Turing’s Analysis of Computation - Part 2 - Reception

In this chapter we discuss the reception of Turing’s 1936 paper discussed in the previous chapter. We shall examine several key figures that might be supposed to have been influenced by it: Post, Church, Gödel, Hilbert & Bernays, von Neumann, Scholz, Kleene, Wittgenstein, Newman. This discussion will end circa 1956, as by then Turing’s results become well enough known to include in textbooks shortly there after (e.g. Davis 1958).

Post

We begin our study in this chapter with Post, as Post was enthusiastic about Turing’s results. In fact, he even wrote (1994 [1944], pp. 462):

> “Recent developments of symbolic logic have considerable importance for mathematics both with respect to its philosophy and practice. That mathematicians generally are oblivious to the importance of this work of Gödel, Church, Turing, Kleene, Rosser and others as it affects the subject of their own interest is in part due to the forbidding, diverse and alien formalisms in which this work is embodied. Yet, without such formalism, this pioneering work would lose most of its cogency. But apart from the question of importance, these formalisms bring to mathematics a new and precise mathematical concept, that of the general recursive function of Herband-Gödel-Kleene, or its proved equivalents in the developments of Church and Turing. It is the purpose of this lecture to demonstrate by example that this concept admits of development into a mathematical theory much as the group concept has been developed into a theory of groups.”

From this we can extract several features of Turing’s analysis that Post found congenial. First of all, Post tells us that Turing’s paper was important because it involved a formalism. Next, Post remarks that that something new has been created. Turing’s development is put (even **shown to be**) on a par with those of the other seminal works in the field. Furthermore, Post thinks that these developments are so important that he compares them to the development of the theory of groups - a very important praise, given that Post himself had worked (1994 [1940]) in group theory.

Post also cites Turing in several other important works. In his (1994) manuscript, Post compares his own approaches to that of Gödel, Turing and Church. These should be kept in mind, especially when we discuss von Neumann, below. He writes (pp. 377, italics in original):

> “Apart from the indirect rôle the present paper may play as a different, if imperfect, pair of lenses with which to view recent developments, it has a direct contribution to make to present day literature in adding still another precise formulation to the list of general recursiveness, λ-definability, computability. Where in these formulations the informal basic idea is that of effective calculability, our own is that of a *generated set*. This derives from the idea of a symbolic logic rather than that of an algorithm, and may be described by saying that each member of the set is at some time generated by the continued application of a given method, while that method will at no time yield an individual on the primitives of the set which is not in the set.”

Post points out that Turing (and the others) has analyzed the notion of an algorithm, and compares it to his own work in what he takes to be symbolic logic. Needless to say, this way of understanding Turing’s work was what many were to pick upon. However, Post may be over selling the difference; as we know (now), a Turing computor can generate a set after a fashion (see also Epstein and Carinelli 2000, pp. 140-142).

Further, Post goes on to criticize Turing’s presentation. As Popper is reported to have said repeatedly, taking someone’s ideas seriously enough to want to refute them is a sign that one regards them as important. By Popper’s lights, Post must have thought Turing’s work important. Post distinguishes between Turing’s analysis, which analyzes all possible finite processes and methods for setting up finite processes, Post’s description of his own contribution.

Post also addresses some of the assumptions of Turing’s analysis. He discusses Turing’s assumption of the finitude of mental states in this paper as well. Post claims that his analysis of normal systems presupposes Turing’s assumption. He tells the reader that his conclusions in the paper follow more strictly if this assumption holds up under criticism. These conclusions have to do with the mechanization of mathematical activity. Post seems[1] to defend the (later articulated) Lucas-Penrose idea that the limitative results (such as Turing’s) show that (1994, pp. 429):

[1] Sieg (2003) has pointed out how thorny this paper of Post’s is. The current remarks should be taken *cum grano salis*. They will not be as important as some of the other work in this section in what follows anyway.

> “the logical process is essentially creative”

and that:

> “We see that a *machine* would never give a complete logic; for once the machine is made we could prove a theorem it does not prove.”

Post’s remarks discuss the connection between real numbers and creativity. He thinks, therefore, that Turing’s analysis of the nature of computation only works on the finiteness assumption because if he postulated essential continuity in certain aspects of the computor, the analysis would not go through. It is not clear, however, a continuous thing is necessarily creative. As we shall discuss in the sequel, Post is overlooking the possibility that there could in principle be continuous machines.

