. ASPRAY, Jr., William F., 1952-
FROM MATHEMATICAL CONSTRUCTIVITY TO COMPUTER SCIENCE: ALAN TURING, JOHN VON NEUMANN, AND THE ORIGINS OF COMPUTER SCIENCE IN MATHEMATICAL LOGIC.

The University of Wisconsin-Madison, Ph.D., 1980,
History of Science

Xerox University Microfilms, Ann Arbor, Michigan 48106

(This title card prepared by the University of Wisconsin)

PLEASE NOTE:

The negative microfilm copy of this dissertation was prepared and inspected by the school granting the degree. We are using this film without further inspection or change. If there are any questions about the film content, please write directly to the school.

UNIVERSITY MICROFILMS

FROM MATHEMATICAL CONSTRUCTIVITY TO COMPUTER SCIENCE:

ALAN TURING, JOHN VON NEUMANN, AND THE ORIGINS

OF COMPUTER SCIENCE IN MATHEMATICAL LOGIC

A thesis submitted to the Graduate School of the University of Wisconsin-Madison in partial fulfillment of the requirements for the degree of Doctor of Philosophy

by

William F. Aspray, Jr.

Degree to be awarded: December 19____ May 19____ August 19 80

Approved by Thesis Reading Committee:

Major Professor

Date of Examination

Dean, Graduate School

FROM MATHEMATICAL CONSTRUCTIVITY TO COMPUTER SCIENCE: ALAN TURING, JOHN VON NEUMANN, AND THE ORIGINS OF COMPUTER SCIENCE IN MATHEMATICAL LOGIC

BY

WILLIAM F. ASPRAY, JR.

A thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY
(History of Science)

at the

UNIVERSITY OF WISCONSIN—MADISON

1980

Acknowledgements

A number of people have helped in a variety of ways in the writing of this dissertation. The most tangible assistance has been a continuing fellowship for the writing of the dissertation from the Cahrles Babbage Institute. The CBI, especially Paul Armer and Erwin Tomash, have helped not only financially, but also through their encouragement, their constant supply of primary and secondary materials on the history of computers, and their introductions to a number of scholars in the field. This dissertation would never have been completed without their generous assistance.

I am also thankful for financial assistance during my graduate work from teaching assistantships at Wisconsin and Toronto and from fellowships at Wisconsin and Princeton.

A great deal of credit must be assigned to my advisor, Victor Hilts, for the overall direction of a dissertation in a field foreign to his research interests. The other readers on my dissertation committee have also provided valuable advice: Daniel Siegel, especially for criticism of my historical analysis, and Terry Reynolds, especially for improvements in my historical style. Others who have provided suggestions or criticisms on this work include Thomas Drucker, Brian Randell, Robert Doran, Gregory Moore, Kenneth Kunen, Charles Slavin, Helene Shapiro, and David Erbach.

I thank the Library of Congress for the chance to examine the von Neumann archives. Shirley Shen, the mathematics librarian, and John Neu, the history of science bibliographer, both at the University of Wisconsin, have been generous with their time, helping me to locate hard to find materials. I also must thank J. Barkeley Rosser and Stephen Kleene for their willingness to be interviewed on the early history of logic and computer science. Professor Kleene, in particular, has been exceedingly generous with his time, offering to answer questions and reading various drafts I have written.

Carol Voelker has provided outstanding service typing, proof-reading, and removing the most egregious of grammatical and stylistic errors in my dissertation.

Finally, I would like to make two special acknowledgements: to Professor Robert Rosenbaum of Wesleyan University and to the late Professor Kenneth O. May of the University of Toronto. Professor Rosenbaum introduced me to logic, foundations, and the history of mathematics when I was an undergraduate and encouraged me to pursue graduate work in the history of logic. Professor

May supported my research in the history of mathematics at a low point in my graduate career and directed me toward a dissertation in mathematical logic of the 1930's.

Table of Contents

Introduction

This dissertation examines the origins of computer science in that branch of mathematical logic known as recursive function theory. It argues that in response to a foundational crisis in mathematics, a new mathematical discipline of recursive functions was developed. The recursive functions were the formal characterizations of the functions one could actually compute in mathematics. Thus, the recursive functions provided the theoretical basis for computer science. In fact, two of the more important researchers in logic, Alan Turing and John von Neumann, used their experience with the mathematical theory of recursive functions in the design of physical computing machinery and in the development of theoretical computer science. The dissertation traces this movement from mathematics to computer science and focuses in particular on the work of Turing and von Neumann.

As well as being a popular account of the development of computer science and information processing in general, the dissertation should be of specific interest to the mathematician, the computer scientist, the logician, the philosopher, and the cognitive scientist. Part of the reason for this is the rather different focus of this dissertation from most work in the history of computers. Typically, scholarship in the history of computers has focused on the genealogy of computers and computer technology. It has been extremely internalist and has not been interested in the relation of computer science to other disciplines. This dissertation, however, aims to examine some of the roots of computer science in mathematics, philosophy, psychology, and physics.

The computer scientist should be interested in this dissertation because it will help him to see the roots of his field in other disciplines, particularly in mathematics. Besides the intrinsic interest of the subject, it helps the computer scientist in understanding the logical design of the early computers, which, to a large degree, has set the framework for the logical design of present computing machinery. This is especially true because the machines von Neumann helped design were crucial ancestors in the genealogy of American computing machinery. Turing's work is important, although to a lesser extent, in the development of British computing machinery.

This work should be of interest to mathematicians because it shows the various connections of computer science to mathematics. It shows, in particular, how computer theory came out of widely diverse areas of mathematics, including logic, numerical analysis, mathematical physics, and applied mathematics. It also throws light on the issue of what bearing the computer has in pure mathematical research, for von Neumann had strong and well-developed ideas on the subject.

The dissertation helps the philosopher to get an idea of the relationship between two related sciences, mathematics and computer science, one entirely mental and one partly empirical. It gives a good idea of the use of the computer as an experimental tool. This work also addresses the issues of specific philosophical importance, namely artificial intelligence and automata theory (which is concerned with questions, for example, such as what is the logical nature of self-replication). Finally, the dissertation traces the actual attempts made by mathematicians to treat the philosophical problems of a foundational crisis in their field. This provides a valuable balance to the philosophical study of intuitionism, logicism, and formalism, which is altogether too frequently studied by philosophers outside of the historical context of researchers actually trying to resolve problems so as to continue work in their field.

For a rough understanding of the dissertation, a general knowledge of mathematics and science should suffice. Certain sections do require special training if one is to follow the details as well as the general structure of the argument. The first two chapters concentrate on the mathematical background to the development of computer science. Therefore, it would help for the reader to have a knowledge of mathematics. Chapter Two is the only truly technical chapter. To follow the arguments in it in complete detail would require an undergraduate degree in mathematics and some graduate work in logic. Even for those not versed in advanced mathematics, however, it is easy to follow the general argument. The mathematically uninitiated might wish to only skim the two sections of Chapter Two on recursive functions, for these are highly technical and only tangentially connected with the remainder of the dissertation.

The third and fourth chapters concern the development of physical computing machinery. To appreciate fully these sections it would be useful to have at least rudimentary knowledge of the working of computers and computer engineering. The fifth chapter provides a retrospective of the various fields that developed after the second world war which were based on the concept of information processing. Thus, a general knowledge of philosophy, physics, physiology, and psychology would be of use in understanding the backgrounds to these new interdisciplinary fields. Chapters Six and Seven are primarily self-contained, and no special knowledge is required.

The dissertation consists of three major sections. The first section describes the mathematical background to the development of the mathematical theory of computing. The first chapter argues that, due to the use of abstraction and infinity, there was a foundational crisis in mathematics around the turn of this century. The crisis centered around counter-intuitive and meaningless results, unwarranted assumptions of powerful principles, and outright contradictions in mathematics. The chapter describes one of the major answers to the crisis, a conservative, philosophically based reaction, known as constructivism, which attempted to reformulate the foundations of mathematics by constructing every object of mathematical interest from primitively accepted objects. The chapter concludes with an assessment of the failure of the earliest of the constructivist attempts.

