INFORMATION AND CONTROL 54, 3–24 (1982)

Why Gödel Didn't Have Church's Thesis*

MARTIN DAVIS

Courant Institute, New York, New York 10012

To celebrate the occasion of the twentieth anniversary meeting on Foundations of Computer Science, in October 1979, it was held at a very special location, San Juan, Puerto Rico, and three distiguished pioneers of theoretical computer science, Sheila Greibach, Juris Hartmanis, and Stephen C. Kleene were invited to give addresses on the history of the field. The present article was directly stimulated by my hearing Kleene's thoroughly delightful talk "Origins of Recursive Function Theory," which is now available in printed form (Kleene, 1981). It was my great good fortune to have been, during the late 1940s, a student of two of the most important early workers in the field of recursive function theory, Alonzo Church and Emil Post. Later, I edited an anthology (Davis, 1965) of basic papers in the field and marvelled at the richness of the interactions among the remarkable community of logicians that historical crosscurrents had brought to the East coast of the United States, and especially to Princeton, New Jersey, in the 1930s. It is truly remarkable (Gödel, 1946, speaks of a "kind of miracle") that it has proved possible to give a pecise mathematical characterization of the class of processes that can be carried out by purely mechanical means. It is in fact the possibility of such a characterization that underlies the ubiquitous applicability of digital computers. In addition it has made it possible to prove the algorithmic unsolvability of important problems, has provided a key tool in mathematical logic, has made available an array of fundamental models in theoretical computer science, and has been the basis of a rich new branch of mathematics. Kleene's account, which is particularly valuable bacause he is able to write as one of the key participants in the unfolding drama, restimulated my interest in the early history of these ideas. Another source of stimulation was the appearance of Webb (1980), a provocative philosophical and historical study of Church's thesis at an unusually deep level. I am very grateful for the extremely helpful criticisms, corrections, and new historical material provided by Kleene after reading a preliminary version of this article, although, of course, responsibility for the opinions expressed is entirely my own.

* Work supported by National Science Foundation Grant MCS-8002438. This article is part of the NSF-sponsored Workshop on Recursion Theoretic Aspects of Computer Science held at Purdue University in May 1981.

0019-9958/82 $2.00

1. λ-Definability

The work of Church, Kleene, and Rosser in the early 1930s was largely in the context of the λ-calculus. The idea of the λ-notation is familiar to most computer scientists.[1] (Thus $\lambda x[x^2]$ denotes the "squaring function", and $\{\lambda x[x^2]\}(3)$ denotes the number 9.) It was incorporated by John McCarthy into his LISP, and has been employed in various studies of programming languages. More recently it has been fundamental in the development of so-called *denotational semantics*. The λ-notation was developed by Church as part of an attempt to produce a logical system which would be adequate for ordinary mathematics in which the notion of function or mapping would play a fundamental role. Since the λ-operator converts an expression containing free variables into one which denotes a function, the λ-notation arose out of this project in a very natural way. Church published a pair of substantial papers[2] on the system he developed and set his students Stephen C. Kleene and J. Barkley Rosser to work on it. Their work was extremely effective, if not exactly what one dreams of having one's graduate students accomplish for one: Kleene and Rosser proved that Church's system was invonsistent![3] Although this ended Church's hopes for his ambitious system, it seemed that it should not be the end for something as natural and elegant as the λ-notation. In fact it turned out to be possible to extract a demonstraby consistent subsystem of Church's system, the λ-calculus.[4] The λ-calculus is developed using rules of *λ-conversion* by means of which expressions of the λ-calculus may be transformed in such a manner that the object an expression intuitively denotes remains unchanged under conversion. (An example would be $\{\lambda x[x^2]\}(3)$ which can be "converted" to 3^2. Other rules would permit conversion of $\lambda x[x^2]$ into $\lambda y[y^2]$ and of 3^2 into $\{\lambda x[x^2]\}(3)$. This example is not quite right because x^2 and 3 are not expressions of the "pure" λ-calculus, but it serves to give the idea, and in any case, as will be clear shortly, the missing items can be introduced by definition.) Church had proposed to develop arithmetic within his system by using suitable λ-expressions to code the positive integers. His code, which had particular technical advantages, was as follows:

[1] For those unfamiliar with the λ-notation, Kleene (1981) contains a brief and excellent intuitive introduction and further references. See in particular Church (1935).

[2] For references, see Kleene (1981).

[3] The list of logicians who have seriously proposed systems of logic that have later turned out to be inconsistent reads like an honor roll. It includes, in addition to Church: Frege, Curry, Quine, and Rosser.

[4] See Church (1935). Actually there are several λ-calculi which are minor variants of one another.

1 *stands for* $\lambda x[\lambda y[\{x\}](y)]]$,
2 *stands for* $\lambda x[\lambda y[\{x\}(\{x\}(y))]]$,
3 *stands for* $\lambda x[\lambda y[\{x\}(\{x\}(\{x\}(y)))]]$,

etc. Using this coding or abbreviation, we may say that an expression M of the λ-calculus *λ-defines* the (total) function f from the positive integers to the positive integers if the expression $\{M\}(n)$ is convertible by the rules of λ-conversion into $f(n)$ for each positive integer n. Similarly a function f of two variables is *λ-defined* by M if $\{\{M\}(m)\}(n)$ is convertible to $f(m, n)$ for all positive integers m, n, and similarly for functions of three or more variables. A function is then *λ-definable* if there is some λ-expression which λ-defines it. This definition is all but inevitable in the present context; Kleene (1981, p. 55) says, "··· we cannot escape [it]." It seems to have developed out of Kleene's realization in 1932 that he could prove one of Peano's postulates in Church's system if he could find an expression which λ-defines the predecessor function, and Church (1936, footnote 3; Davis, 1965, p. 90) credits the notion of λ-definability "jointly to [Church] and S. C. Kleene." Kleene quickly succeeded in showing that the predecessor function is λ-definable. Church's original intuition for the extent of the class of λ-definable functions was such that Kleene's result (1981, p. 57) surprised him. But Kleene soon showed that the class was very extensive indeed, containing all primitive recursive functions and being closed under minimalization. It was easy to transform any λ-expression which λ-defines a function into an explicit algorithm for computing that function. The decisive step was to declare that the converse was likewise true, that any mechanically calculable function was indeed λ-definable.