In another paper (1994 [1946]), Post made use of Turing’s analysis to solve an important problem, that of a particular question about undecidability in a Thue system. He performs a reduction necessary to prove the unsolvability in question by means of a Turing machine, rehearsing a version of Turing’s description of such. However, Post does find fault with Turing’s analysis here as well (see pp. 506), and in his appendix elaborates his criticism. There he suggests several technical changes to streamline things and perform a better analysis.

As it happens, this influence was later to go the other way. Turing’s last published paper (1992 [1954]) was to use Post’s system to perform its analysis, thus showing a great mutual affinity for each other’s work.

Another way to evaluate Turing’s importance for Post is to remark on his letters to Gödel that mention him in passing. Post wrote (see Post forthcoming [1938]) to Gödel concerning a conversation that they had at the meeting of the American Mathematical Society. In this particular letter (one of three) Post discusses the significance and scope of Gödel’s work and relates both it and his own work to that of Turing. Post describes Turing’s paper as a way of (pp. 171) “analyzing all finite processes of the human mind.”

Post also considered the status of what became to be known as the Church-Turing thesis, the statement that will become important in the sequel. In his 1936 paper, he tells the reader that his identification of effective calculability with his own system (one which is **formally** similar to Turing's) should be taken as a working hypothesis. Although he was unaware of Turing's results when this paper was written, we can take the following remark (in light of the above as expressing how he felt about this hypothesis:

> "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's remark about this identification not being a definition leads us immediately to Church's reception of Turing's paper.

Church

Post's remark concerning definitions is addressed in part to Church's own statements (in, e.g. 1936) of what came to be known as the Church-Turing Thesis. Here he **defines** effective calculability in terms of recursive function of positive integers (see his footnote #3 on page 90) or λ-definable function of positive integers. (Note, however, that Turing's elucidation is more general in a certain way: Turing allows for [restricted] real numbers as well as other [in modern terminology] data types. This will become important in the sequel. There we will analyze the question of representation, which can be phrased loosely as: "if we code something as integers, and reduce our problem to one over the integers, how are we to understand this reduction?")

Church wrote reviews of both Post's 1936 paper and Turing's (1937a, 1937b). Here he continued to maintain (against Post) his view of the putative thesis as a definition. He tells us that it is a definition as "effectiveness" in the ordinary sense has not been given an exact definition, and so the hypothesis has no exact meaning. He claims that the the hypothesis is eliminable if we define effectiveness as computability by an arbitrary machine with appropriate finiteness conditions.

It is important to note that it is possible that Church and Post had two differing notions on how to read (for example) Turing's paper and how to understand definitions. A brief digression here (see also Epstein and Carinelli 2000, pp. 226-7 for more on this theme) will explain this.

In pure mathematics, we often introduce notational definitions. For instance, in single variable calculus, we may define the derivative of a function f with respect to a variable x as follows:

$$\frac{df}{dx} =_{df} \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Rather than write $\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$ everywhere, we can simply write $\frac{df}{dx}$ (or one of several even shorter notational variants). It is safe to say that Church did not take himself and Turing as having defined "effective calculability" in **that** way.

Another kind of definition sometimes spoken of in mathematics are the "implicit definitions" found in an axiomatized theory. For instance, a set (or the membership relation) just is what is characterized by (e.g.) ZFC. It seems likely that this usage of "definition" is what Church had in mind. That is, Church would say if asked what the features of "effectively calculable" are, he would tell you to study either the postulates of the lambda calculus, or the recursive functions, etc.

However, there is a feature of **Turing's** analysis at least (and, given this possibility, likely Post thought of himself as doing much the same) that goes beyond the definition elucidated by an "axiomatic" system. (This can be explicit, like in the postulates for the general recursive functions, or implicit, like of "set" or "membership" in ZFC.)

This is the appeal to **non-mathematical** things, events, processes, etc. Turing's analysis is not like the definition of "set" implicit in ZFC because it appeals to notions which are not captured in the formalism. If one is a mathematical fictionalist, these are factual items; if a Platonist like Gödel, these are real rather than ideal objects, etc. This in turn makes Turing's paper similar to Newton's *Principia* or another piece of exact (though not necessarily axiomatic) natural science. Here, Newton's laws are not definitions (e.g. of "force") but hypotheses about the interrelations of the primitives. We shall return to this theme later[2] .