Chapter Two describes a later, and more sophisticated, attempt by David Hilbert to use constructivist techniques to provide a foundation for mathematics. Unfortunately, using the same mathematical technique, Kurt Gödel was able to show that Hilbert's program was also doomed to failure. The situation was not entirely bleak, however, for Hilbert's and Gödel's techniques provided the development of a new mathematical discipline, recursive function theory, which provided a precise mathematical characterization of those functions which the constructivists thought were acceptable in mathematics. Many different characterizations were given by the mathematicians of the recursive functions. However, the first one to be both precise enough to be used in formal mathematics and intuitive to convince the mathematical community that it characterized the constructable functions was given by Alan Turing. His characterization was in the form of a theoretical machine which would carry out the computations of the functions. Turing perfected his theoretical machines while completing a graduate degree at Princeton University. There he met the famous mathematician and logician, John von Neumann. The two of them discussed the possibilities of putting Turing's characterization to work in the development of physical computing machinery. Unfortunately, the war intervened and their plans had to be put aside. The last half of Chapter Two describes the technical details of the development of recursive function theory, explains the importance of Turing's characterization for mathematics, and concludes with a description of the first discussions of making Turing's work useful for physical computing as well.

The second section of the dissertation examines the attempts by Turing and von Neumann to work Turing's logical idea of an idealized computing machine into the design of modern computing machinery. Chapter Three discusses Turing's work on the design of physical computing equipment during the war at Bletchley Park on the Colossus and Robinson series machines, and his work after the war at the National Physical Laboratory on Pilot ACE and at the University of Manchester on MANIAC. Chapter Four discusses von Neumann's work on the design of computing equipment at the

University of Pennsylvania on ENIAC and EDVAC and on the Institute for Advanced Study computer. The emphasis is on the logical design of the computers. However, other technical details are also considered.

The third section of the dissertation examines the development of theoretical computer science and the more general development of the theoretical study of information processing. Chapter Five describes the general development of new interdisciplinary sciences of communication theory, cybernetics, physiological psychology, automata theory, and artificial intelligence. It argues that the developments of these fields were related and that they stemmed from war-related work on the concept of information processing. The last two chapters detail Turing's and von Neumann's contribution to the subject. Chapter Six describes Turing's work on artificial intelligence. Chapter Seven discusses von Neumann's work on automata theory. These three chapters show that the ideas from logic were useful for the development of theoretical computer science, which studies this concept of information processing, as well as the basis for the design of physical computing equipment.

Chapter One: Brouwer, Constructivity, and the Foundational Crisis Facing Twentieth Century Mathematics

L. E. J. Brouwer completed his doctoral dissertation at the University of Amsterdam in 1907. In this famous dissertation he introduced a radical new philosophy of mathematics, known today as intuitionism. This philosophy was a reaction to what Brouwer saw as a foundational crisis in mathematics which had developed over the three decades prior to his dissertation. The cause, he believed, was the growth of abstract mathematics and the use of actual infinities in mathematical practice. The effect, he perceived, was meaningless and contradictory results in the new mathematics.

Brouwer's intuitionist system is the prime twentieth century example of a conservative, philosophical reaction, known as constructivism, which has arisen repeatedly in the history of mathematics whenever it has appeared that the foundation of mathematics was not secure. The purpose of this chapter is to examine the origins and nature of this twentieth century crisis as well as to describe Brouwer's and others' constructivist attempts to rectify this problem. Subsequent chapters will show how these constructivist activities led to the development of a theoretical framework for computer science. However, in order to understand fully the rationale of Brouwer's position, the rise of abstraction and infinity must first be traced.

Axiomatics, Abstraction, Foundational Crisis

Although the rise of abstraction in mathematics is a complex issue, the main stimulus seems to have been the development of non-Euclidean geometry. The existence of non-Euclidean geometries equal in stature and contradictory to Euclidean geometry obliged mathematicians to reassess their beliefs on the nature of mathematics and its relation to the physical world. While the earliest propositions were formulated at the beginning of the nineteenth century, the importance of non-Euclidean geometry was not generally recognized until the 1860's because Bolyai and Lobachevsky's work was overshadowed by projective geometry in the 1830's and 1840's and because Gauss' work on the subject was published only posthumously in 1855.

The existence of non-Euclidean geometries convinced students of mathematics that Euclidean geometry could not be equated with the geometry of physical space. The fact that two consistent, workable mathematical systems which were mutually contradictory could exist contemporaneously indicated that the foundation for mathematics was not to be found in the physical world. What then would provide the requisite foundation? The answer for mathematics was a heavy reliance on axiomatics. If one could axiomatize a mathematical system, then one could check the foundation of the entire system by merely checking the few axioms and logical principles of reasoning for the system.

This approach was taken with respect to Euclidean geometry. Euclid had assumed the truth of the parallel postulate;[1] whereas the non-Euclidean geometers had shown that it was independent of the other axioms of geometry and was thus only relatively true or false, depending on the system within which one was working. If the problems of the parallel postulate had gone untreated by Euclid, what other logical gaps could be found in The Elements? Thereafter, all of Euclid's proofs were studied with critical acumen for logical gaps. The position taken by mathematicians at the turn of the century towards The Elements was summarized by

[1] Euclid's parallel postulate (Postulate Five) postulates: That, if a straight line falling on two straight lines makes the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which the angles are less than the two right angles. Elements, Book I. The issue amounts to how many lines may be parallel to a given line through a given point not on that line. Euclid assumes, incorrectly, that the number must be one. It may, in fact, be any whole number or infinitely many, depending on the system in which one is working.

Bertrand Russell:[2]

> It is customary . . . to defend him [Euclid] on the ground that his logical excellence is transcendent, and affords an envaluable training to the youthful powers of reasoning. This claim, however, vanishes on a close inspection. His definitions do not always define, his axioms are not always indemonstrable, his demonstrations require many axioms of which he is quite unconscious. A valid proof retains its demonstrative force when no figure is drawn, but very many of Euclid's earlier proofs fail before this test. . . . The value of his work as a masterpiece of logic has been very grossly exaggerated.

One group of mathematicians, led by Pasch, showed how Euclidean geometry could be reformulated in terms of projective geometry.[3] Another group, of more interest here, led by Peano and Hilbert, developed the axiomatic approach in order to provide a secure foundation for Euclidean geometry.[4] For the most part, this approach was successful. Scores of different axiomatic systems were propounded, each beginning with slightly different primitive concepts. The culmination of the movement was

[2] Bertrand Russell, Foundations of Geometry, 1902, excerpted from the quotation in Morris Kline, Mathematical Thought from Ancient to Modern Times, p. 1005.

[3] For details see M. Kline, op. cit., Chapter 42, and M. Pasch and M. Dehn, Vorlesungen über neuere Geometrie, 2nd ed., 1926.

[4] See M. Kline, op. cit., Chapter 42 for an overview. Peano's work can be found in his Opere Scelte, 3 vol., 1957-59.

undoubtedly Hilbert's axiomation of geometry, first given in 1899,[5] but improved throughout his life.

Of more importance here is the growth of an axiomatic movement out of the attempts to provide an axiomatic foundation for Euclidean geometry. Poincaré pointed out[6] that these axiomatic systems must be consistent. Peano discussed[7] the condition of independence and its desirability for the axiomatic systems for Euclidean geometry. Veblen developed[8] the conditions of categoricalness and completeness for an axiomatic system. These conditions, first developed as properties required for the axiomatic systems for Euclidean geometry, soon became the standards for all axiomatic systems. Soon axiomatics were being used in almost every field of mathematics.[9] Hilbert was the

[5] See the various versions of D. Hilbert, Grundlagen der Geometrie, 1st ed. (1899) through 7th ed. (1930).

[6] H. Poincare, "On the Foundations of Geometry," Monist, 1898, vol. 9, p. 38.

[7] See G. Peano, Arithmetices Principia (1889) and Sui fondamenti della geometria (1894).

[8] See O. Veblen, "A System of Axioms for Geometry," Transactions of the American Mathematical Society 5 (1904), pp. 343-384.

[9] See M. Kline, op. cit., Ch. 42-43 for an overview.

champion of this movement:[10]

> Everything that can be the object of mathematical thinking, as soon as the erection of a theory is ripe, falls into the axiomatic method and thereby directly into mathematics. By pressing to ever deeper layers of axioms . . . we can obtain deeper insights into scientific thinking and learn the unity of our knowledge. Especially by virtue of the axiomatic method mathematics appears called upon to play a leading role in all knowledge.

