

## Turing's Hidden Legacy: Towards a Metaphysics of Computing

Keith Douglas

Our story begins in the late 19th century with the beginning of the renaissance in logic. Frege, Schroeder and Peirce have begun work that would change the world of mathematics and beyond forever. Also around this time, as we all know, began an increasing awareness of the importance of validating our mathematical practices. Within 40 years after Frege and his peers, we had heard Hilbert's call to investigate what could and could not be done within given mathematical systems. In 1930, we had Gödel's landmark paper on the completeness theorem. Shortly thereafter was his incompleteness theorem, and we had what Douglas Hofstadter was to call much later something approaching a programming language. Hofstadter went so far as to claim that Gödel had almost invented LISP. But this is curious. Here we have a result in pure mathematics - or so it seemed - that appeared to some to be concerned directly with artifacts. Perhaps in some philosophies of mathematics this is not so bizarre; we shall meet this view again. For in 1936, as we know, came Turing's paper on the Entscheidungsproblem. It bears reminding ourselves what exactly was involved in this problem in apparently pure mathematics, for here again, a strange "concrete" application was to apparently follow.

Again this problem was in the metatheory of logic. Hilbert had asked the world's mathematicians whether there was an effective procedure for determining the validity of a first order formula. In the propositional logic case - as we would say now - the analogous problem had been solved through the method of truth tables, developed by Post in 1921. Remarkably, Hilbert had the prescience to realize that the problem may in fact have no solution. In fact, as we may all remember, this is exactly what Turing was able to show in the epochal 1936 paper.

How was Turing's solution to this important problem accomplished? The usual story is that he reformulated Hilbert's question precisely and proceeded to use this exactification to great advantage. I do not dispute the usual story. What I do want to explore in the remaining portion of this talk is what exactly Turing exactified.

The famous "Turing Machine" is the result of this exactification. This formulation concerns itself with what an ideal agent of some kind is able to do given certain resources. This agent in some ways embodies the formalist dreams; he is able to make marks on paper, and read others off. He keeps a state of mind which evolves according to what is read and his previous such states. These days, computability theory textbooks (as the subject that grew around results such as Turing's and Gödel's came to be called) leave the person out of the picture, and concentrate on a device or mechanism capable of performing the operations and maintaining state as Turing's idealized clerks were able to do so. In either

case, we have something strange. The Turing Machine formalism appears to have broken out of pure mathematics, and instead has started talking about clerks or machines and their functioning.

Perhaps some of you will tell me that I am reading too much into the "prose surrounding the variables" as some like to say. They may be right. After all, one can put Turing's entire argument in contemporary recursion theory, and not worry about the didactic remarks concerning states, and processes, and recognition, and all those ...

Well, it is about time I say it:

### metaphysical

concepts. Of that more later. For the moment I'd like to continue to explore the idea that I've "missed the point" or been too mathematically illiterate or whatever.

I have two reasons for not sharing the view of those who would consign what I have labeled the metaphysics (and related concepts which are not exactly mathematical) simply didactical or the like.

The first, and probably most important reason concerns what was to grow out of Turing's paper. As you are all no doubt aware, within 10-15 years of Turing's paper there were what came to be known as stored program computers. These machines are said to be a direct outgrowth of the work of Turing and his contemporaries in areas like the decision problem. Yes, yes, there were more "concrete" things done, by Turing himself, by von Neumann and so on. But we all know that these machines are subject to the limitations Turing described in his paper! People discuss the so called "halting problem" in connection with real machines, and real processes going on in those machines. Surely the halting problem is not of interest in pure mathematics alone because of this. But if Turing's paper were just about mathematics, and not about the (non-Platonic, for those of you might think the paper is about Plato's forms) world, how would it apply to things like my Power Macintosh, a Playstation 2 and so many other devices? This result would be unheard of. Even Newton's earthshattering Principia makes assumptions (correspondence rules, whatever you wish to call them) that tie symbols and their operations to the world. We can illustrate this quickly:

$$\dot{\vec{F}} = m\dot{\vec{a}}; \quad m = 1 \text{ kg}; \quad \dot{\vec{F}} = 1 \text{ N north}; \quad \text{hence} \quad \frac{\dot{\vec{F}}}{m} = \dot{\vec{a}} = 1 \text{ m/s}^2 \text{ north}.$$

This elementary calculation tells us something about the world - namely that the object with mass 1 kilogram will accelerate 1 m/s<sup>2</sup> northward - because we have interpreted its variables in factual terms. By contrast, what is the following about?

$$qw = osu \quad ; \quad \text{hence} \quad \frac{qw}{os} = u.$$

This is a purely formal manipulation; it does not tell us anything at all, even if we give values for  $q$ ,  $w$ ,  $o$  and  $s$  and use them to solve for  $u$  as I have done. On the other hand, if I tell you that  $q$  represents a pressure in atmospheres,  $w$  a volume in liters,  $o$  is the amount of a gas in moles,  $s$  a constant with units  $L \text{ atm} / \text{mol K}$ , and  $u$  a temperature in Kelvin, then you can use the same formal operations to deduce a temperature of an ideal gas from the other values.

So it would be very strange indeed if Turing happened to prove a result that had nothing directly to do with states and processes and things and so forth as it were by accident.

The other argument for not ignoring what I have called the metaphysics is of course to simply take what he says at face value. This of course is a very messy situation. I am perfectly in agreement with those who would remind us that very often the prose in a mathematics paper should not be taken in the same way as the mathematics proper. But in this case, the first reason already pulls us in that direction. Of course this means that this second reason is parasitic on the first.

