Turing and Post on Computability

Selections appear in "Computability: Computable Functions, Logic, and the Foundations of Mathematics" by Epstein and Carnielli published by Wadsworth & Brooks/Cole Publishers 1989

A. Turing on Computability (Optional)

One of the first analyses of the notion of computability, and certainly the most influential, is due to Turing.

Alan M. Turing, from "On Computable Numbers, with an Application to the Entscheidungsproblem," 1936

The "computable" numbers may be described briefly as the real numbers whose expressions as a decimal are calculable by finite means. . . . According to my definition, a number is computable if its decimal can be written down by a machine. p. 116

[Turing then gives his formal definitions and in particular says that for a real number or function on the natural numbers to be computable it must be computable by a machine that gives an output for every input.]

No attempt has yet been made to show that the "computable" numbers include all numbers which would naturally be regarded as computable. All arguments which can be given are bound to be, fundamentally, appeals to intuition, and for this reason rather unsatisfactory mathematically. The real question at issue is "What are the possible processes which can be carried out in computing a number?"

The arguments which I shall use are of three kinds.

a. A direct appeal to intuition.

b. A proof of the equivalence of two definitions (in case the new definition has a greater intuitive appeal). [In an appendix to the paper Turing proves that a function is calculable by his definition if and only if it is one of Church's effectively calculable functions.]

c. Giving examples of large classes of numbers which are computable. ...

[I.] Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. I shall also suppose that the number of symbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent. The effect of this restriction of the number of symbols is not very serious. It is always possible to use sequences of symbols in place of single symbols. Thus an Arabic numeral such as 17 or 999999999999999 is normally treated as a single symbol. Similarly in any European language words are treated as single symbols (Chinese, however, attempts to have an enumerable infinity of symbols). The differences from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, cannot be observed at one glance. This is in accordance with experience. We cannot tell at a glance whether 9999999999999999 and 999999999999999 are the same.

The behaviour of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that there is a bound B to the number of symbols or squares which the computer can observe at one moment. If he wishes to observe more, he must use successive observations. We will also suppose that the number of states of mind which need be taken into account is finite. The reasons for this are of the same character as those which restrict the number of symbols. If we admitted an infinity of states of mind, some of them will be "arbitrarily close" and will be confused. Again, the restriction is not one which seriously affects computation, since the use of more complicated states of mind can be avoided by writing more symbols on the tape.

Let us imagine the operations performed by the computer to be split up into "simple operations" which are so elementary that it is not easy to imagine them further divided. Every such operation consists of some change of the physical system if we know the sequence of symbols on the tape, which of these are observed by the computer (possibly with a special order), and the state of mind of the computer. We may suppose that in a simple operation not more than one symbol is altered. Any other changes can be split up into simple changes of this kind. The situation in regard to the squares whose symbols may be altered in this way is the same as in regard to the observed squares. We may therefore, without loss of generality, assume that the squares whose symbols are changed are always "observed" squares.

Besides these changes of symbols, the simple operations must include changes of distribution of observed squares. The new observed squares must be immediately recognisable by the computer. I think it is reasonable to suppose that they can only be squares whose distance from the closest of the immediately previously observed squares does not exceed a certain fixed amount. Let us say that each of the new observed squares is within L squares of an immediately previously observed square.

In connection with "immediate recognisability", it may be thought that there are other kinds of square which are immediately recognisable. In particular, squares marked by special symbols might be taken as immediately recognisable. Now if these squares are marked only by single symbols there can be only a finite number of them, and we should not upset our theory by adjoining these marked squares to the observed squares. If, on the other hand, they are marked by a sequence of symbols, we cannot regard the process of recognition as a simple process. This is a fundamental point and should be illustrated. In most mathematical papers the equations and theorems are numbered. Normally the numbers do not go beyond (say) 1000. It is, therefore, possible to recognise a theorem at a glance by its number. But if the paper was very long, we might reach Theorem 157767733443477; then, further on in the paper, we might find "... hence (applying Theorem 157767733443477) we have ...". In order to make sure which was the relevant theorem we should have to compare the two numbers figure by figure, possibly ticking the figures off in pencil to make sure of their not being counted twice. If in spite of this it is still thought that there are other "immediately recognisable" squares, it does not upset my contention so long as these squares can be found by some process of which my type of machine is capable. This idea is developed in [III] below.