Not only were axiomatics used in every field of mathematics to provide a foundation for the work already done. They were also used to explore new problems by slightly varying the axiom system of the classical theory and by then contrasting the resulting system with the classical theory. This was radically different from classical mathematics, where theory revolved around the study of the structure and properties of known objects such as the real line. Mathematics had been led even further from physical reality than before. The effect was a mathematics which studied abstract structures created by man--not given by the physical world. The mathematician was now free

[10] D. Hilbert, "Axiomatisches Denken," Mathematische Annalen, 78 (1918), pp. 405-415. Also in Hilbert's Gesammelte Abhandlungen, 3, pp. 145-156. Quoted as translated in Kline, op. cit., p. 1027.

to create any mathematical system he pleased--provided that it was internally consistent. Hilbert emphasised this point:[11]

> Indeed the axiomatic method is and remains the one suitable and indispensable aid to the spirit of every exact investigation no matter in what domain; it is logically unassailable and at the same time fruitful; it guarantees thereby complete freedom of investigation.

The same point is made by Cantor[12] even more directly:

> Mathematics is entirely free in its development and its concepts are restricted only by the necessity of being noncontradictory and coordinated to concepts previously introduced by precise definitions. . . . The essence of mathematics lies in its freedom.

Notice that both emphasised the freedom of mathematics--not its abstraction. However, the emphasis on the axioms themselves and only secondarily on what could be derived from them, the loss of a physical interpretation for mathematics, and the loss of a distinguished set of axioms (such as there had been for Euclidean geometry) which meant that one had no reason to study the details of any particular system in depth--all of these led mathematics to become more abstract as well as more free.

[11] D. Hilbert, "Neubegründung der Mathematik, Erste Mitteilung," Abhandlungen Mathematische Seminar der Hamburger Universität, 1 (1922) pp. 157-177. Also in Hilbert's Gesammelte Abhandlungen, 3, 157-177. Excerpted from the quotation as translated in M. Kline, op. cit., p. 1027. My emphasis.

[12] G. Cantor, "Ueber unendliche, lineare Punktmannichfaltigkeiten," Mathematische Annalen, 21 (1883), pp. 563-564. Also in Cantor, Gesammelte Abhandlungen, p. 182. Quoted as translated in M. Kline, op. cit., p. 1031.

Although every branch of mathematics was influenced by the axiomatic approach and the consequent increase in freedom and abstraction, one of the most significant changes was the development of infinite sets and transfinite numbers by Cantor. Actually, Cantor was not an axiomatist. However, as the above quotation suggests, he utilized the freedom of the new mathematical approach to introduce infinite sets of points into mathematics, where previously only finite sets of points[13] had been considered as legitimate, and to extend the system of natural numbers to include all of the transfinite numbers. The procedure in both cases involved abstracting properties from finite cases which could be used for definition in infinite cases. This is especially clear in the definitions Cantor gave of ordinal and cardinal numbers:

> Definition: To every ordered set M there corresponds a definite "order type," which we shall designate by $\overline{M}$; by this we mean the general concept which arises

[13] Strictly speaking, this is not correct. Completed infinite sets of points, as given by a finite definition, were certainly accepted in mathematics. One could talk about the integers, the rationals, or the reals, for example. One could speak in general, for example, about all the limit points of a set. But, if they were infinite in number, nothing could be said about all the individual points at once, except what was given by the definition of the function. Certainly nothing could be said about exactly how many there were. Exactly what could and could not be said about infinite point sets before Cantor is hard to characterize.

> from M if we only abstract from the character of its elements m but retain the order in which elements occur in M .[14]
>
> Definition: We call the "power" or "cardinal number" of M that concept which is derived from the set M by the help of our active powers of thought. The concept is abstracted from the character of the various elements m and from the order in which they occur in M .[15]

Cantor used the symbol $\overline{\overline{M}}$ to designate the cardinality of M , with the two strokes representing the two acts of abstraction required to form the cardinal from the set M . Order and size, the two properties measured by ordinality and cardinality, were well-known properties of finite sets. In fact, Cantor even utilized his new concepts to explain the difference between finite and infinite sets. For finite sets M , N , $\overline{\overline{M}} = \overline{\overline{N}}$ implies $\overline{M} = \overline{N}$; but for infinite sets P , Q , $\overline{\overline{P}} = \overline{\overline{Q}}$ does not necessarily imply $\overline{P} = \overline{Q}$.

Although Cantor capitalized on the freedom of the new mathematics to develop his theory of infinite point sets, his theory was in touch with many of the classical branches of

[14] G. Cantor, "Beiträge zur Begrundung der transfiniten Mengenlehrer," Part I, Mathematische Annalen 46 (1895), p. 297. Also in Contributions to the Founding of the Theory of Transfinite Numbers, ed. and trans. by P. E. B. Jourdain (1915), p. 111.

[15] Ibid., p. 282. Also in Jourdain, op. cit., p. 86.

mathematics. Cantor had first attended the problem of infinite point sets in trying to establish a uniqueness theorem for trigonometric representations of functions.[16] His theory of infinite point sets answered the uniqueness question for all but a small minority of cases. In fact, the theory also provided answers to two other well-known problems from classical nineteenth century analysis. He was able to provide the first simple and general proof that transcendental numbers do exist.[17] He was also able to prove that all the finite dimensional Euclidean spaces were of the same cardinality.[18] Besides these proofs and the development of an arithmetic of transfinite numbers, Cantor's work introduced a number of new concepts that were used extensively in topology and analysis, including: well-order, closure, density, limit points, perfect sets.

At first other mathematicians were hesitant toward Cantor's theory because it broke with a two millenia tradition which eschewed the use of actual infinities, because of Cantor's

[16] See J. Dauben, op. cit., Ch. 1, 2 for details.

[17] See G. Cantor, "Uber eine eigenschaft des Inbegriffes aller reellen algebraischen Zahlen," Journal für die reine und angewandte Mathematik 77, pp. 258-262. Also in Cantor's Gesammelte Abhandlungen mathematischen und philosophischen Inhalts, Ed. by E. Zermelo, Berlin, 1932, pp. 115-118.

[18] Cantor's results were first published in a series of letters to Dedekind throughout the 1870's. Their letters are discussed in J. Dauben, op. cit., Ch. 3.

difficult personality, and because of Kronecker's vehement reaction against Cantor's theory. However, as the theory became more familiar and the power of the methods became more apparent, mathematicians began to use Cantor's theory to solve problems in their own fields. Schroeder, Bernstein and later Schoenflier and Zermelo developed set theory.[19] Poincaré was the first to use Cantor's theory in a completely different field when he applied it to solve some problems in differential equations in the 1880's. This work was also utilized in the early development of functional analysis, as can be seen in the work of Arzela and Volterra. Hilbert[20] adopted and extended to a theory of infinite spaces material from a paper by Fredholm[21] on integral equations which utilized Cantor's results. Many of the new concepts

[19] For a report on the early development of set theory, see A. Schoenflies, Entwickelung der Mengenlehre und Ihrer Anwendungen. Umarbeitung des im VIII Bander der Jahresbericht der Deutschen Mathematiker-Vereinigung erstatteten Berichts. Erste Hälfte: Allgemeine Theorie der Unendlichen Mengen und Theorie der Punktmengen. Leipzig, 1913. Also see J. Dauben, op. cit., Ch. 11.

[20] Hilbert did this work in a series of papers written between 1904 and 1910 for Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen. These papers were reproduced in his book, Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen (1912).

[21] I. Fredholm, "Sur une classe d'equations fonctionelles," Acta Mathematica, 27 (1903), pp. 365-390.

introduced by Cantor were developed into point set topology between 1880 and 1920.[22] These same concepts were used to advantage by Lebesque, Baire, and especially Borel[23] in the development of measure and integration theory. Thus, soon Cantor's theory of infinite sets had been incorporated into a number of important fields of mathematics. This was the Cantorian paradise to which Hilbert was to refer.[24]

The growth of abstraction and the employment of infinity undoubtedly added to the richness of mathematics. Many new fields, such as topology and set theory, opened up, and many old problems, such as the uniqueness of Fourier representations of functions, were solved. Unfortunately, there was a negative aspect to these developments as well. It was to these negative aspects in the

[22] See J. H. Manheim, Genesis of Point Set Topology, 1964.