2. General Recursive Functions

The class of what are now called *primitive recursive functions* was introduced in 1931 by Gödel in his epoch-making paper on undecidable propositions, where he called them simply "rekursiv." Previously, Dedekind, Peano, and Skolem among others had studied recursive definitions. Hilbert had suggested the use of more general kinds of recursions (e.g., on more than one variable simulatenously) and Ackermann followed up on this suggestion by using such a recursive definition to construct his famous example of a function which is not primitive recursive. In the period February to May 1934, Gödel gave a series of lectures at the Institute for Advanced Study in Princeton which have been preserved in the lecture notes taken by Kleene and Rosser (Gödel, 1934). In these lectures Gödel (1934; Davis, 1965,

pp. 43, 44)[5] noted that primitive recursive functions "have the important property that, for each given set of values of the arguments, the value of the function can be computed by a finite procedure." To this remark Gödel (1934, footnote 3; Davis, 1965, p. 44) added the suggestive footnote[6]

> The converse seems to be true, if besides [primitive] recursions ⋯ recursions of other forms (e.g., with respect to two variables simultaneously) are admitted. This cannot be proved, since the notion of finite computation is not defined, but it serves as a heuristic principle.

We may refer to

GÖDEL'S THESIS.[7] *Every mechanically calculable function can be defined using recursions of the most general kind.*

Later in the same lectures, Gödel (1934, Davis, 1965, p. 69), following up on a suggestion of Jacques Herbrand, proposed an answer to the question of "what one would mean by 'every recursive function'." This answer anticipated the notion of "recursive" as used by computer scientists as in *recursive programs*. The idea is to permit definition of a function by using equations connecting values of the function with other values of the same function and of other functions in the most general conceivable manner. It is only required (this last was essentially Gödel's addition to Herbrand's conception) that values of the function be derivable from the equations using only the simplest rules of substitution. The idea should be clear from a pair of examples. First consider the four equations

$$\sigma(x, 0) = x,$$
$$\sigma(x, Sy) = S\sigma(x, y),$$
$$\pi(x, 0) = 0,$$
$$\pi(x, Sy) = \sigma(\pi(x, y), x).$$

Here S stands for *successor* and we are using the primitive notation $S \cdots S0$ for natural numbers. These equations give a recursive definition of multiplication. Here is a derivation of $2 \cdot 2 = 4$ from these equations:

[5] I am indebted to Kleene for the information that Gödel's lectures took place in the period February through May 1934. In correspondence, Kleene stated: "... the cover page of my original set of Gödel's notes reads 'Notes on lectures by KURT GÖDEL February–May, 1934'."

[6] The word "primitive" in brackets has been added in accord with contemporary usage.

[7] Webb (1980, pp. 186, 188, 203) properly emphasizes the roots of this "thesis" in the ideas of Skolem and Hilbert.

$$\begin{aligned}\sigma(0, SS0) &= S\sigma(0, S0)\\ &= SS\sigma(0, 0)\\ &= SS0,\end{aligned}$$

$$\begin{aligned}\pi(SS0, S0) &= \sigma(\pi(SS0, 0), SS0)\\ &= \sigma(0, SS0)\\ &= SS0,\end{aligned}$$

$$\begin{aligned}\sigma(SS0, SS0) &= S\sigma(SS0, S0)\\ &= SS\sigma(SS0, 0)\\ &= SSSS0,\end{aligned}$$

$$\begin{aligned}\pi(SS0, SS0) &= \sigma(\pi(SS0, S0), SS0)\\ &= \sigma(SS0, SS0)\\ &= SSSS0.\end{aligned}$$

Of course multiplication is primitive recursive and we have just used the ordinary primitive recursive definitions of addition and multiplication. In the following more revealing example,[8] it is assumed that the three equations we are writing follow a list of equations which permit calculation of the values of ρ:

$$\begin{aligned}\sigma(0, x, y) &= y,\\ \sigma(Sz, x, y) &= \sigma(\rho(x, Sy), x, Sy),\\ \phi(x) &= \sigma(\rho(x, 0), x, 0).\end{aligned}$$

It is not difficult to see that if the full set of equations is used to calculate values of ϕ, then the function obtained is

$$\min_y[\rho(x, y) = 0].$$

A precise definition of general recursive functions of course requires that the details be filled in some precise manner. Gödel in his lectures filled in the details in one particular way. Later Kleene (1936) showed that the class of functions obtained was not at all sensitive to the details of the definition.

When I was preparing the anthology, Davis (1965), it seemed to me that Gödel's footnote above (which I have paraphrased as Gödel's thesis) together with his proposed definition of general recursive function in the same lectures amounted to a precise characterization of the mechanically calculable functions and hence to a statement of Church's thesis. I suggested as much in a first draft of my brief introduction to Gödel's lectures, and

[8] Due to Kleene (1936a, 1943); see Davis (1965, pp. 247, 259).

submitted the draft to Gödel for his comments. In a reply, dated February 15, 1965, Gödel took strong exception to my suggestion. He wrote:

> ··· it is *not true* that footnote 3 is a statement of Church's Thesis. The conjecture stated there only refers to the equivalence of "finite (computation) procedure" and "recursive procedure." However, I was, at the time of these lectures, not at all convinced that my concept of recursion comprises all possible recursions; and in fact the equivalence between my definition and Kleene's[9] ··· is not quite trivial.

In the light of Gödel's letter we can clearly characterize Gödel's views during the spring of 1934. He believed "Gödel's Thesis" as a heuristic guide and had even attempted a definition of "general recursive" function. But he was still "not at all convinced" that his definition was sufficiently inclusive.

3. Church's Thesis

In the published version of his 1979 address, Kleene (1981, p. 59) states[10]

> The concept of λ-definability existed full-fledged by the fall of 1933 and was circulating among the logicians at Princeton. Church had been speculating, and finally definitely proposed, that the λ-definable functions are all the effectively calculable functions.... When Church proposed this thesis, I sat down to disprove it by diagonalizing out of the class of the λ-definable functions. But, quickly realizing that the diagonalization cannot be done effectively, I became overnight a supporter of the thesis.

Kleene has explained (in personal correspondence) that he did not intend this account to place Church's "definite proposal" in the fall of 1933. Kleene states that "all of these events... (except Church's earliest speculations)" took place after Kleene's return to Princeton on February 7, 1934, and before something like the end of March 1934. Since Gödel's lectures at the Institute for Advanced Study were taking place during the period Febrary through May 1934, Church's statement of his "thesis" occurred either just before these lectures began or while they were in progress.