The simple changes must therefore include:

a. Changes of the symbol on one of the observed squares.

b. Changes of one of the squares observed to another square within L squares of one of the previously observed squares.

It may be that some of these changes necessarily involve a change of state of mind. The most general single operation must therefore be taken to be one of the following:

A. A possible change (**a**) of symbol together with a possible change of state of mind.

B. A possible change (**b**) of observed squares, together with a possible change of state of mind.

The operation actually performed is determined, as has been suggested [above] by the state of mind of the computer and the observed symbols. In particular, they determine the state of mind of the computer after the operation is carried out.

We may now construct a machine to do the work of this computer. To each state of mind of the computer corresponds an "*m*-configuration" of the machine. The machine scans B squares corresponding to the B squares observed by the computer. In any move the machine can change a symbol on a scanned square or can change any one of the scanned squares to another square distant not more than L squares from one of the other scanned squares. The move which is done, and the succeeding configuration, are determined by the scanned symbol and the *m*-configuration. ...

[III] We suppose, as in [I], that the computation is carried out on a tape; but we avoid introducing the "state of mind" by considering a more physical and definite counterpart of it. It is always possible for the computer to break off from his work, to go away and forget all about it, and later to come back and go on with it. If he does this he must leave a note of instructions (written in some standard form) explaining how the work is to be continued. This note is the counterpart of the "state of mind". We will suppose that the computer works in such a desultory manner that he never does more than one step at a sitting. The note of instructions must enable him to carry out one step and write the next note. Thus the state of progress of the computation at any stage is completely determined by the note of instructions and the symbols on the tape. That is, the state of the system may be described by a single expression (sequence of symbols), consisting of the symbols on the tape followed by Δ (which we suppose not to appear elsewhere) and then by the note of instructions. This expression may be called the "state formula". We know that the state formula at any given stage is determined by the state formula before the last step was made, and we assume that the relation of these two formulae is expressible in the functional calculus [see Chapter 21 of this text]. In other words, we assume that there is an axiom **A** which expresses the rules governing the behaviour of the computer, in terms of the relation of the state formula at any stage to the state formula at the preceding stage. If this is so, we can construct a machine to write down the successive state formulae, and hence to compute the required number.

Turing, pp. 135–140

B. Emil L. Post on Computability (Optional)

Post's analysis of computability was done independently of Turing, though not of Church. It is therefore surprising how very similar it is to Turing's analysis in his paper in Chapter 9 (similarities to our formalization of Turing's ideas are not so remarkable since we've been influenced by developments since then, including Post's paper). Post, too, attempts to justify his formulation in intuitive terms. Note that, unlike Church, he does not view Church's Thesis as a *definition* but claims that if, as it turned out, the Most Amazing Fact holds, then Church's Thesis amounts to a *natural law*.

"Finite Combinatory Processes — Formulation 1" *

The present formulation should prove significant in the development of symbolic logic along the lines of Gödel's theorem on the incompleteness of

* Received October 7, 1936. The reader should compare an article by A. M. Turing, "On computable numbers," shortly forthcoming in the *Proceedings of the London Mathematical Society*. The present article, however, although bearing a later date, was written entirely independently of Turing's. *Editor* [of *The Journal of Symbolic Logic*].

symbolic logics[1] and Church's results concerning absolutely unsolvable problems.[2]

We have in mind a *general problem* consisting of a class of *specific problems.* A solution of the general problem will then be one which furnishes an answer to each specific problem.

