Why Study Computability?

A *decision procedure* for a set S is an algorithm – a fully mechanical computation procedure – that correctly answers questions of the form "Is so-and-so in S?" A *proof procedure* for S is the positive half of a decision procedure. If *a* is in S, then there is a "proof" that *a* is in S, and the proof can be found by diligent seaching; if *a* isn't in S, then there is no such proof. Moreover, the processes of searching for a proof and recognizing a proof when you find one can be carried out in a purely mechanical fashion.

In Logic I, we learned a couple of decision procedures for the set of valid sentences of the sentential calculus, the method of truth tables and the search-for-counterexamples method. We also learned a system of rules that enable us to derive any valid sentence of the predicate calculus. These rules constitute a proof procedure for the set of valid sentences. We did not, however, learn a decision procedure. If we try to prove ϕ and we don't succeed, there will be no stage at which we can be sure that the reason we haven't succeeded is that ϕ isn't provable, rather than we haven't looked long enough. The explanation why we haven't found a decision procedure for the set of valid sentences of the predicate calculus shows more than merely that we haven't been clever enough to devise one. There is, in fact, no decision procedure there to be had. This was demonstrated by Alonzo Church and Alan Turing:

> **Church-Turing Theorem.** There is no decision procedure for the set of valid sentences of the predicate calculus.

What a remarkable result! To point out that there is no known procedure for testing validity in the predicate calculus is no big deal; it just means that finding a test for validity is an unsolved problem. But the Church-Turing theorem tells us much more than this. It tells us that no time, ever, even in the distant future, will anyone discover a decision procedure for the validity in the predicate calculus.

The Church-Turing theorem is a result of an extensive investigation, carried out during the 1930s by Turing, Gödel, Kleene, Church, and others, of the question, “When is there an algorithm for solving a particular mathematical problem?” That is, when is there a system of explicit rules, which can be followed in a perfectly mechanical way and which will always give a correct answer to a certain category of mathematical question? The investigation was quite fruitful. For one thing, the investigation preceded the development of the electronic computer and facilitated its development. Turing developed a highly simplified and schematic model of what we do when we calculate with pencil and paper, and much of the early work in developing digital computers was aimed at implementing Turing’s model electronically.

It is not surprising that these investigations have been of interest of mathematicians and computer scientists. For one thing, before these investigations, one had no way of showing an undecidable problem to be undecidable. Thus if one were working on a problem of the form “Find an algorithm for so-and-so” – and a great many important mathematical problems take this form – and if there were, in fact, no such algorithm, then one might spend one’s entire career in a futile search for an algorithm that isn’t there. This could still happen, even after the advent of the theory of computability, but it’s made less likely by the advent of techniques that enable you to work on the problem from both ends. Try to find an algorithm, and if that doesn’t work, try to prove that there is no such algorithm.

An example is Hilbert's tenth problem, the tenth problem on a famous list of the world's most prominent unsolved mathematical problems, promulgated by David Hilbert in 1900. The problem was to find when a polynomial (in several variables) with integer coefficients has an integer solution. It is clear that there is a proof procedure for the set of polynomials with integer coefficients and integer solutions. You just test possible solutions one by one by multiplying and adding. It is not so clear whether there is a method for identifying those problems that don't have integer solutions. Many bright people worked many hours on this problem, and they would be working on it still had not Yuri Matijasevic in 1970 applied the methods of computability theory (or *recursion theory*, as it's called) to prove that there is no such algorithm.

Let me now give some reasons why recursion theory ought also to be of interest to philosophers.

Recursion theory is concerned with problems that can be solved by a computing machine. What counts as a computing machine? It doesn't matter what they thing is made of. A computing machine could be made of Tinkertoys®, of embroidery thread, or of animal flesh. What matters is how the thing is organized. The device has to make its way from inputs to outputs in a sequence of purely mechanical steps; an IBM PC that acts under demonic influences will not count as a computing machine. The philosophically interesting thesis is that we ourselves ought to be counted as computing machines. Our inputs are sensory stimuli and our outputs are behaviors. The chemical and electrical processes by which our behaviors are produced are simple natural processes not different in kind from the processes that occur on a silicon chip. Thus any problem we can solve is going to be decidable.[1]

The content of this observation is not entirely abstract. Computer models of the mind have become prominent in recent psychology and philosophy of mind.

Recursion theory is concerned with problems that can be solved by following a rule or a system of rules. Linguistic conventions, in particular, are rules for the use of a language, and so human language is the sort of rule-governed behavior to which recursion theory applies. Thus, if, as seems likely, an English-speaking child learns a system of rules that enable her to tell which strings of English words are English sentences, then the set of English sentences has to be a decidable set. This observation puts nontrivial constraints upon what the grammar of a natural language can look like.

As Wittgenstein never tired of pointing out, when we learn the meaning of a word, we learn how to use the word.[2] That is, we learn a rule that governs the word's use.

---

1 It appears likely that our stimuli don't uniquely determine our behaviors; instead, there are elements of chance involved. But this makes no difference, since it turns out that the same problems can be solved by nondeterministic computing machines as can be solved by deterministic machines.

2 It may be that, when we learn the word "duck," we learn to associate the word this the universal canardity. But even if that's true as a matter of metaphysics, it has no value in explaining how we learned the word, for our parents didn't teach us the word by repeating the word "duck" while pointing to the universal.

So a theory that gives us the limitation on what we can accomplish by following a rule also gives us limitations on the structure of the meanings of words.