In tracing the development of Church's ideas, it is interesting to consider an address entitled "The Richard Paradox" (Church, 1934) which Church delivered in December 1933. We quote from Church's address

> The Richard paradox can be said to consist in the following problem. How is it possible that a system of symbolic logic, in which the set of all formulas is enumerable, should be adequate for any branch of mathematics which deals with the members of a non-enumerable set... ?

[9] That is, in Kleene (1936a).

[10] Kleene suggests, as a clarification of the chronology, adding the words "in 1934" at the beginning of his line 4 from the bottom, left-hand column.

> Given a system of symbolic logic, let us try to construct the function of positive integers such that there is no formula in the system that stands for it. What we must do is first to enumerate all formulas, and then, going through this enumeration, to pick out in order those formulas which stand for functions of positive integers. The result is an enumeration of all formulas which stand for functions of positive integers. And if we let $f_n(x)$ be the function of positive integers represented by the nth formula in this enumeration, then $1 + f_x(x)$ is the function of positive integers such that there is no formula in the system that stands for it.
>
> But this function $1 + f_x(x)$ is not, in general, defined in such a way that it is always possible to calculate its value for a given positive integer x. For, in the process of going through the list of all formulas and picking out those which stand for functions of positive integers, we may at some stage find a formula about which we do not know whether or not it stands for a function of positive integers. ... Indeed, to be sure of always being able to determine whether a given formula stands for a function of positive integers, we must have discovered a method of procedure which would enable us to solve any problem of number theory whatever. Therefore the infinite sequence (about which we have been talking) of all formulas which stand for functions of positive integers almost certainly is not such an infinite sequence that it is possible to calculate as many terms of it as we please. And therefore the function $1 + f_x(x)$ has not been defined in a way which could be called constructive, but has merely been proved by an indirect argument to exist. (Church 1934, pp. 357–358)

Although Church's keen interest in effective calculability is evident and he is clearly aware of the relevance of the diagonal construction, there is no hint that he proposed to identify effective calculability with some precise mathematical notion.

In a letter to Kleene[11] dated November 29, 1935, Church gives a fascinating account of his discussion of effective calculability with Gödel, presumably early in 1934.

> In regard to Gödel and the notions of recursiveness and effective calculability, the history is the following. In discussion [*sic*] with him the notion of lambda-definability, it developed that there was no good definition of effective calculability. My proposal that lambda-definability be taken as a definition of it he regarded as thoroughly unsatisfactory. I replied that if he would propose any definition of effective calculability which seemed even partially satisfactory I would undertake to prove that it was included in lambda-definability. His only idea at the time was that it might be possible, in terms of effective calculability as an undefined notion, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis. Evidently it occurred to him later that Herbrand's definition of recursiveness, which has no regard to effective calculability, could be modified in the direction of effective calculability, and he made this proposal in his lectures. At that time he did specifically raise the question of the connection between recursiveness in this new sense and effective calculability, but said he did not think that the two ideas could be satisfactorily identified "except heuristically."

[11] A copy of this letter was supplied to me by Kleene. Kleene (1981, p. 59) gives a shorter excerpt from this letter.

Church's announcement to the mathematical world of this "thesis" was at a meeting of the Americal Mathematical Society in New York City on April 19, 1935 in a ten minute contributed talk. We quote in full the abstract (Church, 1935) of that paper (received by the Society on March 22, 1935).

> Following a suggestion of Herbrand, but modifying it in an important respect, Gödel has proposed (in a set of lectures at Princeton, N. J., 1934) a definition of the term *recursive function*, in a very general sense. In this paper a definition of *recursive function of positive integers* which is essentially Gödel's is adopted. And it is maintained that the notion of an effectively calculable function of positive integers should be identified with that of a recursive function, since other plausible definitions of effective calculability turn out to yield notions which are either equivalent to or weaker than recursiveness. There are many problems of elementary number theory in which it is required to find an effectively calculable function of positive integers satisfying certain conditions, as well as a large number of problems in other fields which are known to be reducible to problems in number theory of this type. A problem of this class is the problem to find a complete set of invariants of formulas under the operation of conversion (see abstract 41-5-204). It is proved that this problem is unsolvable, in the sense that there is no complete set of effectively calculable invariants.

Thus, Church chose to state his thesis not in terms of λ-definability, but rather in terms of "Herbrand–Gödel general recursiveness." It is interesting that λ-definability occurs only by implication in the reference to "other plausible definitions of effective calculability... either equivalent to or weaker than recursiveness."The wording leaves the impression that in the early spring of 1935 Church was not yet certain that λ-definability and Herbrand–Gödel general recursiveness were equivalent. (This despite Church's letter of November 1935 in which he reported that in the spring of 1934 he had offered to Gödel to prove that "any definition of effective calculability which seemed even partially satisfactory... was included in lambda-definability.") The full paper (Church, 1936) which appeared in the April 1936 issue of the *American Journal of Mathematics* does not contradict this impression. The fact that avery λ-definable function is recursive is there said to have been obtained by Church and by Kleene "independently... at about the same time," whereas the converse is attributed to Kleene. In fact, an abstract of Kleene (1936b) which announced th equivalence of λ-definability and recursiveness was received by the American Mathematical Society on July 1, 1935.[12] Also received on July 1, 1935 was an abstract of Kleene (1936a) which contained Kleene's famous

[12] Kleene (1981, p. 60) mistakenly asserts that Church (1936) also contains such an equivalence proof; Kleene suggests the correction: on page 60, line 17, left-hand column, omit the words "Church (*1936*) and" and change "equivalence proofs" to "an equivalence proof." Kleene also lists four additional errata: Page 52, right-hand column, add at the end of line 5 the words "the first of" so that what is said "might not be taken as implying incorrectly that Gödel's second theorem was announced at the Königsberg conference." Page 57, left-hand

NORMAL FORM THEOREM. *Every general recursive function can be expressed in the form* $f(\min_y(g(x_1,\ldots,x_n,y)=0))$, *where* f *and* g *are primitive recursive functions.*

This theorem has made equivalence proofs for formalisms in recursive function theory rather routine, and must have gone a considerable distance towards convincing Gödel that his "concept of recursion" indeed "comprises all possible recursions." That Church and Kleene did not have the results Kleene (1936a, b) available at the time of the April 1935 meeting is also suggested by the fact that Kleene, whose name is included in the published list of those attending that meeting, did not give at least one of these papers at the meeting.

