SOME COMMENTS ON THE DIAGONAL METHOD

David McGoveran
Alternative Technologies
15900 Bear Creek Road
Boulder Creek, California 95006

By a suitable choice of a system from those systems which should be susceptible to the diagonal method, it is possible to show that the diagonal method is a flawed method of argument. The method has been applied in proofs of Godel's Incompleteness Theorem among others and yet, by demonstrating that it gives wrong results in one case, we show that it is not trustworthy in the mode commonly used for decision and halting problems by recursion theorists.

We take as a starting point the assumption that all effectively computable procedures are representable in terms of a Turing machine. Consider all strings of a given system S that have been thus far constructed. Assume that there is a Turing machine that serves to generate those strings. (Each instruction of a Turing machine can be given by a quadruple of the form internal input state n, input operation a, output operation b, internal output state m.) For each string, there will be a set of instructions (a function) which serves to cause the Turing machine to generate the string. We can order these sets of instructions by ordering the output strings. Namely, interpret the string as a binary number. Order the list of output strings by this index in increasing order. Now, create a catalog index (Godel numbering) which simply enumerates the strings - i.e. label the first string "1", the second string "2", etc.. Let each such label represent the set of instructions necessary to generate the given string. All that we have done is effectively computable. We now have an enumerable list of functions which generate the system up to output string N.

Now assume that there exists a decision procedure for the system S. It follows that there exists a function f which takes as input the index for an entry in the set of instructions and which generates, as output, the string which would be produced by the set of instructions. The decision procedure is but one step beyond this - namely, from the function f, we can construct a decision procedure h. The function h takes as input a given string and outputs a 1 if the string is one that will be generated from the current catalog of functions and outputs a 0 otherwise. That this is possible given the function f is clear: h uses f recursively to generate the strings represented by the catalog, comparing each string to the input string until it finds a match or until it terminates - i.e. halts.

Let Q_N be the (N+1)st derivation or set of instructions.

Let f_N (I)

be the function corresponding to Q_N. However the

function f does not exist. Assume there is such a recursive f. It can be used to define a new partial function g as follows:

$$g(x) = f_x(x) + 1 \qquad \text{(II)}$$

Evidently, we have an algorithm for computing g: namely, to get h(x) for a given x, generate the list of derivations out of Q_x

then employ Q_x to compute $f_x(x)$, then add one. On the other hand

g cannot be partial recursive. If it were, we would have h = g_N for some N. But then we would have

$$g_N(N) = h(N) = g_N(N) + 1, \qquad \text{(III)}$$

a contradiction. Since g_N is not

partial recursive, and the operation of adding one to a partial recursive function is a partial recursive function, it follows that f is not partial recursive - not effectively computable as was assumed.

Since it would be difficult to argue that the constructiveness (effective computability) of the system S is not itself a valid assumption (as assumed), it would seem to follow that the decision procedure does not exist. However, notice that we have not provided an instance of a system S nor an interpretation of the system S. We now do so. Let the system S consist of the symbols s and ' and the following rule of inference:

for all x, x ∈ S ---> x' ∈ S

This system shall have two isomorphic interpretations:

1) function interpretation:
   s is the zero symbol
   ' is the operation of adding one (successor function)

2) sentential interpretation:
   s is the initial string of S ("This is a string of S")
   ' is the operation of enclosing in quotes and concatenating "is a string of S".

Following Godel, both of these interpretations are partial recursive. It follows that formulas (I) and (II) are, by interpretation of S, functions of S since they involve only the operations of adding one to a given function of S. Therefore, the contradiction (III) must lie in other than the assumption that f is partial recursive and thus the diagonal method fails.

The reason for the failure is now clear. We generated a system S for our enumerable list up to string N. The diagonal method, however, argues by induction beyond N (formula (II) above) to a completed list where N is infinite. Hence, the diagonal method is neither constructive nor a valid method of proof in the context of decision problems.

Rogers, Hartley, Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, NY, c. 1967, especially pages 10-49