[23] See H. Lebesque, La Mesure des grandeurs. Published originally as "Sur la mesure des grandeurs," Enseignement mathematique 31-34 (1933-36). Later reprinted as the first of the Monographies de l'Enseignement Mathematique. Translated by Kenneth O. May in Measure and the Integral, 1966. Also Leçons sur l'intégration et la recherche des fonctions primitives (1904). Also see T. Hawkins, Lebesque's Theory of Integration, 1970. Also see R. Baire, "Sur les fonctiones de variables réeles," Milan, 1899, doctoral dissertation, Faculté des Sciences, Paris, no. 977.

[24] D. Hilbert, "Uber das Unendliche," Mathematische Annalen, 95 (1926), p. 170. Also in Grundlagen der Geometrie, 7th ed., 1930, p. 274.

new mathematics that Brouwer reacted in the development of his intuitionistic system.

The first objection to abstract mathematics focused on the charge that it was strongly counter-intuitive, in the sense that it contradicted the intuitions of practitioners of the discipline gained working with the classical results in their field. For example, there was dissatisfaction over Weierstrass' (1872) example of an everywhere continuous, nowhere differentiable function[25] and over Peano's (1890) space-filling curve[26] because both contradicted all of the analyst's past experience. As both of the curves were constructed by infinite processes,[27] there was at first suspicion and consequently close scrutiny of the processes that were used in constructing these examples. A similar concern arose over Cantor's theory of transfinite numbers. Other mathematicians believed that infinite numbers were either meaningless of self-contradictory. However, in all three cases the suspicion passed as the objects became more familiar and so

[25] In an unpublished lecture. See J. Mannheim, op. cit., for details.

[26] G. Peano, Mathematische Annalen, 36 (1980), pp. 157-160). Also in G. Peano, Opere Scelte, 1, pp. 110-115.

[27] Peano's curve was the limiting curve f approximated by finitely constructed approximating curves f_n, for n a natural number. Weierstrass' function was defined by an infinite series.

became part of their new intuition of mathematics. Thus, it seems that the objection that the new results were counter-intuitive was the weakest of all the objections and was one that the mathematicians learned to live with as the new results became more familiar. However, a vestige of this suspicion was to remain, only to resurface as a strong negative reaction once there was a difficulty with the new results, such as the emergence of the set-theoretic paradoxes.

A second objection was that the new mathematicians were using powerful axioms unwarrantably. Usually the objection was that the axioms had not been shown to be consistent (with the other axioms of the systems), and there was fear that the application of such axioms would lead to contradictions in mathematics. The foremost example of this was the axiom of choice.[28] The assumption of this axiom, which stated that, no matter what the circumstances, one could make as large an infinite number of different choices as needed, with each choice representing a different set, seemed suspicious to many mathematicians. When, in 1904, Zermelo showed that the Axiom of Choice was equivalent

[28] For a discussion of objections to the Axiom of Choice, see A. Fraenkel and Y. Bar-Hillel, Foundations of Set Theory, pp. 80-86.

to the Well-Ordering Theorem,[29] there was a flood of protest against the proof,[30] and Lebesque, Borel, Baire, Lusin, and Pasch, the very mathematicians who used set theory (including the axiom of choice) to develop measure and integration theory, objected to the use of the axiom itself.[31] A similar situation arose later with Russell's logicist program, where critics argued that there was no reason to allow the multiplicative axiom (his version of the axiom of choice), the axiom of infinity (which postulated the existence of an infinite number of objects), and the axiom of reducibility (which argued that any construction in the ramified theory of types could be reduced to a type one construction) as logical principles useful in the reductionist program of the logicists.

A third criticism of the new mathematics was of its meaninglessness, usually due to the abstract nature of the results. The argument was that the extreme generality and abstractness

[29] See Mathematische Annalen, vol. 60 (1905). Zermelo's proof had appeared in volume 59.

[30] The Well-Ordering Theorem states that every set can be well-ordered, that is, put in an order such that each subset of the set has a least element.

[31] However, these mathematicians were not entirely consistent, in that they continued to use the Axiom of Choice in their own research. This point is discussed in more detail later in this chapter with respect to Borel.

of the new results did not relate in any concrete way to the results of classical mathematics. This argument was made against the axiom of choice by some mathematicians, for no indication had been given as to how to use classical techniques to make choices of these representative elements as required by the axiom. However, the most significant instance of this criticism involved the 1906 doctoral dissertation of Frechet at the University of Paris.[32] In the dissertation, Frechet generalized the notion of function by abstraction:

> For that, we will say that a functional operation is defined on a set of elements E of arbitrary nature (numbers, curves, points, etc.) when, to each element A of E corresponds a numerically determined value of U: U(A). The examination of the properties of these operations constitutes the object of the Functional Calculus.[33]

In this attempt to develop a general functional calculus, Frechet argued that function theory must resort to the general viewpoint given by Cantorian set theory. Frechet focused on a general concept of limit which he showed subsumes the objects of study of his predecessors in function theory: Cauchy integrals, Riemann integrals, Lebesque integrals, functionals. Instead of

[32] M. Frechet, Rendiconti del Circolo Mathematico di Palermo (1906).

[33] Ibid., p. 1. Quoted as translated in J. Manheim, op. cit., pp. 116-117. Emphasis in the original.

studying particular points, lines, planes, or function sets, he studies arbitrary sets of elements. Moreover,

> Frechet's study of abstract spaces brought into existence very general geometries, geometries which did not necessarily conform to Klein's method of classification. A space, in this new view, was a set of objects called points and a set of relations among these points; geometry became simply the theory of such a space.[34]

At first many mathematicians did not understand the point of Frechet's work. They were willing to grant the significance of each of the particular theories subsumed by Frechet's theory, but they could not see the point of his abstract theory itself. Where Frechet saw himself as generalizing the work of his predecessors, his critics saw him as removing all the content from the work of his predecessors. According to the critics of the new mathematics,[35] Frechet's work epitomized the difficulties inherent in the abstractness of the new mathematics.

The last, but most devastating, criticism of the new mathematics was the existence of contradictions in Cantorian set theory, the so-called antinomies of set-theoretic paradoxes.[36]

[34] J. Manheim, _op_. _cit_., p. 132.

[35] A similar position is held today by many mathematicians towards category theory, which is commonly known as "abstract nonsense."

[36] There are numerous treatments of the set-theoretic paradoxes. See, for example, A. Fraenkel and Y. Bar-Hillel, _op_. _cit_., or B. Russell, _Principles of Mathematics_ (1903).

Cantor realized, perhaps as early as 1891,[37] that the set of all sets, supposedly the largest set, would have to give rise to an even larger set, the set of all its subsets, according to the rules of his set theory. However, Cantor avoided the apparent contradiction by arguing that such sets are "absolutely infinite, inconsistent collection[s]."[38] In fact, Cantor utilized this fact in his religious interpretation of set theory and was not at all concerned about it. Although a version of the paradox was published as early as 1895,[39] its significance was not made clear until Russell introduced his version of the paradox[40] and popularized it in his 1903 Principles of Mathematics. Numerous treatments of the paradoxes appeared in English, German, French, and Italian. Many solutions were proposed, but none was really

[37] See J. Dauben, op. cit., Ch. 7 for a discussion of the chronology. The earliest paper in which Cantor discusses the problem at all is, "Über eine elementare Frage der Mannigfaltigkeitslehre," Jahresbericht der Deutschen Mathematiker-Vereinigung I, pp. 75-78. Also in Cantor, Gesammelte Abhandlungen, pp. 278-280.

[38] Letter from G. Cantor to R. Dedekind, 3 August 1899. Reprinted in Gesammelte Abhandlungen (1932), p. 445. Quoted as translated in J. Dauben, op. cit., p. 243.

[39] See C. Burali-Forti, Rendiconti del Circolo Mathematico di Palermo, 11 (1897), pp. 154-164, 260.

[40] Russell's paradox involves the Russell Set S, the set of all sets which are not elements of themselves. It is impossible to consistently answer the question whether S is an element of S.

successful until the modifications were completed to Zermelo set theory by Fraenkel and Skolem in the late 1920's. Thus, the whole theory of sets, together with all the work in topology, functional analysis, and measure theory based on this set theory, was suspect until the problem of the set-theoretic paradoxes could be resolved. Many of the critics of the new mathematics thought that the paradoxes would never be resolved and that infinite mathematics should be purged in order to preserve the foundation of classical mathematics.