Of course, by the time the final version of Church (1936) was written, Church was fully aware of the equivalence of λ-definability and recursiveness, and he was able to argue forcefully that

> The fact, however, that two such widely different and (in the opinion of the author) equally natural definitions of effective calculability turn out to be equivalent adds to the strength of the reasons addressed below for believing that they constitute as general a characterization of this notion as is consistent with the usual intuitive nderstanding understanding of it. (Church, 1936; Davis, 1965, p. 90)

However, recursiveness continues to get top billing even in the published version which contains this "official" statement of Church's thesis:

> We now define the notion, already discussed, of an *effectively calculable* function of positive integers by identifying it with the notion of a recursive function of positive integers (or of a λ-definable function of positive integers). This definition is thought to be justified by the considerations which follow, so far as positive justification can every be obtained for the selection of a formal definition to correspond to an intuitive notion. (Church, 1936; Davis, 1965, p. 100)

In a footnote to the just cited paragraph, Church briefly alludes to a conversation with Gödel which evidentally occurred after the discussion mentioned in his letter to Kleene.

> The question of the relationship between effective calculability and recursiveness (which it is here proposed to answer by identifying the two notions) was raised by Gödel in conversation with the author. The corresponding question of the relationship between effective calculability and λ-definability had previously been proposed by the author independently.

We can now summarize. In the early months of 1934, Church (1936; Davis, 1965, p. 100) "proposed" the "question of the relationship between effective calculability and λ-definability." He definitely asserted their

---

column, line 21 from bottom, "retain" should be "contain." Page 63, left-hand column, line 4 from bottom, "*1944*" should be "*1954*." Page 64, left-hand column, bottom line, "Δ_0^1" should be "Δ_1^0".

equivalence to his student Kleene, and he suggested to Gödel in conversation that effective calculability simply be *defined* to be λ-definability. Gödel indicated that he found Church's proposal "throughly unsatisfactory." The discussion took place around the time that Gödel (1934) was giving his lectures. Church tells us that in the conversation "it developed that there was no good definition of effective calculability." Very near the beginning of Gödel's lectures, in connection with what we have called Gödel's Thesis, appears the statement[13] "... the notion of finite computation is not defined." It is certainly tempting to imagine that there is a connection here. But it seems useless to speculate whether the conversation may have developed out of Gödel's lectures, or whether Gödel may have been stimulated by the conversation to bring up these matters in his lectures. In his cited letter to Kleene, Church wrote, speaking of Gödel, "Evidently it occurred to him later that Herbrand's definition of recursiveness, which had no regard to effective calculability, could be modified in the direction of effective calculability, and he made this proposal in his lectures."[14] The question of the equivalence of the class of these general recursive functions with the effectively calculable functions was implicitly raised in "Gödel's Thesis," and, as we have seen, explicitly raised by Gödel in conversation with Church. Nevertheless, Gödel was not convinced by the available evidence, and remained unwilling to endorse the equivalence of effective calculability, either with recursiveness or with λ-definablity. He insisted (as Church later reported to Kleene) that it was "thoroughly unsatisfactory" to *define* the effectively calculable functions

[13] With regard to this statement, I asked in a letter to Kleene, "Did the statement of footnote 3 of Gödel's 1934 lectures ... occur in the oral presentation as early as its position in the text suggests ...?" Kleene replied (emphasis his): "It *very likely* did .... But I can't be positive."

[14] Thus Church had concluded that Gödel thought of his "general" definition of recursive function only after his discussion of effectiveness with Church. Concurring with this conclusion, Kleene (1981) in his introductory abstract writes, "The notion of 'λ-definability' was the first of what are now accepted as equivalent exact mathematical descriptions of the class of functions for which algorithms exist." Perhaps thinking of the order of publication (and accepting the Kleene–Rosser mimeographed notes on Gödel's lectures as a publication), Turing (1939; Davis, 1965, p. 160) wrote, "Such a definition [i.e., of an effectively calculable function] was first given by Gödel at Princeton in 1934 ...." It should perhaps be noted that if it is a question of simply giving an "exact mathematical description of the class" as opposed to singling out the class as consisting of the "functions for which algorithms exist," then the class (or more accurately, the corresponding class of relations) is already to be found in 1931 in Gödel's notion of a relation being *entscheidungsdefinit* (p. 189 of the German original), or *decidable* (Davis, 1965, p. 26).

Although it is certainly interesting to attempt to recover the order of events in this fascinating drama of ideas, what is much more interesting than who did what first is the remarkable fact that all of the proposed answers to the question: "Which functions are effectively calculable?" turned out to be correct and equivalent to one another.

to be some particular class without first showing that "the generally accepted properties" of the notion of effective calculability necessarily lead to this class. As we shall see, it was not until Turing's work became known that Gödel was willing to concede that this difficulty had been overcome.

Meanwhile, Church and Kleene each proved that all λ-definable functions are recursive. Church submitted an abstract of his work on March 1935, basing himself on recursiveness rather than λ-definability. By the end of June 1935, Kleene had shown that every recursive function is λ-definable, after which Church (1936) was able to put his famous work into its final form. Thus while Gödel hung back because of his reluctance to accept the *evidence* for Church's thesis available in 1935 as decisive, Church (who after all was right) was willing to go ahead, and thereby to launch the field of recursive function theory.

Church was immediately aware of the significance of this work for the possibility of obtaining unsolvability results for problems of independent mathematical interest. The main unsolvability result obtained in Church (1936) was for a problem in the λ-calculus (that of determining whether or not a given formula can be "converted" into a formula in so-called "normal" form). Church emphasized that this problem "appears to be of the same class as ... problems of number theory and topology.... The temptation is strong to reason by analogy that other important problems of this class may also be unsolvable." Indeed, Post's unsolvability proof (1947) for the word problem for semigroups was the result of a suggestion by Church that this was an appropriate problem on which to try Post's combinatorial methods.

It is also worth noting that Church calls the identification of effective calculability with recursiveness a "definition." The use of the word "thesis" in this connection was proposed by Kleene (1943; Davis, 1965, p. 274) much later. As we shall see, for reasons not unrelated to Gödel's scruples, Post was greatly opposed to speaking of Church's thesis as a "definition."

4. Turing Machines and Gödel's Thinking

Turing's model of computation is of course very well known to computer scientists. In a "postscriptum" to his 1934 lecture notes which Gödel prepared for (Davis 1965), Gödel made it clear that in his view, Turing's work was of fundamental importance in establishing the validity of Church's thesis. Gödel stated

> Turing's work gives an analysis of the concept of "mechanical procedure" (alias "algorithm" or "computation procedure" or "finite combinatorial procedure"). This concept is shown to be equivalent with that of a "Turing machine." (Davis, 1965, p. 72)

In a footnote that references Turing's work and "the almost simultaneous paper by E. L. Post" (1936), Gödel goes on to say

> As for previous equivalent definitions of computability, which, however, are much less suitable for our purposes, see A. Church [1936].... One of these definitions is given in ... these lectures.

