WHAT IS...

Turing Reducibility?

Martin Davis

Intuitively, to say that something is *computable* means that there is an algorithm for computing it. Computability theory makes this concept precise. So, in particular, we are enabled to define with full rigor what it means to say that a function defined on the natural numbers and with natural number values is *computable*. Likewise a set of natural numbers is *computable* if its characteristic function (defined to be 1 for members and 0 for non-members) is computable. Having such definitions makes it possible to prove that certain objects are not computable. A fundamental result is that there is a computable function whose range (the set of all values that it assumes) is not computable. Sets that are the range of a computable function are called *recursively enumerable*. (The empty set is also considered to be recursively enumerable.) The fact that there is a recursively enumerable set that is not computable is a special case of a more general result that will be explained later in this article. It is the use of this fact that has made it possible to prove that a number of important mathematical problems are unsolvable, that algorithms that mathematicians had been seeking simply do not exist. Among these problems are Hilbert's 10th Problem (to decide whether a given Diophantine equation has solutions), the word problem for groups (to decide whether a given product of generators and their inverses is the identity element of a group defined by a finite set of equations between such products), and the homeomorphy problem (to decide whether the topological spaces defined by a given pair of simplicial complexes are homeomorphic).

Martin Davis is professor emeritus of mathematics and computer science, New York University, and is a visiting scholar in mathematics at the University of California, Berkeley. His email address is `martin@eipye.com`.

The concept of *Turing reducibility* has to do with the question: can one non-computable set be more non-computable than another? In a rather incidental aside to the main topic of Alan Turing's doctoral dissertation (the subject of Soloman Feferman's article in this issue of the *Notices*), he introduced the idea of a computation with respect to an oracle. An *oracle* for a particular set of natural numbers may be visualized as a "black box" that will correctly answer questions about whether specific numbers belong to that set. We can then imagine an oracle algorithm whose operations can be interrupted to query such an oracle with its further progress dependent on the reply obtained. Then for sets A, B of natural numbers, A is said to be *Turing reducible* to B if there is an oracle algorithm for testing membership in A having full recourse to an oracle for B. The notation used is: $A \leq_t B$. Of course, if B is itself a computable set, then nothing new happens; in such a case $A \leq_t B$ just means that A is computable. But if B is non-computable, then interesting things happen.

As the notation suggests, Turing reducibility is a partial order. If sets A, B are each Turing reducible to the other, they are said to be Turing equivalent, written $A \equiv_t B$. And if A is Turing reducible to B but not conversely we write $A <_t B$. By considering all oracle algorithms having access to an oracle for a particular set A, one can construct a new set A' that contains all the information concerning membership in any set Turing reducible to A. (The construction is analogous to that of a "universal" Turing machine.) The operation $'$ is called the *jump* because it can easily be proved (by using a Cantor-style diagonal argument) that $A <_t A'$. Setting $A =$

$\varnothing$, the set $\varnothing'$ provides an example of a recursively enumerable set that is not computable. Iterating the jump operation, one obtains the sequence of more and more unsolvable problems, $\varnothing', \varnothing'', \ldots$.

The relation of Turing equivalence is, naturally, an equivalence relation, and the equivalence classes are called *Turing degrees*. One speaks of the degree of a set of natural numbers to mean the equivalence class to which it belongs. The degrees inherit the partial order, and the jump operation is a degree invariant. All of the computable sets form a single degree written $\mathbf{0}$ which is at the bottom of the partial order. That is, $\mathbf{0} \leq \mathbf{a}$ for every Turing degree $\mathbf{a}$. Also, $\mathbf{a} < \mathbf{a}'$. Are there degrees between $\mathbf{a}$ and $\mathbf{a}'$? Kleene and Post were able to show that the ordering of degrees is complicated and messy. For example, for any degree $\mathbf{a}$, they showed how to obtain degrees $\mathbf{b}, \mathbf{c}$ such that $\mathbf{a} < \mathbf{b} < \mathbf{a}'$, $\mathbf{a} < \mathbf{c} < \mathbf{a}'$, but $\mathbf{b}$ and $\mathbf{c}$ are incomparable: neither is less than the other. They also found densely ordered degrees; that is, they showed that for a given degree $\mathbf{a}$, an infinite linearly ordered set $\mathcal{W}$ of degrees between $\mathbf{a}$ and $\mathbf{a}'$ can be found such that if $\mathbf{b}, \mathbf{c} \in \mathcal{W}$, there is a degree $\mathbf{d} \in \mathcal{W}$ between $\mathbf{b}$ and $\mathbf{c}$.

The degree of every recursively enumerable set is $\leq \mathbf{0}'$. There is a sense in which the typical mathematical problems that have been proved to be unsolvable are of degree $\mathbf{0}'$. For example, if we enumerate all polynomial Diophantine equations with integer coefficients in some standard way, the degree of the set of natural numbers n such that the nth equation has a solution in natural numbers is exactly $\mathbf{0}'$. So we can say that Hilbert's tenth problem is not only unsolvable but has exactly the degree of unsolvability $\mathbf{0}'$. A degree is called recursively enumerable if it contains a recursively enumerable set. $\mathbf{0}$ and $\mathbf{0}'$ are both recursively enumerable degrees with $\mathbf{0} < \mathbf{0}'$. In a classic paper Post raised the question of the existence of other recursively enumerable degrees, and this became known as Post's Problem. It required a new combinatorial technique, known as the priority method, to settle the question. The idea was to list a countable infinity of requirements that the desired objects would need to satisfy, and to mediate among conflicting requirements in a manner that would result in all of them being ultimately satisfied. By using this technique, it was shown that not only are there recursively enumerable degrees strictly between $\mathbf{0}$ and $\mathbf{0}'$, but indeed that pairs of such degrees can be found that are mutually incomparable. The use and refinement of the priority method has made it possible to prove a number of striking facts about the recursively enumerable degrees. For example, the Sacks Density Theorem states that for given recursively enumerable degrees $\mathbf{a} < \mathbf{b}$, there is a recursively enumerable degree $\mathbf{c}$ such that $\mathbf{a} < \mathbf{c} < \mathbf{b}$.

References

[1] Hartley Rogers, *Theory of recursive functions and effective computability,* New York: McGraw-Hill 1967.

[2] Joseph Shoenfield, *Recursion theory*, Lecture Notes in Logic 1, Berlin, New York: Springer-Verlag, 1993.

[3] Robert Soare, *Recursively enumerable sets and degrees: a study of computable functions and computably generated sets,* Berlin, New York: Springer-Verlag, 1987.