Brouwer's Solution to the Foundational Crisis

These were the problems which stimulated Brouwer to develop his intuitionist program: many of the developments of the new mathematics were strongly counter-intuitive; the new mathematicians were using powerful theorems unwarrantably; much of the new mathematics was meaningless, due especially to the abstract nature of the results; and the new mathematics led to contradictions, in particular to the set-theoretic paradoxes. Brouwer believed that these problems were merely symptomatic of a larger problem which involved the entire methodology of mathematics. He felt that the new mathematics erred in adhering to a formalist approach, one which accepts objects as mathematically legitimate because of their form or formal properties when expressed in a linguistic or axiomatic logical system. Prior to Brouwer, all mathematicians had been willing to accept a substantial part of mathematics and the rules of classical logic without any justification of their legitimacy. Even objects with dubious legitimacy, such as the continuum (dubious because of its abstract definition in terms of infinite sets), were hardly questioned by the mathematical community. As Brouwer, himself, more specifically assessed the state of mathematics at the time:[41]

> The situation left by Formalism and Pre-intuitionism can be summarized as follows: for the elementary theory of natural numbers, the principle of complete induction, and more or less considerable parts of algebra and theory of numbers, exact existence, absolute reliability, and noncontradictority were universally acknowledged, independently of language and without proof. There was little concern over the existence of the continuum. Introduction of a set of predeterminate real numbers with a positive measure was attempted by logico-linguistic means, but a proof of the noncontradictory existence of such a set was lacking. For the whole of mathematics the rules of classical logic were accepted as reliable aids in the search for exact truths.

Brouwer felt that reliance on the form of an object or statement of a result was no guarantee of mathematical legitimacy, and that such reliance would lead to either meaningless or contradictory results. Statement XX adjoined to the end of his

[41] L. E. J. Brouwer, "Historical Background, Principles and Methods of Intuitionism," South African Journal of Science, October-November 1952, p. 140.

dissertation explicitly states:[42]

> To secure the reliability of mathematical reasonings one cannot succeed solely by starting from some sharply formulated axioms and strictly adhering to the laws of theoretical logic.

To him the only legitimate mathematical objects were those that could be constructed "from scratch" in a non-controversial way; the only legitimate proofs were those in which the proposition to be proved was exhibited by means of a construction. This insistence on constructions as the basis for proofs and for the legitimacy of mathematical objects is the major tenet of the constructivist position.

Such an approach, Brouwer argued, immediately would eliminate the problems created by abstraction and infinity in mathematics. Suspect abstract objects either would be given meaning by showing their construction or would be banished from mathematics until their construction could be shown, for mathematical existence was to be equated with constructibility.

Brouwer was also able to answer the problem of the infinite easily. A number of the difficulties the new mathematicians were finding themselves in, Brouwer believed, were due to the

[42] L. E. J. Brouwer, "Over de Grondslagen der Wiskunde," doctoral dissertation, University of Amsterdam, 1907. Quoted as translated in Brouwer's Collected Works, I, p. 101.

use of finitistic reasoning in an infinitistic setting. Classical logic had formulated rules, such as the principle of the excluded third, for finitistic situations. Such a principle was perfectly admissable to Brouwer in finite situations where each of a finite number of conditions could be individually checked. It was not admissible, however, in the case of the continuum, for example, where an infinite number of cases would have to be checked to determine P or not-P . Thus, the classical mathematician was getting into trouble with infinity by using principles of reasoning never intended for use in an infinite setting.

Consider, for example, the principle of the excluded third. It states that "every assignment τ of a property to a mathematical entity can be judged, i.e., either proved or reduced to absurdity."[43] This principle was perfectly admissable to Brouwer in finite situations, where each of a finite number of conditions could be individually checked. For example, to assert of two finite sets, A and B, that either they contain the same members or they do not will not lead to a constadiction. This

[43] L. E. J. Brouwer, "Consciousness, Philosophy and Mathematics," Proceedings of the 10th International Congress of Philosophy, Amsterdam, 1940. Quoted from P. Benacerraf and H. Putnam, Readings in the Philosophy of Mathematics, 1964, p. 79.

is because one can compare the elements, one by one, of the two sets to determine whether they are the same. So, one has a means for judging on the proposition. However, if the two sets, A and B, were infinite, then there would be no such means, even in principle, for judging the equivalence or non-equivalence of the two sets. Thus, in the infinite case, there is no possibility of providing a construction proving the proposition (that the two sets are equivalent) or of providing a construction proving the absurdity of the proposition without checking an infinite number of cases--which Brouwer disallows. To Brouwer's mind, this means that it is unjustified to apply the principle of excluded third in other than definite, finite cases. As Brouwer, himself, states his position on the principle of the excluded third:[44]

> Then for a single such assertation τ the enunciation of this principle is non-contradictory in intuitionistic as well as in classical mathematics. For, if it were contradictory, then the absurdity of τ would be true and absurd at the same time, which is impossible. Moreover, as can easily be proved, for a finite number of such assertions τ the simultaneous enunciation of the principle is non-contradictory likewise. However, for the simultaneous enunciation of the principle for all elements of an arbitrary species of such assertions τ this non-contradictority cannot be maintained.

[44] Ibid., p. 79.

The problems with infinity could then be resolved by showing how the problems resulted from basing the work on unfounded abstract principles and from applying finitistic logical reasoning in an infinite situation:

> . . . they [Brouwer's predecessors] seem to have introduced the continuum by having recourse to some logical axiom of existence lacking sensory as well as epistemological evidence, such as the 'axiom of ordinal connectedness' or the 'axiom of completeness' . . . in their further development of mathematics, they unreservedly contimued to apply classical logic, including the principle of the excluded third. They did so regardless of the fact that the non-contradictority of systems thus constructed had become very doubtful after the discovery of the logico-mathematical antinomies.[45]

Brouwer completed his treatment of infinite sets using what he termed the first and second acts of intuitionism, which are discussed below.

Where was this intuitionist program of Brouwer to begin? If each object in the system had to be constructed, there had to be some primitive building blocks out of which the remainder of mathematics was to be constructed. For Brouwer, the primitives were the natural numbers, which were to be understood by a vague metaphysical concept which Brouwer called the "primal" or "basal intuition" of "two-oneness." The formation of this "primal

[45] L. E. J. Brouwer, "Historical Background, . . .," p. 140.

intuition" Brouwer termed "the first act of intuitionism":[46]

> FIRST ACT OF INTUITIONISM
> completely separates mathematics from mathematical language, in particular from the phenomena of language which are described by theoretical logic and recognizes that intuitionist mathematics is an essentially languageless activity of the mind having its origin in the perception of a move of time, i.e. of the falling apart of a life moment into two distinct things, one of which gives way to the other, but is retained by memory. If the two-ity thus born is divested of all quality, there remains the empty form of the common substratum of all two-ities. It is this common substratum, this empty form, which is the basic intuition of mathematics.

Brouwer realized, however, that if the first act of intuitionism were its only platform, intuitionistic mathematics would be quite weak in its power to prove theorems. The first act of intuitionism was designed to provide a secure basis for mathematics. Any result which could be derived from the first act would clearly have a secure foundation. Unfortunately, in gaining security, the first act sacrificed breadth of coverage. A mathematics based solely on the first act would be a sterile mathematics. Thus Brouwer intervened with a second act of intuitionism.

> In this situation [in consequence of the problems over the new mathematics introduced by formalism] intuitionism intervened with two acts, of which the

[46] Ibid. pp. 140-141.

> first seems necessarily to lead to destructive and sterilizing consequences; then, however, the second yields ample possibilities for recovery and new developments.[47]

The second act was designed specifically to enable the intuitionist to state and prove results about the real numbers. Classical analysis of the real numbers was the heart of mathematics at that time. Without the capability of formulating results similar to those of classical analysis in his system, Brouwer recognized that intuitionism would not be a viable alternative to classical mathematics. The second act provides for the formation of the real number system through an intuitionistic version of Cauchy sequences, called infinitely proceeding sequences, and through the formation of intuitionistically defined sets, called species, of previously defined numbers.

