain, whose argument, therefore, can remain anonymous, but must we always regard constants as such functions? That, I am afraid, would be a similar sledgehammer. And, if I may raise the question, is the use of sledgehammers to crack eggs honest, in as far as the need for the sledgehammer is suggested?)

Other proposals for defining programming language semantics have been developed, and superiority over the axiomatic method has been claimed on account of an argument, that I do not understand. The argument is that in such alternatives, the axioms of the axiomatic method need not be "postulated" but can be proved as theorems. Fine and interesting. But what is the gain? Is it unfair to invert that argument, and to point out the superiority of the axiomatic method on account of the fact that there you don't need to prove them as theorems, as just postulating them suffices? If those few axioms in the one scheme and theorems in the other are the only interface through which the theory has its impact upon the programming activity --and let us assume this for the sake of the argument to be the case--, is it then not more efficient just to postulate that interface, ignoring the elaborate fabric that could be woven underneath them? Here we are clearly back at Russell's verdict. Is the method of postulation unfair, or perhaps deadly dangerous?

One possible attack on the method of postulation is "But how do you know that your postulates are true, if you have not proved them?", but that question only makes sense if we regard the postulates as stating properties of something already defined otherwise, and the question of "truth" becomes irrelevant when we regard the postulates as "interface specifications". A possibly more valid doubt may be raised by the question "How do you know that your axioms are consistent?".

My inclination --but am I perhaps awfully naive?-- is to say "Well, I am pretty sure that they are consistent. And if they are not, well that is my risk! The worst that I could have done is talking about an empty universe, and, although fruitless, that cannot have done much harm either." My impression is that, when logicians talk about "interpreting" or "providing a model" --as soon as they start doing that, I always get confused-- they are trying to demonstrate(?) the consistency of a set of axioms by showing the existence(?) of a universe to which they apply. This they do in terms of another formal system, about the consistency of which they are more confident? Is this a fair description of what logicians are doing? If so, it strikes me as an utterly respectable way of trying to raise a confidence level, the relevance of which, however, seems closely related to our having reasons for being not too confident about the first system, and the effectiveness of which seems closely related to our sympathy with the second system.

If the point of "model making" is to give an "existence proof" I think that I can understand the purpose of the exercise. I hesitate --to put it very mildly-- when the semantics of the first system is identified with --or: defined as-- the properties of the model, because besides --hopefully!-- displaying the intended properties, the model will, in general, have many other properties as well, some of them already known, some of them, perhaps, not discovered yet. And that seems a most unattractive way of defining an interface!

And now we are back at our old dilemma. Either we take by definition all properties of the model as relevant, or we specify in one way or another which of its properties are the relevant ones. In the first case we have failed to introduce in the name of "divide et impera" an interface that would allow us to divide and rule and the study how we could build upon the (only implicitly defined) interface seems bound to deteriorate in a study of the model itself; in the second case we are again close to the axiomatic method....

In short: I am convinced of "the great advantages" mentioned by Russell, but why "theft"? Who is being robbed of what? Or is it "dishonest" to create a system in which questions are void that otherwise could be solved by "honest toil"?

Even if I ignore the moral judgement implied by Russell's verdict, I find it difficult to swallow. I judge a theory not only by its intrinsic beauty, but also by which problems it evokes, which efforts it attracts, how much is needed for it, which problems it solves and how efficiently it does so, etc. And as far as I can see, the method of postulation, when applied to programming language semantics, scores rather high in most of these various aspects. Am I blind to some philosophical questions --in very much the same way as some other people are tone-deaf or colour-blind-- or come Russell's verdict and its echos from an intellectual climate in which pure science was fairly divorced from applied science, let alone from engineering?

Or, to put it in another way: if the traditional automata theory tends to make us insensitive to the role interfaces could and should play in coping with complex designs, should it then (continue to) occupy a central position in computing science curricula?

17th November 1974  
Burroughs  
Plataanstraat 5  
NUENEN - 4565  
The Netherlands

prof.dr.Edsger W.Dijkstra  
Burroughs Research Fellow