Note 4: The Church-Turing Thesis

Long before the advent of the computer (as we understand the word today) mathematicians had been much concerned with the idea of *algorithm* or *effective procedure*: a process which proceeds in a purely mechanical fashion, according to a set of well defined rules. A number of models were proposed with the aim of formalising the notion of effective procedure, the Turing machine being one. The Turing machine is such a simple model that we can have little doubt that all procedures which can be defined within the model are effective in the above sense. However, the very simplicity of the model also leads us to question whether *all* the procedures which we would intuitively regard as effective can be described by Turing machines. Perhaps the model was made *too* simple. Perhaps a modern programming language, such as Java, can describe processes which cannot be carried out by a Turing machine.

Alan Turing argued that his model *was* a correct formulation of effective computability. He defended the following proposition, which has come to be called the *Church-Turing thesis* in acknowledgment of the contribution of the logician Alonzo Church, who proposed a parallel formalism based on ideas from logic and known as the λ-*calculus*. (Occasionally the proposition is also referred to as Turing's Thesis, or as Church's Hypothesis.)

> *Any process which could naturally be called an effective procedure can be realised by a Turing machine.*

Note that this is a *thesis* and not a *theorem*, since "effective procedure" is an intuitive notion incapable of formalisation.

One argument he advanced in favour of the thesis was an "appeal to intuition." In his original paper[1] he fashions this appeal to intuition into a powerful case, which is reproduced verbatim below. Two things need to be borne in mind when reading this abstract more than 80 years after it was written. First, it will be seen that Turing is concerned with computations of a mathematical or numeric nature, whereas today we more often encounter computers in the role of data processing devices. However, there is nothing in the argument that does not carry over to the more general setting. Second, when Turing uses the word "computer" he is clearly not using it in its modern sense; instead he means a *person* performing a computation (who might be relieved of his task when the advance of technology permits). The argument runs as follows:

> "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 nature of the paper is sometimes used. But such a use is always avoidable, and I think it will be agreed that the two-dimensional character of the paper is no essential of computation. I assume 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 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

[1] A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*, Proc. London Math. Soc. **42** (1937).

to observe more, he must use successive observations. We will also suppose that the number of states of mind which need to 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 the 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 to be 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 consisting of the computer and his tape. We know the state of the 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 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 the ‘observed’ squares.

“Besides these changes of symbols, the simple operations must include changes of distribution of the 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 squares 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 only be 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 sequences 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 those squares can be found by some process of which my type of machine is capable.

“The simple operations must therefore include:

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

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

“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.

"We may now construct a machine to do the work of this computer. To each state of mind of the computer corresponds a [state] 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 [state]. The machines just described do not differ very essentially from computing machines defined [elsewhere], and corresponding to any machine of this type a [Turing machine] can be constructed to compute the same sequence, that is to say the sequence computed by the computer."

The machine which is described in the final paragraph of the quotation is very close to the Turing machine defined in the Sipser book. The main difference is that the machine described above can observe B squares simultaneously (i.e., has B heads) and can move each head through L squares in a single move. The definition has been simplified in Sipser by setting L and B to be 1. (The machine has one head which can move one square in a single move.) It is easy to check that, as with other detailed tweaks of the model, there is no loss of generality involved in this simplification: a machine with one head moving one square at a time can do the work of a machine with B heads moving L squares at a time (albeit more slowly).

Observe that in this careful argument Turing has justified each of the apparently limiting features of his model: that the tape is one-dimensional, that the set of states Q is finite, that the set of symbols Γ is finite, and that only a fixed number of tape symbols are visible at any time. For some features he argues that it would be unreasonable to have things otherwise: an infinite set of states or an infinite tape alphabet would inevitably lead to different states or symbols being confused. For others he argues that no loss of generality is involved: any computation which can be carried out on a two-dimensional sheet of paper can equally well be carried out on a long thin tape.