Thus what was crucial for Gödel was Turing's "analysis of the concept of 'mechanical procedure' " which Gödel insists *shows* "this concept" to be "equivalent with that of a 'Turing machine'."

Turing did his work (1936–1939) in England entirely independently of the related research being done in Princeton. As Kleene (1981), p. 61 stated,

> Turing learned of the work at Princeton on λ-definability and general recursiveness just as he was ready to send off his manuscript, to which he then added an appendix outlining a proof of the equivalence of his computability to λ-definability."

Turing's "analysis" (1936–1937; Davis, 1965, pp. 135–138) is a remarkable piece of applied philosophy in which, beginning with a human being carrying out a computation, he proceeds, by a process of elimination of irrelevant details, through a sequence of simplifications, to an end result which is the familiar model consisting of a finite state device operating on a one-way infinite linear tape.[15] In the letter from Church to Kleene which we have been citing, Church says of Gödel, "His only idea at the time was that it might be possible, in terms of effective calculability as an undefined notion, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis." Now, this is very much in line with what Turing accomplished. Although his treatment was not "axiomatic" in any formal sense, he did manage to show that "generally accepted properties" of effective calculability lead inevitably to a definite class of functions (which subsequently turned out to be the same as the λ-definable or recursive functions). It is therefore not difficult to see why Turing's work was so crucial for Gödel. (For another discussion of some of these matters, see Wang (1974, pp. 81–99).

Turing's analysis leads to what we may call

TURING'S THESIS. *Every algorithm can be programmed on a one-tape Turing machine.*

Turing's paper contains a great deal of interesting material in addition to his analysis, including the unsolvability of the halting problem and the

[15] This analysis is still very much worth reading. I regard my having failed to mention this analysis in my introduction to Turing's paper in Davis (1965) as an embarrassing omission.

decision problem for first order logic, as well as the now familiar construction of a universal Turing machine.[16]

Independent of Turing's work, but not of the work in Princeton, Emil Post (1936; Davis, 1965, pp. 289–291) formulated yet another equivalent version of computability which is extremely close to Turing's. Post's formulation used a two-way infinite tape or "symbol space" and lists of instructions (today we would call them "programs") rather than finite-state automata. In his paper, Post took strong exception to Church's used of the word "definition" in his statement of his "thesis." He emphasized that the purpose of his formulation "... is not only to present a system of a certain logical potency but also, in its restricted field, of psychological fidelity." "Church's identification of effective calculability with recursiveness" is characterized as a "working hypothesis." Post continued in a footnote,

> 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.

Post proposed a program of developing "wider and wider formulations" maintaining "psychological fidelity" all of which could presumably be shown to be "logically reducible" to the present formulation. The success of this program would "change this hypothesis not so much to a definition or to an axiom but to a *natural law*."

Our discussion of Gödel's ideas would not be complete without mention of his notion of a function being *computable* ("rechenbar") in a formal system. On June 19, 1935,[17] Gödel, having returned briefly to Austria, gave a colloquium talk at the University of Vienna, an abstract of which has been published (Gödel, 1936). The main content of the paper is a "speedup" theorem for the length of proof of propositions in a formal system. The precise statement of the theorem uses the notion of a function $\phi(x)$ being "*computable in*" a formal system S, which Gödel defines as meaning that "to each numerical m there corresponds a numeral n such that $\phi(m) = n$ is provable in S." Gödel defines a sequence $S_1, S_2, \ldots$ of formal systems each stronger than the preceding and speaks of functions being computable in S_i

[16] Turing's universal machine contains serious bugs. See Post (1947; Davis, 1965, Appendix, pp. 299–303).

[17] The original gives the date of this talk as June 19, 1934. I am indebted to John W. Dawson, Jr. for calling my attention to his realization that this must have been a typographical error. As he explained, "That it *is* a mistake is indicated by the conflict between the dates and the colloquium session numbers; e.g., the 80th session met June 11, 1934, while the session at which Gödel presented his length-of-proof result was the 92nd, which must have been much more than 8 days later."

as though the notion were really dependent on i. However, while correcting the printer's proofs, Gödel (Davis, 1965, p. 83) added the "remark"

> It may also be shown that a function which is computable in one of the systems S_i or even in a system of transfinite type, is already computable in S_1. Thus, the concept "computable" is in a certain definite sense "absolute," while practically all other familiar metamathematical concepts (e.g. provable, definable, etc.) depend quite essentially on the system with respect to which they are defined.

It would be of great interest to known just when Gödel first realized that "computability" in this sense is "absolute." Evidently not at the time of his Princeton 1934 lectures, since they preceded his Vienna talk. The absoluteness result is a trivial consequence of Kleene's normal form theorem, but it is easy to imagine how Gödel could have obtained it without knowing the normal form theorem. (It can be proved in two lines, using the methods of Gödel (1931), that a function ϕ is computable in any one of the systems S_i if and only if the predicate $y = \phi(x)$ can be expressed in the form $(\exists z)\, R(x, y, z)$, where R is a *primitive* recursive predicate.) In Gödel's address (1946) before the Princeton University Bicentennial Conference, he emphasized the significance of this absoluteness:

> Tarski has stressed in his lecture (and I think justly) the great importance of the concept of general recursiveness (or Turing's computability). It seems to me that this importance is largely due to the fact that with this concept one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen. In all other cases treated previously, such as demonstrability or definability, one has been able to define them only relative to a given language, and for each individual language it is clear that the one thus obtained is not the one looked for. For the concept of computability however, although it is merely a special kind of demonstrability or decidability the situation is different. By a kind of miracle it is not necessary to distinguish orders, and the diagonal procedure does not lead outside the defined notion.

It should be mentioned also that Church (1936; Davis, 1965, pp. 101–102) defines a function being "calculable within" a logic exactly in the manner Gödel did and notes that all functions calculable within a logic are recursive.