In particular, when we are learning arithmetic, what we are learning, presumably, are the rules for the use of numerical language. The natural alternative, which is that the child learns to use "three" the way she learns to use "Mama" or "duck," by learning to recognize three when she sees it, seems preposterous. But if learning arithmetic really consists in learning rules of language, then, it would seem, we ought to be able to state what the rules are, then to say that a sentence is recognizable as a truth of arithmetic if and only if it is can be produces by following the rules. But, in fact, we cannot do this. A deep theorem of Kurt Gödel, which we shall discuss presently, shows that, for any system of rules, either there si a false arithmetical sentence that is a consequence of the rules, or else there is a true arithmetical sentence that is not a consequence of the rules. Worse: for any system of rule we can specify for generating truths of arithmetic, there will be an arithmetical sentence that isn't a consequence of the rules that we can recognize as true.

Gödel's theorem places important limits upon the power of the axiomatic method. Before Gödel's theorem, it was generally supposed that hoe we acquired mathematical knowledge was by drawing out the consequences of mathematical axioms. The axioms themselves were traditionally regarded as self-evident truths; after the advent of non-Euclidean geometry, it became common to suppose that the mathematician's job was simply to develop the consequences of axiom systems, without concerning themselves with whether the axiom systems are true. Gödel's theorem shows that that traditional account can't be all there is to the epistemology of mathematics. The precise details of how we formulate the axioms doesn't matter. However we formulate the axioms of number theory, as long as we can see that they're all true, there will be further sentences that we can recognize as true even though they aren't derivable from the axioms.

Gödel's theorem takes on a particular urgency when we connect it to the question, "How do mathematical terms get their meanings?" It appears that the answer has to be something like, "The meanings of the terms are established by the mathematical theory." That can't be the whole answer, because we need to take account of the use of mathematical terms in such activities as counting and measuring, but these activities don't go very far toward pinning down the meanings of the terms. What emphatically we don't have for the names of numbers is an analogue for what we have for the names of dogs: our causal connection with the dog Fido helps to pin down the meaning of the name "Fido." In the absence of causal connections, there is precious little left, other than the theory, to pin down the meanings of the terms. Yet the theory can't pin down the meanings, since there are sentences that, as far as the theory is concerned might be true and might be false, even though it follows from the biconditionals

$\ulcorner\phi\urcorner$ is true if and only if ϕ

$\ulcorner\phi\urcorner$ is false if and only if $\neg\phi$

that every sentence is either true or false. This discrepancy has driven some philosophers to mathematical skepticism.

A close relative of Gödel's result is a theorem of Tarski's to the effect that, if our commonsense understanding of the notion of truth is correct, then it will not be possible to develop a theory of truth for a language within the language itself. Instead, the theory of truth for a language 𝔏 must be developed within a *metalanguage* which is essentially richer than 𝔏 in expressive power. The philosophical consequences of this result are staggering. It implies that we cannot give a theory of truth for English (or for any natural language) for we don't have any metalanguage richer than English in expressive power. Moreover, the result entails that it is not possible to obtain a unified science that comprehends nature, human language, and human thought as a unified whole, for the language we use when we talk about language lies outside the reach of linguistic inquiry.

Another, quite different, reason why philosophers ought to be interested in recursion theory is this: since Plato, philosophers have been trying to give *analyses*, in which the meaning of a difficult or troublesome term is explained in terms of other terms which are simple, clearer, and better understood. For example, to fill in the blank in

> It is right for S to do A iff ___________________,

where the blank is filled in by something from outside the circle of moral terms.[3] The right sort of thing would be:

> It is right for S to do A iff, of the possible acts available to S, A is most conducive to the greatest happiness for the greatest number,

provided that "happiness" is understood in such a way that it is not infused with moral value. The biconditional you get by filling in the blank needs to be a conceptual truth. If we give a

---

[3] If we explain the meaning of "right" in terms of other moral terms, we may have done something quite valuable and helpful, but we will not have done what has traditionally been regarded as an analysis. An analysis of "right" would explain the meaning of "right" to someone who doen't have any moral vocabulary at all.

characterization of the class of right acts whose correctness is merely accidental, or even if we give a characterization is a matter of physical law, we would not say that we have given an analysis of what it is for an act to be right.

The program of trying to understand a difficult concept by presenting a conceptual analysis is a venerable part of our philosophical heritage, but it is a program which has scarcely ever succeeded. One such example is the analysis of decidability that we get from recursion theory. Recursion theory gives us a biconditional of the form

There is an algorithm for testing membership in S if and only if ____________,

where the blank is filled in with a statement form pure mathematics. Thus on the left there is a statement about procedures and capacities, and on the right there is a statement that has nothing to do with procedures and capacities, a statement you would be able to understand even if you were a member of a race of superbeings that could recognize mathematical truths by direct intuition and had no inkling what computations or algorithms are.[4] Moreover, the biconditional, if it is true at all, is a conceptual truth. So we have a successful analysis.

If we are interested in giving conceptual analyses, it is a good idea to look carefully at the few examples we already have of successful analyses. One reason this is so is that it is usually a good strategy to emulate success. See what methods were employed to give our successful analysis of decidability in order to see whether these same methods can be applied to give analyses of other concepts. Even if this doesn't give us the analysis we seek, it may still give us some useful information. Understanding why the concept of rightness doesn't yield to the same

---

[4] According to Thomas Aquinas, that the way it is with angels.

analytical techniques that gave us a successful analysis of decidability will point out to us some of the obstacles that lie in the way of a successful analysis of the right.

Incidentally, the only other example I know of a successful conceptual analysis (apart from trivialities like "Vixen are female foxes") is the ϵ-δ analysis of the concepts of limit and continuity.