In the following formulation of such a solution two concepts are involved: that of a *symbol space* in which the work leading from problem to answer is to be carried out,[3] and a fixed unalterable *set of directions* which will both direct operations in the symbol space and determine the order in which those directions are to be applied.

In the present formulation the symbol space is to consist of a two way infinite sequence of spaces or boxes, i.e., ordinally similar to the series of integers … , −3, −2, −1, 0, 1, 2, 3, … . The problem solver or worker is to move and work in this symbol space, being capable of being in, and operating in but one box at a time. And apart from the presence of the worker, a box is to admit of but two possible conditions, i.e., being empty or unmarked, and having a single mark in it, say a vertical stroke.

One box is to be singled out and called the starting point. We now further assume that a specific problem is to be given in symbolic form by a finite number of boxes being marked with a stroke. Likewise the answer is to be given in symbolic form by such a configuration of marked boxes. To be specific, the answer is to be the configuration of marked boxes left at the conclusion of the solving process.

The worker is assumed to be capable of performing the following primitive acts: [4]

(a) *Marking the box he is in (assumed empty),*
(b) *Erasing the mark in the box he is in (assumed marked),*
(c) *Moving to the box on his right,*
(d) *Moving to the box on his left,*
(e) *Determining whether the box he is in, is or is not marked.*

The set of directions which, be it noted, is the same for all specific problems and thus corresponds to the general problem, is to be of the following form. It is to be headed:

Start at the starting point and follow direction 1.

It is then to consist of a finite number of directions to be numbered 1, 2, 3, … n. The ith direction is then to have one of the following forms:

(A) *Perform operation* O_i [O_i = (a), (b), (c), *or* (d)] *and then follow direction* j_i,
(B) *Perform operation* (e) *and according as the answer is yes or no correspondingly follow direction* j_i' *or* j_i''.
(C) *Stop.*

Clearly but one direction need be of type C. Note also that the state of the

[1] Kurt Gödel, [1931].
[2] Alonzo Church, [1936].
[3] Symbol space, and time.
[4] As well as otherwise following the directions described below.

symbol space directly affects the process only through directions of type B.

A set of directions will be said to be *applicable* to a given general problem if in its application to each specific problem it never orders operation (a) when the box the worker is in is marked, or (b) when it is unmarked.[5] A set of directions applicable to a general problem sets up a deterministic process when applied to each specific problem. This process will terminate when and only when it comes to the direction of type (C). The set of directions will then be said to set up a *finite 1-process* in connection with the general problem if it is applicable to the problem and *if the process it determines terminates for each specific problem*. A finite 1-process associated with a general problem will be said to be a *1-solution* of the problem if the answer it thus yields for each specific problem is always correct.

We do not concern ourselves here with how the configuration of marked boxes corresponding to a specific problem, and that corresponding to its answer, symbolize the meaningful problem and answer. In fact the above assumes the specific problem to be given in symbolized form by an outside agency and, presumably, the symbolic answer likewise to be received. A more self-contained development ensues as follows. The general problem clearly consists of at most an enumerable infinity of specific problems. We need not consider the finite case. Imagine then a one-to-one correspondence set up between the class of positive integers and the class of specific problems. We can, rather arbitrarily, represent the positive integer n by marking the first n boxes to the right of the starting point. The general problem will then be said to be *1-given* if a finite 1-process is set up which, when applied to the class of positive integers as thus symbolized, yields in one-to-one fashion the class of specific problems constituting the general problem. It is convenient further to assume that when the general problem is thus 1-given each specific process at its termination leaves the worker at the starting point. If then a general problem is 1-given and 1-solved, with some obvious changes we can combine the two sets of directions to yield a finite 1-process which gives the answer to each specific problem when the latter is merely given by its number in symbolic form.