> . . . a much wider field of development which includes analysis, and in several places far exceeds the frontiers of classical mathematics is opened by the
>
> SECOND ACT OF INTUITIONISM
>
> which recognizes the possibility of generating new mathematical entities:
>
> Firstly in the form of infinitely proceeding sequences $p_1, p_2, \ldots$, whose terms are chosen more or less freely from mathematical entities previously acquired; in such a way that the freedom of choice existing perhaps for the first element p_1 may be subjected to a lasting restriction at some following p_ν, and again and again to sharper lasting restrictions or even abolition at further subsequent p_ν's, while all

[47] Ibid., p. 140.

> these restricting interventions, as well as the choices of the p_ν's themselves, at any stage may be made to depend on possible further mathematical experience of the creating subject;
>
> secondly in the form of mathematical species, i.e. properties supposable for mathematical entities previously acquired, and satisfying the condition that, if they hold for a certain mathematical entity, they also hold for all mathematical entities which have been defined to be equal to it, relations of equality having to be symmetric, reflexive and transitive; mathematical entities previously acquired for which the property holds are called elements of the species. . . .[48]

Brouwer continues on to say that the second act of intuitionism creates the possibility of introducing the intuitionistic continuum and the intuitionist n-dimensional Cartesian space. It also provides the technical answer to the difficulties with infinite sets. For, according to Brouwer's theory, the only cardinalities that exist are finite, denumerably infinite, "ever denumerable, ever unfinished," and continuous.[49] No one objected to any of these cardinalities. Under such restrictions the set-theoretic paradoxes could never even be formulated. For example, Russell's paradox involved the set S of all sets which are not elements of themselves. This set could never be constructed according to Brouwer's strict rules for constructivity. Even if it had been constructed, there was no intuitionistically

[48] Ibid., p. 142.

[49] L. E. J. Brouwer, "Over de Grondslagen . . .," statement XII attached to the end of the dissertation, p. 99, in Collected Works.

means by which to check whether S was identical to any member of S .

Metaphysical Foundations: Intuitions of Space and Time

Although it seemed appropriate to use the natural numbers as a foundation for mathematics, many mathematicians of Brouwer's time (and of today as well) do not understand his reliance on this "primal intuition of two-oneness" which is described in the first act of intuitionism. To fully understand Brouwer's position, it is necessary to go beyond the boundaries of mathematics proper and examine his entire metaphysical system. This is necessary in Brouwer's case because he had a unified, systematic philosophy which began with a study of "consciousness," described the formation of language and social relations, and ended--as part of this systematic whole--with his philosophy of mathematics. While Brouwer was primarily a mathematician, not a philosopher, his metaphysical beliefs did enter in an essential way into both his philosophy and his practicing methodology of mathematics. This metaphysical system is explained in his early book, Life, Art, and Mysticism.[50] But, even upon examination of this book,

[50] Published in Amsterdam, 1903. Excerpts reprinted in Brouwer's Collected Works, I, pp. 1-10.

Brouwer's position is hard to understand. To understand it fully, one would have to trace its relation to the Dutch semiotic movement, known as significs, which was led by the philosopher-linguist Mannoury. Such an examination is outside the scope of this work.

One aspect of Brouwer's philosophy that has more direct bearing on his mathematics is of importance here--his reliance on intuition of time as the basis of mathematics. Brouwer explicitly states that he took this position directly from Kant.

> In Kant we find an old form of intuitionism, now almost completely abandoned, in which time and space are taken to be forms of conception inherent in human reason. For Kant the axioms of arithmetic and geometry were synthetic a priori judgements, i.e., judgements independent of experience and not capable of analytical demonstration; and this explained their apodictic exactness in the world of experience as well as in abstracto. For Kant, therefore, the possibility of disproving arithmetical and geometrical laws experimentally was not only excluded by a firm belief, but it was entirely unthinkable.[51]

In the Critique of Pure Reason, Kant argued that mathematical knowledge is gained by reason, through construction of concepts. That is, mathematics entails constructing, for every concept, an object which exhibits an intuition corresponding to the concept, and mathematical knowledge results from these constructions.

[51] L. E. J. Brouwer, "Intuitionism and Formalism," quoted from P. Benacerraf and H. Putnam, op. cit., p. 67.

Since Kant believed that mathematics is synthetic *a priori*[52] and "the only intuition which is given *a priori* is that of the mere form of appearances, space and time,"[53] mathematics is the study of the construction of objects according to our intuitions of space and time. To elucidate his position, Kant described the procedure a geometer would go through in discovering (and proving) the relationship between the sum of the angles of a triangle and a right angle:[54]

> He at once begins by constructing a triangle. Since he knows that the sum of two right angles is exactly equal to the sum of all the adjacent angles which can be constructed from a single point of a straight line, he prolongs one side of his triangle and obtains two adjacent angles, which together are equal to two right angles. He then divides the external angel by drawing a line parallel to the opposite side of the triangle, and observes that he had thus obtained an external adjacent angle which is equal to an internal angle--and so on. In this fashion, through a chain of inferences guided throughout by intuition, he arrives at a fully evident and universally valid solution of the problem.

Kant's reputation as a philosopher, the realistic nature of the position, and its similarity to Greek constructivity made

52 By "synthetic" Kant meant a proposition to be true by virtue of material rather than logical reasons. By "a priori" he meant that a truth could be known without (physical) experience of it.

53 I. Kant, *Critique of Pure Reason*, A720 (=B748).

54 *Ibid.*, A716 (=B743).

Kant's philosophy of mathematics popular among practicing mathematicians for the first half of the nineteenth century. However, what the mathematicians seemed to hold in common with Kant was not his overall metaphysical system but rather the specific beliefs that mathematics in some essential way was the study of constructed objects and, second that these constructions were made possible by our intuitions of space and time.

Many mathematicians specifically mentioned Kant when describing their philosophy of mathematics; but, again, it was for the specific ideas about intuitions of space and time. William Rowan Hamilton, the mathematician, based his algebraic theory of complex numbers and quaternions on the intuition of time.[55] Although it is believed that Hamilton acknowledged the role of intuition of time in his mathematical theory before he read the Critique of Pure Reason, it is clear that he used Kant to bolster his position. Hamilton wrote[56] that reading Kant's Critique of Pure Reason "encouraged [him] to entertain and publish this

[55] See Michael Crowe, A History of Vector Analysis, p. 25 for details.

[56] Sir William Rowan Hamilton, Lectures on Quaternions, Dublin, 1853, preface, p. 10.

view . . . [on complex numbers and quaternions]." Or again:

> . . . and my own convictions, mathematical and metaphysical, have been so long and so strongly converging to this point (confirmed no doubt of late by the study of Kant's Pure Reason), that I cannot easily yield to the authority of those other friends who stare at my strange theory.[57]

That algebra is the science based on the intuition of time Hamilton makes clear:[58]

> . . . a SCIENCE of Algebra: a Science properly so called; strict, pure, and independent; deduced by valid reasonings from its own intuitive principles; and thus not less an object of a priori contemplation than Geometry . . .

And a little later:[59]

> The argument for the conclusion that the notion of time may be unfolded into an independent Pure Science, or that a Science of Pure Time is possible, rests chiefly on the existence of certain a priori intuitions, connected with the notion of time, and fitted to become the sources of a pure Science; and on the actual deduction of such a Science from those principles, which the author conceives that he has begun.

The work to which Hamilton refers is "Theory of Conjugate Functions, or Algebraic Couples; with a Preliminary and Elementary

[57] Quoted in Reverend Robert Perceival Graves, The Life of Sir William Rowan Hamilton, vol. II (of 3 vol.), 1882-89, p. 142.

[58] Sir William Rowan Hamilton, "Theory of Conjugate Functions, or Algebraic Couples; with a Preliminary and Elementary Essay on Algebra as the Science of Pure Time," Transactions of the Royal Irish Academy, 17 (1837), pp. 293-422. See p. 295 for the quotation.

[59] Ibid., pp. 296-297.

Essay of Algebra as the Science of Pure Time."