<u>Gödel</u>

Gödel was quick to connect Turing's work with his own. He considers Turing's work as a successful analysis of "mechanical procedure." This allows Gödel to remark (1995 [193?], pp. 166) that Turing's work allows statement of his own results in greater generality. Gödel tells us that a rule of inference (as used in a formal system) is a mechanical procedure with a well specified applicability and result. He also tells us his (1931) was limited in its conclusions because it had no rigorous specification of the notion of mechanical procedure, the result he takes Turing to have obtained. Gödel further remarks (pp. 168) that the definition of computable that he uses in this paper is "really correct" because of results "established beyond any doubt" by Turing. He tells us that this is so because there is no (mechanical?) procedure for determining whether a series of postulates actually defines a function. Continuing, and interestingly for future use of Turing's results, he says that the computable functions defined in this manner are those calculable by a finite **machine**. This machine is said both to have a finite number of parts and a finite running time.

Gödel later was to state unequivocally that Turing had analyzed "mechanical procedure" in a satisfactory way (See Gödel 1964, pp.

[2] The present author is of the opinion that the appeal to non-mathematical items in this case make Turing's work a piece of exact metaphysics. But that is another story for another time.

72). For someone with as exacting standards of correctness as Gödel, this is high praise indeed.

However, Gödel’s understanding of what Turing was up to in his paper is perhaps a little strange. As he told Wang (1974), he took Turing as arguing for the claim (pp. 325):

> “... that mental procedures cannot carry any farther than mechanical procedures.”

This is probably a misrepresentation of what Turing was trying to do, though this viewpoint has been very common since then. Nevertheless, Gödel will discuss what going beyond the Turing formulation would involve in this light. This we shall examine in chapters three and four.

Hilbert & Bernays

Hilbert and Bernays in their 1970 (1939) make two references to Turing. The first is in the context of calculable functions (pp. 356):

> “Bei den Kriterien der Widerlegbarkeit, die wir aus dem Herbrandschen Satz entnommen haben, wurde der Allgemeinbegriff der berechenbaren Funktion benutzt.”

Hilbert and Bernays footnote this remark with the reference to Turing (and to Post, Church and Kleene), suggesting that they take the notion of calculability to have been either defined or elucidated by Turing (and the others).

The second remark, equally brief, comes later (pp. 433). There they make reference to what they call Turing’s theory of machines bearing his name (very loose translation mine). This is in the context of a “impossibility of the general solution to the decision problem(s)” (again loose translation mine) as one might expect. There is no direct development or discussion of Turing’s ideas in either of these places.

Kleene

Kleene makes reference to Turing in his 1964 (1943) discussion of algorithmic theories. These are brought up in the context of incompleteness theorems in number theory. He asks if an algorithmic theory can give effective means for deciding the truth of propositions. He rephrases this question in more Turingesque terms by asking whether the decision procedure terminates and gives an affirmative answer.

However, he claims the notion of well defined procedure, or of effective decidability and indeed, effective computability is somewhat subjective. He then gives a version of Church's thesis which summarizes these and other considerations; it is here he invokes Turing. Turing is said to implicitly invoke it in his description of "computing machines" (Kleene's phrase). This is stated as (pp. 274):

> "Thesis 1. Every effective calculable function (effectively decidable predicate) is general recursive."

He points out that to the extent that we have a pretheoretic notion of "effectively decidable" this in fact a hypothesis (cf. the debate between Church and Post, above). He then points to an earlier paper (1938) where he refers to evidence for this:

> "This notion of effectiveness appears, on the following evidence, to be general. A variety of particular effective functions and classes of effective functions (selected with the intention of exhausting known types) have been found to be recursive. Two other notions, with the same heuristic property, have been proved equivalent to the present one, viz., Church-Kleene λ-definability and Turing computability. Turing's formulation comprises the functions computable by machines."

Again, Turing is taken to have analyzed **machine** computation.

Kleene's influential book *Introduction to Metamathematics* (1952) also devoted a substantial number of pages to discussing Turing's work and the Church-Turing thesis. However, the first remark of Kleene's that we must note is on pp. 64-65 of his text. As shall be discussed further in the sequel, finitism (understood in various ways) is vital to the Church-Turing thesis and to what Turing proved. In this passage, Kleene is explaining the scope and tools of metamathematics. He points out (correctly?) that metamathematics cannot deal with non-finitary interpretations of classical mathematics because it only uses finitary tools. The conclusion, therefore, for our purposes, is: if Turing is taken to be doing metamathematics (and of course that Kleene is correct), then the finitism of Turing’s paper is strictly necessary.