But I have a third reason, which will come clear when I explain (finally) why I am calling some of Turing's work metaphysical. The reason for this is actually because reading the work as a piece of exact metaphysics is exactly the way I am going to suggest the paper should be read.

Turing does not give many details about the "machines" he describes. He does not talk about cellular proteins, or doubly ionized calcium ions being used as neurotransmitters. Nor does he discuss making automata of the sorts that fascinated Descartes so many years before. Cogs of steel are missing; and of course we cannot fault him for not discussing the transistor, then still 11 years in the future. But why could he not have at least talked about these things in general terms if that's what he was on about?

But that's exactly what metaphysics (in my understanding) is about! Implementation details are not important so much as the general categories (a la Aristotle or Kant, etc.) involved. It is not surprising then that a paper ostensibly in pure mathematics is actually also a paper in metaphysics. Hilbert's problem asked about a certain mathematical problem. Well, problem solving does not take place "in mathland" (so to speak) but outside it - even a Platonist must admit that humans do mathematics in some way or other. So is it not at all surprising that a work about mathematics is not purely mathematical in character?

"But," someone might protest, "Turing's work uses sophisticated

recursion [or computability] theory, a precise understanding of logic, and much else, just as Gödel's earlier papers required knowledge of number theory, etc." Granted, and I will make more remarks on Gödel's paper in a bit. But Newton's work on universal gravitation, for all its mathematical sophistication, is not about the mathematics. It could not have taken humanity to the moon otherwise. The prejudice here seems to be that metaphysics cannot be done exactly. But why the heck not? Mathematics is rich in tools for dealing with the general and so on. It stands to reason that one could therefore use mathematics in the aid of metaphysics, particularly when one's concern was over a (meta-)mathematical problem.

But Turing's legacy (as I am telling it) is not merely as an exact metaphysician, though that would add to his honours enough by itself - but one who left us a new field of metaphysics, or at least a new area to be concerned with as metaphysicians. As one might infer from the title of this talk and the theme of our conference, this area is in the metaphysics of computation. To wrap up this first part of the talk, I will simply point to several key terms within Turing's original paper and contemporary presentations of the "Turing Machine" that are metaphysical in flavour.

A word on the meaning of metaphysics first, since this term is far from clear itself. For the purposes of this talk, metaphysics is the most general study of reality, concerning itself with the basic categories to which "stuff" falls under. These categories are found in all scientific fields. This means of proceeding in metaphysics may result in that Turing's legacy affects our understanding of metaphysics itself, not just create a new area of interest within it. But this is for another time.

The terms I'd like to draw the audience's attention to within Turing's paper are (I take it for granted that philosophy of mind is in part an area of metaphysics): configuration, aware, symbol, memory, state of mind, simple (in the context of "simple operations"), recognition, change, and perhaps others. Some of these border on epistemological notions as well; I do not need at this point to distinguish between metaphysics and epistemology. Of course, it may be that Turing's paper makes the case for enlarging this list to include the notions of computation and so forth themselves. The analogous list is all the more salient in contemporary discussions of automata, Turing machines and so forth. For example, such are described in Kozen's 1997 textbook *Automata and Computability* using such terms as: determinism (which of course has meanings in computer science and in metaphysics, but ones that are perhaps portable "to the other side", so to speak), state, transition, spontaneously, pattern, etc.

This leads me to my final area of discussion. I would like to finish this talk by discussing several future directions and areas for which the recognition that the nature of computation is

partially a metaphysical one. One is the curious relation between Gödel's incompleteness theorems and the Halting problem. If the latter concerns itself with the nature of certain kinds of things and their processes, why is it so strongly related to a result in (apparently) pure mathematics? Related are such questions as: why is it said that type zero grammars are equivalent to Turing machines? Well, they describe the same languages in the sense of the theory of languages. This is fine as far as it goes, but how does this work when it comes to "embodying" the Turing machine and the grammar. An embodied grammar (say like the one Chomskians think is in the human brain) seems to be some sort of complicated process or collection of processes. An embodied Turing machine, by contrast is a thing. Of course, the thing undergoes certain processes as it (e.g.) recognizes strings in a language. But that is not quite the same sort of situation. Of course, one could then compare the putative process in our brain to the processes in the (say) Power Macintosh on my desk. But we stray away from the original idea.

My final suggestion is one about the decades old debate about computationalism and intelligence. With Turing's paper looked at as metaphysics, we now can proceed to tweak it. Turn the categorical knobs, as it were, and see which sorts of processes and so forth result. It may be that one can turn enough knobs and get results so far from what we intuitively call computations that perhaps we can find that (for instance) Searle was right in one respect or other. Or maybe we will vindicate Block's insight that to be a computer is to involve a certain kind of counterfactual causal power. But much work has to be done to examine all these knobs - I for one am particularly interested in the continuous/discrete issue that Turing himself deals with - but there are others - and see what sorts of results come out. In some ways, we've been doing this already, and the Church-Turing thesis is an expression of what we've found. But again we have our questions to ask: how is (e.g.) the lambda calculus "machinelike" - as it would have to be if it is metaphysically equivalent to the Turing machine formalism. I do not intend to have answers any of these questions today, but only to raise a different way of looking at problems for the future. Sometime later I hope to talk again on some specific results and in greater detail about some of the above problems, but for now, a *discours sur la methode* shall have to do. Thank you.