The development of non-Euclidean geometry dealt such a heavy blow to Kant's philosophy of mathematics that it fell into general disrespect in the mathematics community. Kant had argued that since our intuitions of space are a priori and universally the same among people, there could be only one mathematics constructed from this intuition. But Euclidean and non-Euclidean geometries were equally valid from a mathematical point of view while being mutually contradictory. Even so, this disrespect was not shared with equal vigor among the philosophers and was certainly not universal among the mathematicians.[60] There were strong neo-Kantian philosophical schools--the most notable being in Marburg--in the last decades of the nineteenth century, and these schools adhered closely to Kant's philosophy of mathematics.[61]

Brouwer's reaction to the problem of reconciling Kant's position with mathematical develcpments was to abandon intuition

[60] For example, Hilbert argued, as one of the extra theses to be submitted with his dissertation, that non-Euclidean geometry did not require the rejection of Kant's philosophy of mathematics. For a discussion of this point, see Constance Reid, Hilbert, 1970.

[61] For more details and extended references, see Lewis White Beck, "neo-Kantianism," Encyclopedia of Philosophy, 1967, Vol.V, pp. 468-473. As a recent example of a product of this neo-Kantian school, see E. Cassirer, Philosophy of Symbolic Forms, 3 vol., 1955-1957.

of space, which had been shown untenable by the development of non-Euclidean geometry, and to focus solely on intuition.

> However weak the position of intuitionism seemed to be after this period of mathematical development [of non-Euclidean geometry], it has recovered by abandoning Kant's apriority of space but adhering the more resolutely to the apriority of time. This neo-intuitionism considers the falling apart of moments of life into qualitatively different parts, to be reunited only while remaining separated by time, as the fundamental phenomenon of the human intellect, passing by abstracting from its emotional content into the fundamental phenomenon of mathematical thinking, the intuition of two-oneness. This intuition of two-oneness, [is] the basal intuition of mathematics. . . .[62]

So, Brouwer's "basal intuition" of "two-oneness," although appearing unduly metaphysical, out of place, and irrelevant to mathematics, was actually only a radical extension of a metaphysical principle which had been held by most nineteenth-century mathematicians and had a long tradition beginning in modern times with Kant.

[62] L. E. J. Brouwer, "Intuitionism and Formalism," quoted from P. Benacerraf and H. Putnam, op. cit., p. 69.

The Greek Tradition of Mathematical Philosophy

It can be argued that the close relation between mathematics and intuitions of space and time can be traced past Kant all the way back to the classical Greeks. This relation is best understood in Greek times by examining Greek attitudes toward infinity.

Greek mathematical philosophy revolved around a concern with the infinite. The actual infinite was regarded as a vague term whose use led to indeterminant results and which was out of touch with the material world and our spatial and temporal intuitions of the world. This position seems to have developed in three roughly defined stages. In the earliest period infinity (apeiron) was equated with unboundedness (as the term was used in the discussion of divine attributes) and with indeterminateness (according to the Pythagorean duality of principles). According to the theological argument, infinite was the ascription given to divine attributes. Thus _apeiron_ was a term used to describe things not of this world.[63] The Pythagoreans ordered knowledge dualistically, by formulating explanations in terms of the

[63] For a discussion of this point, see the controversial, but interesting, Theo Sinnige, _Matter and Infinity in the_ Presocratics and Plato, 1968.

interplay of opposing principles. One such pair of principles was peiras and apeiron. In a philosophical sense, the former represented the restrictions placed on each being, especially the spatial and temporal limitations which define material objects. Contrary to this, apeiron represented indeterminateness. No material object or its attributes were represented by apeiron. Thus, infinity was not a term to apply to the mathematical examinations of this world and so was not relevant to mathematics. The relevance of this Pythagorean dichotomy was made explicit in the Pythagorean study of repeated geometrical figures, known as gnomon:

> The elements of number are the even and the odd, the latter of which is well-determined whereas the former is undetermined apeiron . . .[64]

> The Pythagoreans identify the indeterminate and the even; for, they say, when this is taken up into things and is limited by the odd, it brings indeterminateness to the beings; a proof of this is what happens to numbers; for when the gnomons are being laid around the one, or in the other way, in the latter case the figure is constantly changing, in the former it remains the same.[65]

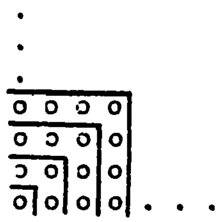
The consensus of opinion[66] is that the gnomon "laid around the

[64] Aristotle, Metaphysics A, 986a 17-19.

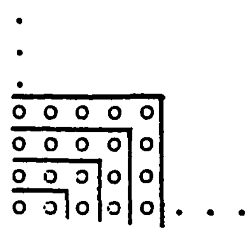
[65] Aristotle, Physics III 4, 203a 10-15.

[66] There is a bit of variation in interpretation here. See Sinnige, op. cit. He provides an extensive bibliography to the secondary sources.

one" refers to a geometrical figure constructed in a series of stages by beginning with one object and adding each time the successive odd number of objects in the following manner:



whereas the gnomon constructed "in the other way" begins with two objects and adds successive even numbers of objects in the following manner:



At the end of each stage, the gnomon "laid around the one" consisted of a perfect square, the ratio of whose sides was stable at one; whereas, at the end of each stage the gnomon constructed "in the other way" consisted of a rectangle of dimension $n \times (n+1)$, whose ratio was always approaching, but never attaining, a ratio of one. This repeated change of shape, ad infinitum, led the Pythagoreans to associate the infinite process, as well as evenness, with indeterminateness (or shape) and with instability (in the shape of the figure).

Thus the Pythagoreans associated infinity, in particular infinite processes, with the negative concept of indeterminateness: that infinite processes, in particular those of mathematics, do not lead to a determinate result. In fact, with both the deistic and Pythagorean definitions of apeiron, there is a suggestion--albeit vague--that only the finite finds a place in the material world.

The second phase in the development of the Greek concept of infinity was initiated by the paradoxes of Zeno. Zeno's paradoxes reinforced the negative connotation of infinity by showing that the introduction of infinite addition (of terms) and infinite division (of line segments) into physical problems conflicted with out perceptions of the workings of the material world. The force of the paradoxes results from conflicts with our intuitions of real time and real space. The first two demonstrate the paradoxes of infinite divisibility of a continuous whole, while the last two demonstrate the paradoxes associated with assuming that there are infinitesimal segments in the real world. For example, the first paradox, known as dichotomy, is stated by Aristotle as follows:[67]

> that which is in motion must first reach a point half-way before it will reach its goal.

[67] Aristotle, Physics VI 239b 9-14.

Aristotle explains the force of this paradox as it involves infinity:

> It is in the same way that we must meet (I) those who ask in terms of Zeno's argument whether it is a fact that you must always first traverse the half of the distance, that the halves are infinite in number, and that you cannot traverse an infinite number of distances, . . .[68]

Before any distance is traversed, a distance half as great must first be traversed, thus leading to an infinite regression and the conclusion that an infinite number of segments must be traversed in a finite amount of time--which is impossible according to our intuitions of space and time. Zeno formulated three other paradoxes,[69] the Achilles, the arrow, and the stade, designed to contradict our intuitions of space and time in a similar manner. Zeno was important because he made explicit the contradictions which would accrue if infinite processes were applied in the physical world and because he demonstrated that any application of infinite addition or infinite divisibility by mathematicians would lead to physical absurdities.

The third and culminating period in the development of the

[68] Aristotle, Physics I 263a4-b9.

[69] Aristotle describes the other paradoxes and explains them as follows: second paradox (Achilles)--stated in Physics Z9 239b 14-18, explained in Physics I 263a4-b9; third paradox (arrow)--stated in Physics VI 239b5-9; fourth paradox (stade)--stated in Physics VI 239b33-36.

Greek concept of infinity was Aristotle's theory of the potential infinity and its application in Euclidean geometry. According to Aristotle, one must distinguish between the actual infinite and the potential infinite. The acutal infinite does not exist in the material world. Only the potential infinite exists, and it is only intended for use as a convenient way of speaking--not as materially existing. In fact, when one used the term "infinite," one meant by it the indeterminately large, as large as the circumstances demanded, but nevertheless the definitely finite. This usage is well-illustrated in Euclid's Elements. Rather than discuss (infinite) lines, as we do today in geometry, Euclid discussed line segments. The first postulate of Book I[70] permits a line segment to be drawn between any two points. The following postulate permits any line segment to be extended in a straight path to any larger line segment. In the discussion on parallelism,[71] Euclid called two lines (by which he meant the line segments

[70] According to T. L. Heath's translation, the two postulates (Book 1, Postulates 1,2) are:
"Let the following be postulated:
1. To draw a straight line from any point to any point;
2. To produce a finite straight line continuously in a straight line."
See Heath for commentary. See Book I, Propositions 11, 16, 20 for the usage.