Next of interest (pp. 136-7) is Kleene’s discussion of decision procedures, which he equates with decision methods and algorithms. He emphasizes that these only deal with finite objects - another example of finitism. It is then only much later in his discussion of the “generalized Gödel theorem” (pp. 300-301) that we first encounter Turing’s name per se in this work.

It is in the context on what Kleene calls “Thesis I.” - that (pp. 301):

> “Every effectively caclulable function (effectively decidable predicate) is general recursive.”

Kleene considers this thesis to be implicit also in Turing and Post (1936), despite it having been proposed by Church (in his 1936). Kleene then proceeds to give some consequences of adopting the thesis: his theorem XII on the the lack of decision procedure for two predicates. In the following chapter of the monograph, he gives the evidence he feels has accumulated in favour of the thesis. This evidence includes the fact that all (intuitively) effectively calculable function examined has proved to be recursive, that methods for transforming such functions into general recursive functions are very robust, that methods that might be considered to bring one outside the class do not work effectively. The latter is important, as it includes explicitly the forms of the Cantor diagonal argument.

Kleene then mentions another class of evidence. In particular, he mentions that general recursiveness, λ-definability, computability (where this is taking to be the notion of Turing’s paper and Post’s 1936), normal systems, etc. are equivalent (coextensive). Kleene also mentions that many of these notions are stable - modifying them slightly yields the same equivalence.

Next, and most interesting for our concern, is that Kleene considers the fact that Turing considered a computing machine in his approach. Further, he points out a difference in Turing’s approach that, so to speak, checks the other ones. This is that (pp. 321):

> “Turing’s notion is thus the result of a direct attempt to formulate mathematically the notion of effective calculability, while the other notions arose differently and were afterwards identified with effective calculability.”

Kleene considers Turing’s formulation to be an independent statement of Church’s thesis. Then in chapter XIII, Kleene devote his attention to the computable functions, emphasizing the atomic acts for which any conceivable act a computer may do may be analyzed as a succession of said operations. He then gives a detailed and fair presentation of the “Turing machine” approach, considering (once again) that Post’s approach was similar. Then, within this chapter, Kleene introduces what he labels “Turing’s thesis”, that (pp. 376):

> “[...] every function which would naturally regarded as computable is computable under his definition, i.e. by one of his machines [...]”

Kleene (via theorem XXX of his book) proves that this thesis is equivalent to Church’s thesis. Nevertheless, he considers separate evidence for it. In particular, he tries to show us that any acts a computor could perform are analyzable into the atomic acts of a Turing machine. He cites Turing’s discreteness and finiteness conditions to this end.

The final part of Kleene’s book for us to consider occurs in his discussion of Post’s solving of the word problem for semi-groups. As this is an example of a undecidable decision problem, Kleene first gives examples of such. He points out that the problems first proved to be such were problems arising out of mathematical notions such as l-definability, general recursiveness, or computability. Crucially, he points out that these notions are identified with effectiveness **by the Church-Turing Thesis.** So here we have a statement of our central concern. Given that Kleene considers the thesis confirmable (see above concerning his amassing evidence for it), it is safe to say that Kleene felt it to be a thesis proper, and not a definition.

von Neumann

Von Neumann's references to Turing's work are largely in the context of planning actual computing devices. The first of these occurs in his important paper "The General and Logical Theory of Automata" (von Neumann 1961 [1951], pp. 288-328). Von Neumann seems to have thought that Turing's work (in particular, the universal Turing machine) described the most powerful automaton possible (pp. 314). Von Neumann recognizes that one of the keys to Turing's analysis is the finiteness of the analysis. Curiously enough, he puts this in terms of the **description** of the machine. One can thus infer that von Neumann considers any automata to be finite.

While it is also in the context of engineering-like proposals, another of von Neumann's references (1961 [1956]) to Turing's work also touches its logical importance. He states that constructive (or intutionistic) logics can be studied in terms of automata. A logical proposition gets represented as a black box with a rule of input-output. In fact, von Neumann regards the former consideration with such great importance he repeats it at least twice more in the paper.