With some modification the above formulation is also applicable to symbolic logics. We do not now have a class of specific problems but a single initial finite marking of the symbol space to symbolize the primitive formal assertions of the logic. On the other hand, there will now be no direction of type (C). Consequently, assuming applicability, a deterministic process will be set up which is *unending*. We further assume that in the course of this process certain recognizable symbol groups, i.e., finite sequences of marked and unmarked boxes, will appear which are not further altered in the course of the process. These will be the derived assertions of the logic. Of course the set of directions corresponds to the deductive processes of the logic. The logic may then be said to be *1-generated*.

An alternative procedure, less in keeping, however, with the spirit of

[5] While our formulation of the set of directions could easily have been so framed that applicability would immediately be assured it seems undesirable to do so for a variety of reasons.

symbolic logic, would be to set up a finite 1-process which would yield the *n*th theorem or formal assertion of the logic given *n*, again symbolized as above.

Our initial concept of a given specific problem involves a difficulty which should be mentioned. To wit, if an outside agency gives the initial finite marking of the symbol space there is no way for us to determine, for example, which is the first and which the last marked box. This difficulty is completely avoided when the general problem is 1-given. It has also been successfully avoided whenever a finite 1-process has been set up. In practice the meaningful specific problems would be so symbolized that the bounds of such a symbolization would be recognizable by characteristic groups of marked and unmarked boxes.

The root of our difficulty however, probably lies in our assumption of an infinite symbol space. In the present formulation the boxes are, conceptually at least, physical entities, e.g., contiguous squares. Our outside agency could no more give us an infinite number of these boxes than he could mark an infinity of them assumed given. If then he presents us with the specific problem in a finite strip of such a symbol space the difficulty vanishes. Of course this would require an extension of the primitive operations to allow for the necessary extension of the given finite symbol space as the process proceeds. A final version of a formulation of the present type would therefore also set up directions for generating the symbol space.[6]

The writer expects the present formulation to turn out to be logically equivalent to recursiveness in the sense of the Gödel–Church development.[7] Its purpose, however, is not only to present a system of a certain logical potency but also, in its restricted field, of psychological fidelity. In the latter sense wider and wider formulations are contemplated. On the other hand, our aim will be to show that all such are logically reducible to formulation 1. We offer this conclusion at the present moment as a *working hypothesis*. And to our mind such is Church's identification of effective calculability with recursiveness.[8]

[6] The development of formulation 1 tends in its initial stages to be rather tricky. As this is not in keeping with the spirit of such a formulation the definitive form of this formulation may relinquish some of its present simplicity to achieve greater flexibility. Having more than one way of marking a box is one possibility. The desired naturalness of development may perhaps better be achieved by allowing a finite number, perhaps two, of physical objects to serve as pointers, which the worker can identify and move from box to box.

[7] The comparison can perhaps most easily be made by defining a 1-function and proving the definition equivalent to that of recursive function. (See Church, loc. cit., p. 350.) A 1-function $f(n)$ in the field of positive integers would be one for which a finite 1-process can be set up which for each positive integer n as problem would yield $f(n)$ as answer, n and $f(n)$ symbolized as above.

[8] Cf. Church, loc. cit., pp. 346, 356–58. Actually the work already done by Church and others carries this identification considerably beyond the working hypothesis stage. But to mask this identification under a definition hides the fact that a fundamental discovery in the limitations of the mathematicizing power of Homo Sapiens has been made and blinds us to the need of its continual verification.

> Out of this hypothesis, and because of it apparent contradiction to all mathematical development starting with Cantor's proof of the non-enumerability of the points of a line, independently flows a Gödel-Church development. The success of the above program would, for us, change this hypothesis not so much to a definition or to an axiom but to a *natural law*. Only so, it seems to the writer, can Gödel's theorem concerning the incompleteness of symbolic logics of a certain general type and Church's results on the recursive unsolvability of certain problems be transformed into conclusions concerning all symbolic logics and all methods of solvability.
>
> Post, 1936