5. Partial Functions

Constructivists in mathematics are those who insist that proofs of existential propositions are only correct insofar as they provide constructions of the objects whose existence is asserted. If "construction" is taken to mean "algorithm," then the relevance of Church's thesis to constructivism is clear. However, the definitions of λ-definable function, recursive function, Turing

computable function, etc. are all then subject to criticism by constructivists on the grounds of circularity. To take the example of general recursiveness: a function $f(x)$ is general recursive if there exists a system of equations E which, for each nonnegative integer value of x, leads to a unique value for f by "appropriate" substitutions in the equations. That is, for every x, there must be a suitable derivation from E. To a constructivist, this means that there is a "construction" for obtaining such a derivation from a given x. If "construction" means "algorithm," this means that the derivation is effectively calculable from x. Thus our precise explication of effective calculability turns out to depend on the very notion purportedly being explained. Church (1936, footnote 10; Davis, 1965, p. 95) dealt with this criticism at the outset by simply informing the constructivist that "... he should take the existential quantifer which appears in our definition of a set of recursive equations in a constructive sense. What the criterion of constructiveness shall be is left to the reader." Not surprisingly, this eminently sensible suggestion did not end the debate.

With hindsight, it is not hard to see that the difficulty comes from considering only total functions. It was Kleene (1938) in his remarkable paper who noted that every set of equations can be used to compute a *partial* function. There is no trace of circularity, even for a constructivist, in the definition of *partial recursive function.* Of course, from the point of view of recursive function theory, the nonconstructive element in the definition of recursive function simply reflects the fact that the class of systems of equations (or equivalently, of Turing machines) which determine total functions is not a recursive set.[18]

Following Webb (1980, pp. 218–219), we can formulate Kleene's key insight, which has played a vital role in the further development of the field as

KLEENE'S THESIS. *Every mechanically computable partial function is partial recursive.*

It is only in terms of partial recursive functions that a proper statement of the famous *recursion theorem* is possible. Kleene (1938) contains a short proof of the recursion theorem, but Kleene was led to the statement by a form of the recursion theorem which he had previously developed in the context of the λ-calculus under the name *circular definition.*[19,20]

[18] It is, of course, a complete Π_2 set.

[19] See Kleene (1936b). For a penetrating discussion of the role of the recursion theorem in buttressing Church's thesis, see Webb (1980, pp. 212–219).

[20] It is difficult for those who have learned about recursive functions via a treatment that emphasized partial functions from the outset to realize just how important Kleene's

6. Post's Work 1920–1922

So far we have been discussing work that was done in the period 1931–1938. But Emil Post's research in the early 1920s, unpublished at the time, anticipated much of this later work. Moreover, Post's work developed in terms of formal structures which have proved extremely important in computer science. In his 1920 doctoral dissertation, Post (1921b) discussed formal logical systems as combinatorial mathematical calculi in the modern sense for the first time, and proved some theorems about what we would today call the propositional calculus.

Although none of those theorems were very hard, what was interesting was that Post quite clearly and consciously took the propositional calculus as a combinatorial calculus as an object of study. In particular he solved the decision[21] problem for the propositional calculus. That is, he provided an algorithm for determining whether or not a given string was derivable from the axioms using the permitted rules. The algorithm was just the truth table method which he proved equivalent to derivability. He went on from there to attempt to get algorithms for more extensive systems. At that time Whitehead and Russell had recently completed their immense work "Principia Mathematica" in which they had shown in gory detail how to develop the basis of classical mathematics in a logical calculus. What Post tried to do, partly anticipating Hilbert's later efforts, was to find algorithms for wider and wider subsystems of "Principia Mathematica," and thereby to mechanize mathematics. Whereas Hilbert and his school went on to approach the decision problem for quantification theory semantically, Post evidently felt that was not a promising direction because the combinatorial intricacies of predicate logic were too great to penetrate in that manner, and what he proposed instead was to simplify by generalization. That is, he proposed to abstract from the kind of rules that occur in quantification theory to obtain a class of rules which included them. Then he could seek algorithms for solving the decision problem for arbitrary combinatorial calculi formulated using these more general rules, and thus incidentally for ordinary quantification theory. In the process, Post developed the basic notions underlying the modern formal theory of languages.

Post's account (1965) of this early work was only published much later in the Davis anthology.[22] Actually, he considered three classes of combinatorial

---

contribution was. Thus Rogers' excellent and influential treatise (1967, p. 12) contains an historical account which gives the impression that the subject had been formulated in terms of partial functions from the beginning.

[21] Post called it the "finiteness" problem.

[22] Post, (1965, pp. 340–433). A fragment was published in Post (1943).

calculi,[23] each of which was formulated to be "canonical" in the sense of being expected to encompass the usual systems of symbolic logic being considered at that time, and, in particular, "Principia Mathematica."Quite early he was able to solve the decision problem for a very restricted case of one of these formulations, Post (1921a). Moreover, he was able to prove that all three canonical forms were equivalent to one another in the sense that a logic which could be reduced to one of these forms could be reduced to any of them, and he carried out the tedious "programming" project of reducing the predicate calculus part of "Principia Mathematica" to a system in canonical form. Post's early work led to a particular combinatorial problem, called the problem of *tag*, to which a number of different considerations seemed to lead. For example, he had tried to solve what apparently amounts to the unification problem for predicate calculus of order ω, which is now known to be unsolvable. "The general problem proving intractable, succesive simplifications thereof were considered, one of the last being this problem of 'tag'," Post (1965, p. 370). Also, attempts to extend the results of Post (1921a) again led "essentially to the selfsame problem of 'tag'. A solution thus appeared as a vital stepping stone in any further progress...." We may explain"tag" as follows:

A *tag system* consists of an alphabet Σ, a map ϕ of Σ into Σ^*, and a positive integer k. Given such a tag system and a string $w \in \Sigma^*$ which begins with the symbol $a \in \Sigma$ we obtain a new *derived string* by forming the concatenated string $w\phi(a)$ and then deleting the first (leftmost) k symbols from this new word. (If there are fewer than k symbols, the derived string is empty.) The problem of tag is that of the behavior of the sequence of strings obtained by iterating this derived string operation. In one form it is to seek an algorithm to determine whether the empty string (and thereby termination of the process) is ever reached. In another, it is to determine for an arbitrary string whether it will be reached.

Post made the problem of tag his "major project" during the tenure of his postdoctoral fellowship at Princeton for 1920–21. The problem proved unexpectedly difficult. Post only managed to solved the problem for the case $k \leqslant 2$, $|\Sigma| \leqslant 2$, and "even this ... involved considerable labor" (1965, p. 372). Going beyond this "led to an overwhelming confusion of classes of cases, with the solution of the corresponding problem depending more and more on ... number theory. Since it had been our hope that the known difficulties of number theory would, as it were, be dissolved in the particularities of this more primitive form of mathematics, the solution of the general problem of 'tag' appeared hopeless ...." The spacial case $\Sigma = \{0, 1\}$, $k = 3$, $\phi(0) = 00$, $\phi(1) = 1101$ already proved intractible, and indeed, still seems to be open! Of course, as we now know, the problem of tag is unsolvable, Minsky (1961).

