Operations of a Post Worker

Robert H. C. Moir

Post (1936) provides a formulation of a system, which amounts to a kind of Turing machine, in which a certain class of problems can be specified and within which solutions to some such problems may be generated. Solutions to *specific problems* may be so generated, or for *general problems*, *i.e.* classes of specific problems, by generating a solution to each problem in the class.

The apparent motivation for specifying such a system is to provide a system that is

1. sufficiently general that it can represent any symbolic logic,
2. is of "psychological fidelity", which I interpret as meaning similar in form to how human beings compute, and
3. is as simple as possible. [1]

He suspects, but does not prove, that other similar more complex systems are logically reducible to the one he provides, and that the system is logically eqivalent to recursion in the sense given by Church and Gödel.

There are two main components to the system: a *symbol space* where the work leading to the answer to a problem is carried out; and a *set of directions*, fixed and unalterable, that determines which operations on the symbol space are to be performed and in which order. As stated it is general enough of a definition to include certain kinds of cellular automata (where the directions do not change over time) but Post provides a specific formulation of this general system.

The symbol space is taken to consist of a two-way infinite sequence of spaces or boxes, within which a worker or problem solver moves and works, being capable of marking or erasing a mark in each of the boxes, but only one box at a time. The boxes themselves have only two possible states, marked or unmarked. One box is identified as the starting point, say the 0 if a correspondence between the integers and the boxes is effected.

It is assumed that a specific problem can be given in a symbolic form by a finite number of boxes being marked, and the answer is to be given in the same form, the state of the symbol space when the worker is finished. The worker is assumed to be capable of the following operations:

(a) Marking the box the worker is in (henceforth the current box), which is assumed empty;

[1] He points out in a footnote that this condition may cohere better with the second condition by adding a finite number, perhaps two, physical objects to serve as pointers that the worker can identify and move from box to box. This would put his formulation closer in line with Turing's.

(b) Erasing the current box, which is assumed marked;

(c) Moving to the box on the right;

(d) Moving to the box on the left;

(e) Determining whether the current box is marked or not.

Finally, a set of directions must be specified. Any set of directions corresponds to some general problem, it solving each specific problem of the class. Each set of directions begins with the directive

Start at the starting point and follow direction 1

and then consists of a finite number of directions (1,2,. . . ,n), where each direction i has one of the following forms:

(A) Perform operation O_i, where $O_i = (a), (b), (c), (d)$, and then follow direction j_i;

(B) Perform operation (e) and if yes follow direction j_i', else follow direction j_i'';

(C) Stop.

Post defines a set of directions to be *applicable* to a general problem if the worker is never directed to mark a marked box or erase an empty one. Then, a set of directions is then said to set up a *finite 1-process* in connection with a general problem if it is applicable to the general problem and *if the process terminates for each specific problem.* A finite 1-process associated with a general problem is said to be a *1-solution* of the problem if the answer it yeilds is correct for each specific problem.

As specified, the system assumes that any given problem is already provided in a coded form. This can also be internalized for any given general problem by setting up a 1-1 correspondence with the natural numbers and the class of specific problems. The natural numbers can be coded by letting each natural number n be coded by the marking of the first n boxes to the right of the starting point. Post then calls a general problem *1-given* if a finite 1-process is set up such that when applied to each coded natural number yields the corresponding specific problem, resetting the worker to the starting point. It is then easy to see how a 1-given and 1-solved general problem can be solved by specifying the natural number code of the problem and composing the two processes.

Post then describes how this formulation is applicable to formal logics. A single initial finite marking of the symbol space can be used to symbolize the primitive formal assertions of the logic. Then, with no (C) direction, an unending process may be set up to produce recognizable invariant finite sequences of marked and unmarked boxes corresponding to the derived assertions of the logic. If the deductive processes of the logic are given in such a way that the logic is said to be *1-given.*

Post mentions the strong idealization involved in assuming an infinite symbol space. He points out that this can be weakened by assuming an indefinitely extendible symbol space, provided that the primitive operations are extended to include directions for extending the symbol space. This, of course, still involves a significant idealization.

Post cares about the idealization involved in the initial formulation because he is interested in the philosophical implications of the logical reductions he thinks are possible to his system, and its supposed equivalence to recursion. His way of describing the philosophical import of Church's identification of effective calculability and recursivness is that "a fundamental discovery in the limitations of the mathematicizing power of Homo Sapiens has been made".[2] (Post, 1936, 105) He thinks that in order for this program to succeed, to make Church's and Gödel's limitative results yield conclusions for all symbolic logics and all methods of solvability, Church's thesis would have to be elevated to the status of a natural law. Thus, it is necessary that the model of computation does not include forms of idealization that would be incompatible with what physical computing systems could actually do.

References

Post, E.L. 1936. Finite combinatory processes-formulation 1. *Journal of Symbolic Logic*, 103–105.

[2]He also emphasizes that that Church's thesis has this status makes considering it as a definition inappropriate. To do so "blinds us to the need of its continual verification". (Post, 1936, 105) This underwrites his contention that Church's thesis must be elevated to the status of a natural law.