Scholz

According to Hodges (1983, pp. 124), Heinrich Scholz wrote two postcards asking for copies of the famous paper we are discussing. However, there are no references to Turing by name in any of the work the present author was able to locate (1961a, 1961b, 1969).

Braithwaite

Braithwaite, the young philosopher of science Turing knew at Cambridge, was one of two (the other being Scholz, above) to order off-prints of Turing's paper. The present author has not been able to locate any reference to Turing in Braithwaite's work.

Wittgenstein

Hodges reports (1983) that Wittgenstein was sent a copy of Turing's paper. Both Monk (1991) and Hodges (1983, pp. 152-154) report on possible indirect responses. They report that Turing attended a few sessions of Wittgenstein’s course on the foundations of mathematics.

In Wittgenstein’s course, Turing and Wittgenstein argued over the status of a contradiction and disputed the nature of a proof. Despite the nature of rules being central to these debates, it is important to note that Turing never made use of his own work. Hodges suggests that the Turing machine formalism itself may have been useful in this regard and does not understand why Turing was reluctant to use it in his debate with Wittgenstein. Monk is not quite so sure, claiming that for Turing to use his formalization to argue against Wittgenstein would be to miss the latter’s point. In retrospect, Monk is probably right, due to Wittgenstein’s more “social” understanding of computation. It is possible that Turing picked up on this. We cannot spend any further time on this “sociality” response (as it is taking us rather far afield) except to note that Wittgenstein was to later write some (unpublished) remarks concerning the performance of calculations by machines. See, e.g., his 1994 (1967), pp. 234, italics in original:

> “If calculating looks to us like the action of a machine it is *the human being* doing the calculation that is the machine.”

The relationship between Wittgenstein’s work and Turing is rather vexed; one interpretation has been given by Shanker (1987). According to Shanker, the key feature of Wittgenstein’s reaction to Turing is summarized in his [1980 (1949)] remark (italics in original):

> “Turing’s ‘Machines’. These machines are *humans* who calculate”

Shanker’s article then attempts an exegesis of Wittgenstein’s scattered views to articulate Wittgenstein’s views on Turing’s work. While this is not the place to examine Wittgenstein in detail, Shanker is right to point out that the likely issue that Wittgenstein had in Turing’s work concerned the nature of rules, given the former’s near obsession with them. Shanker reminds us that for Wittgenstein, a rule is essentially normative (as opposed to causal [pp. 638]) and hence has an irreducibly social component.

Wittgenstein’s thought experiments concerning the calculating machines uses as black boxes or by chance are intended to illustrate this viewpoint. Since (according to Wittgenstein) to follow a rule is to be able to take part in the “language game” surrounding its use, it thus makes little sense to say that someone followed a rule mechanically. Shanker puts this slightly stronger when he says that Turing’s paper was objectionable to Wittgenstein because it mixed up logic with philosophy of mind[3].

Shanker also extracts a more directly metaphysical (though Wittgenstein would not have put it this way) way of stating another facet of the issue, one that does have direct consequences further on in the present work. While Shanker does exaggerate slightly when he says that Wittgenstein would have though that Turing’s analysis was not as sharp as it first appears, he does raise an important issue. This is the question of the atomicity of the operations the computor performs. Wittgenstein would claim that Turing’s analysis is incomplete because he hasn’t carefully scrutinized the basic operations of the computor enough. In particular, Wittgenstein would claim that in order to follow a rule we need also to be able to reflect upon it and justify past actions by reference to the rule (pp. 640-1).

Finally, according to Shanker, Wittgenstein realized that we should not conclude from the simplicity of the subrules of an algorithm that they are thereby mechanical in some relevant respect or other. Later chapters of this present work will analyze this claim as part of the discussion of the Church-Turing Thesis, so it is important to draw attention to it now.

One facet of Wittgenstein’s remarks is not paid much attention to by Shanker. He says that Turing’s point could have been made with **games**. It is not immediately clear how this would work. It is the present author’s supposition that this would connect somehow to the “sociality” concerns we have discussed previously - but there

[3] Of course, this is more or less precisely why the present author finds it **interesting**.

is very little information surrounding this conjecture either way.

Newman

Newman, Turing’s teacher, was the one who first suggested that he tackle the Entscheidungsproblem in the first place. However, the present author has not been able to locate any written remarks concerning Turing’s work in this regard.