[23] One of these occurs already in his dissertation.

It was after this "frustration" that Post, in the summer of 1921, carried out his reduction to the third of his three canonical forms, and we now proceed to describe this third form. The basic idea was that of *canonical production* which we explain using contemporary terminology. Let Σ be an alphabet whose elements we call *terminals*. In addition to terminals, we use other symbols called nonterminals. Here nonterminals will be P with or without subscripts. A *canonical production* has the form

$$\left.\begin{array}{cccc} g_{10}P_{11}g_{11}P_{12} & \cdots & P_{1n_1}g_{1n_1} \\ g_{20}P_{21}g_{21}P_{22} & \cdots & P_{2n_2}g_{2n_2} \\ \vdots & \cdots & \vdots \\ g_{k0}P_{k1}g_{k1}P_{k2} & \cdots & P_{kn_k}g_{kn_k} \end{array}\right\} \rightarrow h_1P_{r_1s_1}h_2P_{r_2s_2}\cdots h_lP_{r_ls_l}h_{l+1}.$$

Here all the g's and h's belong to Σ^* and the P's are nonterminals. Moreover, the subscripts r_is_i are such that

$$1 \leqslant r_i \leqslant k, \qquad 0 \leqslant s_i \leqslant n_{r_i},$$

so that the $P_{r_is_i}$ all already appear on the left. We think of the above production as permitting a transition from k given "premises" on the left to a "conclusion" on the right, where the P's are to be thought of as replaced by particular elements of Σ^*. Now a *canonical system* is given by a finite set of *axioms* or *primitive assertions* which are themselves elements of Σ^* together with a finite set of canonical productions. Such a canonical system *generates* a subset of Σ^*, namely the set of all strings which can be obtained from the axioms by iteratively applying the canonical productions of the system. A subset of Σ^* generated in this manner by some canonical system, we may call a *canonical set*. As we have indicated, Post believed that the notion of canonical production was so general that the set of theorems of "Principia Mathematica" or any other system of logic would form a canonical set. His work had already shown that the set of theorems of the predicate calculus part of "Principia Mathematica" was a canonical set. But now he went on to prove that every canonical set could be obtained from a canonical system in a particularly simple (deceptively simple as it has turned out) *normal form*. A *normal production* is a canonical production of the special form

$$gP \rightarrow Ph.$$

A *normal system* is a canonical system with a single axiom and only normal productions, and a *normal set* is a set which can be generated by a normal system.

In 1921, Post obtained the following remarkable result which was finally published in Post (1943):

POST'S REDUCTION THEOREM. *If C is a canonical set, $C \subseteq \Sigma^*$, then there is an alphabet $\Delta \supseteq \Sigma$ and a normal set $N \subseteq \Delta^*$ such that $C = N \cap \Sigma^*$.*

Thus to solve the decision problem for the entire class of canonical systems it sufficed to do so for the much simpler class of normal systems. But this realization cast an entirely new light on Post's difficulties with "tag," For normal systems are very close indeed to the "seemingly special form of 'tag'" (1965, p. 373), and normal systems must code the full complexity of canonical systems and hence, presumably of "Principia Mathematica" in which all of classical mathematics can be embedded! "... the difficulty of 'tag' is no longer surprising" (1965, p. 401). And then, "... a fuller realization of the significance of the previous reductions led to a reversal of our entire program" (1965, p. 402).

The use of canonical systems as a method for "generating" sets of strings led Post to ask what the most general "finite-process" for generating a set of strings could be. The mathematical generality of "Principia Mathematica" together with its apparant reducibility to a canonical system led Post to conclude that with canonical systems he had already attained full generality. Combining this conclusion with Post's Reduction Theorem, Post (1965, p. 405) was led to what we may call

POST'S THESIS. *Any set of strings on an alphabet Σ which can be generated by a finite process is of the form $N \cap \Sigma^*$, where N is a normal set on an alphabet $\Delta \supseteq \Sigma$.*

By a straightforward application of the Cantor diagonal method, Post was now led to conclude "... there is no finite method which would uniformly enable us to tell of an arbitrary normal system and arbitrary [word] on the letters thereof whether that [word] is or is not generated by [that] ... system" (1965, p. 407). Post further was able to conclude that no system of symbolic logic (including "Principia Mathematica") could be complete with respect to the class of propositions that assert that given strings are generated by given normal systems (1965, p. 415).

How convincing did Post think all of this was? He wrote, "The correctness of this result is clearly entirely dependent on the trustworthiness of the analysis leading to the above generalization" (1965, p. 407), i.e. to what we have called Post's thesis. Of this analysis, Post (1965, p. 408) complained that "... it is fundamentally weak in its reliance on ... 'Principia Mathematica' ...." And, he evidently felt that the very incompleteness of "Principia Mathematica" to which the analysis led undermined its suitability as a basis for such an analysis. Post felt that "... for full generality a complete analysis would have to be made of all possible ways in which the human mind could set up finite processes for generating sequences." "Establishing ... the universal character of our characterization of an

arbitrary generated set of [strings] ... is not a matter for mathematical proof, but of psychological analysis of the mental processes involved in combinatory mathematical processes." As we have already seen, Post's "Formulation I" of 1936 also spoke of "psychological fidelity."

It is not clear whether or not Post accepted Turing's analysis as an adequate "psychological analysis" of "finite processes," but it seems clear that the qualms which Post is expressing here are very much related to the considerations which made Gödel[24] feel that "definitions of computability" which preceded Turing's were "much less suitable for our purpose."

Of course, normal sets are just what are now called recursively enumerable languages, and so Post's Thesis concerns recursively enumerable sets rather than computable functions. As such it escapes criticism from the point of view of constructivism for the same reasons that Kleene's Thesis does.

Judging by the dates mentioned by Post (1965), he worked only sporadically on these problems during the 1920s. There evidently was a burst of activity in 1924, some work in 1925, and some in 1929. In addition to efforts to carry out the desired "psychological analysis," Post mentions a program to prove the incompleteness of "Principia Mathematica" without invoking his "thesis." I have reason to believe[25] that Post lectured at Columbia University on the incompleteness of "Principia Mathematica" during the 1920s. Post's work (1965) was originally submitted to the *American Journal of Mathematics* in 1941 and was rejected with the suggestion that a shorter paper confined to what was new in 1941 be submitted. Post complied and the result is his 1943 paper.