References

Church, Alonzo. 1964 (1936). “An Unsolvable Problem of Elementary Number Theory.” Reprinted in in Davis 1964.

Church, Alonzo. 1937a. “Review of Post, 1936.” Journal of Symbolic Logic. Vol. 2, pp. 43.

Church, Alonzo. 1937b. “Review of Turing, 1936.” Journal of Symbolic Logic. Vol. 2, pp. 42-3.

Davis, Martin. 1958. *Computability and Unsolvability*. New York: McGraw-Hill Book Co.

Davis, Martin. 1964. *The Undecidable*. Hewlett: Raven Press.

Davis, Martin. (ed.) 1994. *Solvability, Provability, Definability: The Collected Works of Emil L. Post.* Boston: Birkhäuser.

Epstein, Richard and Carnielli, Walter. 2000. *Computability: Computable Functions, Logic, and the Foundations of Mathematics.* (2e) Belmont: Wadsworth/Thomson Learning.

Gödel, Kurt. 1964. “Postscriptum.” In Davis 1964.

Gödel, Kurt. 1995 (193?). “Undecidable diophantine propositions.” Reprinted in *Collected Works*, vol. III, ed. Solomon Feferman *et al.* Oxford: Oxford University Press.

Hilbert, David and Bernays, Paul. 1970. *Grundlagen der Mathematik II*. Berlin: Springer-Verlag.

Hodges, Andrew. 1983. *Alan Turing: The Enigma*. New York: Simon and Schuster.

Kleene, Stephen. 1938. “On Notation for Ordinal Numbers.” Journal of Symbolic Logic. Volume 3, Number 4.

Kleene, Stephen. 1952. *Introduction To Metamathematics*. Amsterdam: North-Holland Publishing Co.

Kleene, Stephen. 1964. 1943. "Recursive Predicates and Quantifiers." Reprinted in in Davis 1964.

Monk, Ray. 1991. *Ludwig Wittgenstein: the duty of genius*. New York: Penguin.

Post, Emil. 1994 (1936). “Finite Combinatory Processes - Formulation I.” Reprinted in Davis 1994.

Post, Emil. 1994 (1940). “Polyadic Groups.” Reprinted in Davis 1994.

Post, Emil. 1994 (1944). “Recursively Enumerable Sets of Positive Integers and their Decision Problems.” Reprinted in Davis 1994.

Post, Emil. 1994 (1946). “Recursive Unsolvability of a Problem of

Thue." Reprinted in Davis 1994.
Post, Emil. Forthcoming (1938). "Post to Gödel 2." To be reprinted in *Kurt Gödel Collected Works*, vol. IV, ed. Solomon Feferman *et al.* Oxford: Oxford University Press.
Scholz, Heinrich. 1961a. *Concise History of Logic*. New York: Philosophical Library.
Scholz, Heinrich. 1961b. *Grundzuge der mathematischen Logik*. Berlin: Springer.
Scholz, Heinrich. 1969. *Mathesis universalis: Abhandlungen zur Philosophie als strenger Wissenschaft*. Basel: Scwabe.
Shanker, Stuart. 1987. "Wittgenstein versus Turing on the Nature of Church's Thesis." Notre Dame Journal of Formal Logic, vol. 28 no. 4.
Sieg, Wilfried. 2003. Private communication.
Taub, Abraham (ed.). 1961. John von Neuman: Collected Works. Volume 5: Design of Computers, Theory of Automata and Numerical Analysis. London: Pergamon Press.
Turing, Alan. 1992. (1954). "Solvable and Unsolvable Problems" Reprinted in *Collected Works of A. M. Turing*., vol. 3. Ince, Darrel, ed.
von Neumann, John. 1961 (1951). "The General and Logical Theory of Automata." Reprinted in Taub 1961.
von Neumann, John. 1961 (1956). "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components." Reprinted in Taub 1961.
Wang, Hao. 1974. *From Mathematics to Philosophy*. London: Routledge & Kegan Paul.
Wittgenstein, Ludwig. 1980 (1949). *Remarks on the Philosophy of Psychology.* Chicago: University of Chicago Press.
Wittgenstein, Ludwig. 1994 (1967). *Remarks on the Foundations of Mathematics.* Cambridge: MIT Press.