During the 1920s Post made his living mostly by teaching in the New York City public high schools. He was plagued by recurring bouts of manic-depressive illness. At the time I was his student as an undergraduate at City College during the 1940s, he taught 16 hours per week and had no office or secretarial facilities.

7. Postscript

Although Church's Thesis is nowadays hardly to be doubted,[26] the question of what evidence is required for acceptance of a "thesis" which attempts to give a precise characterization of some intuitive concept is very

[24] Footnote in Davis (1965, p. 72).

[25] From comments I heard J. F. Ritt make to Post in 1948.

[26] We are not concerned here with attempts to distinguish "mechanical procedures" (to which Church's thesis is held to apply) from a possibly broader class of "effective procedures." See for example Wang (1974, p. 89) and Webb (1980, pp. 219–238).

much with us. Particularly relevant to contemporary theoretical computer science is the

KARP–COOK THESIS. *A set of strongs is feasibly computable if and only if it is polynomial time computable.*

This thesis is closely connected with the tantalizing $P \stackrel{?}{=} NP$ question. But is must be admitted that the evidence for the Karp–Cook Thesis is rather thin, and is far less than the evidence on the basis of which Church went ahead to assert his thesis and which Gödel regarded as insufficient. The main evidence comes to this: a large class of problems for which "feasible" algorithms have been long sought in vain are all *NP*-complete. Thus if any one of these problems had a polynomial time algorithm, all of them would. And, it is argued, it is surely likely that if *all* of these problems had *polynomial time* algorithms, some of the efforts to find "feasible" algorithms would have succeeded. But in this form the argument begs the question, retaining its force only so long as we accept the Karp–Cook Thesis. For the "feasible" algorithms which people have sought have been $O(n \log n)$ or $O(n^2)$ or at worst $O(n^3)$. There has not been extensive activity seeking $O(n^{100})$ algorithms! Thus it seems entirely possible, in the present state of knowledge, that all *NP*-complete problems have polynomial time algorithms although none has an algorithm which is feasible in any practical sense.

Note added in proof. Readers will also find of interest: CROSSLEY, J. N. (1975), Reminiscences of logicians, *Lecture Notes in Mathematics* **450** "Algebra and Logic," Springer-Verlag, 1975 New York/Berlin.

REFERENCES

CHURCH, A. (1934), The Richard paradox, *Amer. Math. Monthly*, **41**, 356–361.

CHURCH, A. (1935), An unsolvable problem of elementary number theory, Preliminary Report (abstract), *Bull. Amer. Math. Soc.* **41**, 332–333.

CHURCH, A. (1935), A proof of freedom from contradiction, *Proc. Nat. Acad. Sci. U.S.A.* **21**, 275–281.

CHURCH, A. (1936), An unsolvable problem of elementary number theory, *Amer. J. Math.* **58**, 345–363; reprinted in Davis (1965, pp. 89–107).

DAVIS, M. (1965), "The Undecidable," Raven Press, New York.

GÖDEL, K. (1931), Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I, *Monatsh. Math. Phys.* **38**, 173–198; English translation, Davis (1965, pp. 5–38).

GÖDEL, K. (1934), "On Undecidable Propositions of Formal Mathematical Systems," Institute for Advanced Study, Princeton, N.J. (mimeographed lecture notes by S. C. Kleene and J. B. Rosser); corrected and amplified in Davis (1965, pp. 41–74).

GÖDEL, K. (1936), Über die Länge der Beweisen, *Ergebnisse eines math. Kolloquiums*, 7, 23–24; English translation, Davis (1965, pp. 82–83).

GÖDEL, K. (1946), "Remarks before the Princeton Bicentennial Conference on Problems in Mathematics 1946," Davis (1965, pp. 84–88).

KLEENE, S. C. (1936a), General recursive functions of natural numbers, *Math. Ann.* **112**, 727–742; Reprinted in Davis (1965, pp. 237–253) with a correction and addenda.

KLEENE, S. C. (1936b), λ-definability and recursiveness, *Duke Math. J.* **2**, 340–353.

KLEENE, S. C. (1938), On notation for ordinal numbers, *J. Symbolic Logic* **3**, 150–155.

KLEENE, S. C. (1943), Recursive predicates and quantifiers, *Trans. Amer. Math. Soc.* **53**, 41–73; reprinted in Davis (1965, pp. 254–287) with a correction and an addendum.

KLEENE, S. C. (1981), Origins of recursive function theory, *Ann. Hist. Comput.* **3**, 52–67.

MINSKY, M. (1961), Recursive unsolvability of Post's problem of tag and other topics in the theory of Turing machines, *Ann. Math.* **74**, 437–455.

POST, E. L. (1921a), On a simple class of deductive systems, *Bull. Amer. Math. Soc.* **27** 396–397.

POST, E. L. (1921b), Introduction to a general theory of elementary propositions, *Amer. J. Math.* **43**, 163–185; reprinted in van Heijenoort, J. (1967), "From Frege to Gödel," pp. 265–283, Harvard University Press, Cambridge, Mass.

POST, E. L. (1936), Finite combinatory processes, formulation I, *J. Symbolic Logic* **1**, 103–105; reprinted in Davis (1965, pp. 289–291).

POST, E. L. (1947), Recursive unsolvability of a problem of Thue, *J. Symbolic Logic* **12**, 293–303; reprinted in Davis (1965, pp. 305–337).

POST, E. L. (1943), Formal reductions of the general combinatorial decision poblem, *Amer. J. Math.* **65**, 197–215.

POST, E. L. (1965), Absolutely unsolvable problems and relatively undecidable propositions. Account of an anticipation, *in* Davis (1965, pp. 340–433).

ROGERS, H., JR. (1967), "Theory of Recursive Functions and Effective Calculability," McGraw–Hill, New York.

TURING, A. M. (1936–1937), On computable numbers, with an application to the Entscheidungsproblem, *Proc. London Math. Soc.* (*2*) **42**, 230–265; corrections (1937) **43**, 544–546; reprinted in Davis (1965, pp. 116–154).

TURING, A. M. (1939), Systems of logic based on ordinals, *Proc. London Math. Soc.* (*2*) **45**, 161–228; reprinted in Davis (1965, pp. 155–222).

WANG, H. (1974), "From Mathematics to Philosophy," Humanities Press, New York.

WEBB, J. C. (1980), "Mechanism, Mentalism, and Metamathematics," Reidel, Dordrecht, The Netherlands.