[71] See Book I of the Elements. For usage, see Propositions 27 and 29.

and any continuation along the same straight path) parallel if, no matter how far you extend the line segments, they do not meet at a point. Never is there any discussion of actual infinite lines. Line segments are merely extended far enough to do the construction required for the proof.

The Greek attitude toward the infinite was important because it was adopted by Western mathematics and was not seriously questioned until the publication of Cantor's work. The typical attitude of mathematicians prior to Cantor is elucidated by Gauss:[72]

> I protest against the use of an infinite quantity as an actual entity; this is never allowed in mathematics. The infinite is only a manner of speaking, in which one properly speaks of limits to which certain ratios can come as near as desired, while others are permitted to increase without bound.

Eighteenth and nineteenth century mathematics had adopted from the Greeks a number of specific beliefs concerning infinity. These included: that actual infinity is inconsistent and indeterminant; that it does not accord with human intuitions of physical reality (time and space). and that mathematics must accord with this physical reality, in particular, that it must accord with our

[72] Letter from C. Gauss to F. Schumacher, July 12, 1831, in C. Gauss, Werke, 8, p. 216, as translated in M. Kline, op. cit., p. 993.

intuitions of space and time. This led those mathematicians to the Greek conclusion that the use of actual infinity must be prohibited in formal, rigorous mathematics. It was thought that the actual infinite could be used as a convenient, informal way of considering mathematical problems, but that actual infinity must be replaced in rigorous mathematics by potential infinity (indefinitely large, but finite lines, figures, processes). Moreover, it was felt that the method of proof must not refer to infinite processes, such as actual infinite limiting processes. Rather, reference in proofs must always be to finite figures and their finite properties, and knowledge about the infinite, for example about limits of infinite series, must be inferred from the finite differences in finite figures as they get indefinitely large.[73]

Thus, Brouwer, who strongly opposed the use of actual infinity, can be seen as continuing this long tradition starting with the Greeks. In fact, many of the same mathematicians who so opposed Brouwer's intuitionism had been reluctant to accept Cantor's theory of infinite numbers because of beliefs which stemmed from the same root as Brouwer's. It was only when Cantor

[73] The origin of this last proposition can be found in the Greek method of exhaustion, as practiced by Eudoxes and Archemedes.

demonstrated the power and utility of his methods in solving problems in a variety of areas of mathematics that the doubts about infinite numbers began to subside in the mathematics community.

Varieties of Constructivity

Although the intuitionist program has met with intense criticism from the mathematics community, other people before and since Brouwer have developed programs which emphasized the construction procedure as the basis of a mathematical system. Such programs are called constructivist. The central tenet of mathematical constructivity is that any legitimate mathematical object must be constructed in a finite series of stages, beginning with a small number of primitively acceptable objects, and proceeding from one step to the next by one of a few acceptable means of manipulation. The underlying idea is that for a mathematical object to exist, it must be possible to build it up from basic building blocks by steps, each of which can be comprehended by the mind. Constructivity in this respect is in direct conflict with Platonism, which holds that mathematical objects exist independent of our minds; that they are to be discovered, not invented. Most practicing mathematicians, unlike Brouwer, have tended to be Platonists. Related to this, the methods of the constructivist are in direct conflict with reductio ad absurdum proofs where an object is proved to exist or to have some other property, not through a construction, but through the logical reasoning that, if it did not exist or have that property, one could derive a logical contradiction.

It is not surprising that constructivity can be expanded to a systematic philosophy of mathematics which provides a coherent theory of mathematical ontology and epistemology. In fact, this position, especially as it has been argued by Brouwer, has been a major concern of the philosophers of mathematics. However, there is another aspect of mathematical constructivity; it can be considered a methodological tenet utilized by the practicing mathematician. It is this aspect which is emphasized here. Of course, the two aspects are not entirely separable. For some practitioners of mathematics it is their philosophical system which determines their practicing methodology. But for all strict methodological constructivists, there is at least some vague philosophical belief underlying methodology. However, we are concerned here with constructivity as it relates to mathematical practice. Brouwer's intuitionism is of interest because it provided a research program in mathematics and a mathematical answer to the problems concerning infinity and abstraction.

Constructivism also has a long tradition, beginning with the classical Greeks. In fact, the history of constructivism is closely intertwined with the negative attitudes towards infinity and also with the importance of intuitions of space and time in mathematics. The constructivist position of the Greeks was most carefully formulated in Aristotle's philosophy and exhibited in Euclid's Elements.

Aristotle was clear about his requirements for ontological status for mathematical objects. He argued that a definition tells what an object is, but does not establish its existence; and that an object may be defined without existing. For Aristotle, existence had to be proved, except in a few special cases (such as points and lines) where existence was assumed along with other first principles. The proof of existence was to be by construction.

As influential as Aristotle was, he was not a practicing mathematician. In fact, Aristotle was apparently ignorant of the higher mathematics of his day. His position towards mathematical methodology was probably adopted from Eudoxes. In any case, this position is illustrated by the methodology of the Elements. Geometry dominated Greek mathematics, and it was the methodology of geometry that dominated mathematical methodology. The approach was axiomatic. Euclid allowed points, lines, and planes as the primitive objects. He then developed five postulates and five common notions (axioms) which provide the basic rules for the construction of more complicated objects from points, lines and planes, and which also allowed one to prove that the newly formed objects have the properties they were intended to have. For example, the first three postulates allow the construction of line segments between two points, the extension of an existing line segment, and the construction of a circle with given center and radius; while the common notions provided the rules for addition, equality, and inequality, such as (common notion 2) equals added to equals are equal or (common notion 5) the whole is greater than the part. The construction of more complicated objects proceeded from already existing objects by straight-edge and compass techniques.

Concern over the parallel postulate and the subsequent development of non-Euclidean geometry led late nineteenth-century mathematicians to reexamine Euclid's work. This examination confirmed that Euclid had not been precise about what was to be accepted as primitive or about what steps could be taken in a construction. As mentioned above, these mathematicians attempted to remove the logical gaps in the Elements with a modern axiomatization of the subject, which more clearly listed all of the primitive objects and all of the required postulates and axioms. Thus it is hard to determine precisely what primitives Euclid really allowed and in what way these primitives were to be combined to produce more complicated objects.

One point of note, especially considering the disputes over the technique in the late nineteenth century, is that Euclid never proved existence by a *reductio ad absurdum* argument. So, not only did Euclid establish in every case that a construction existed, he also exhibited the construction. Even when Euclid was showing that exactly so many of a particular type of object existed and that no more exist, he did not resort to any of the various indirect methods of proof used frequently in modern mathematics. Rather, he constructed the objects that did exist and showed directly how the construction insured there could be no others. Such a technique is best illustrated in Euclid's proof that exactly five regular solids exist. This is the final and culminating proof in the *Elements*. In Book XIII, Euclid constructed the regular solids: tetrahedron (Proposition 13), cube (Proposition 15), octahedron (Proposition 14), cosahedron (Proposition 16), and dodecahedron (Proposition 17)—all by ruler and straight-edge constructions. In Proposition 18 Euclid compared the five figures. After comparing the properties and the way in which these five figures are constructed, he wrote:

> I say that *no other figures, besides the said five figures, can be constructed which is contained by equilateral and equiangular figures equal to one another.*[74]

[74] Euclid, *Elements*, Book I, Proposition 18. Sir Thomas L. Heath, translation.

The proof relied on the way in which the five figures themselves were constructed, and not on some abstract property which would not hold if there were more than these five:[75]

> For a solid angle cannot be constructed with two triangles, or indeed planes.
> With three triangles the angle of the pyramid is constructed, with four the angle of the octahedron, and with five the angle of the icosahedron:
> but a solid angle cannot be formed by six equilateral and equiangular triangles placed together at one point,
> for, the angle of the equilateral triangle being two-thirds of a right angle, the six will be equal to four right angles:
> which is impossible, for any solid angle is contained by angles less than four right angles.
> For the same reason, neither can a solid angle be constructed by more than six plane angles.
>
> By three squares the angle of the cube is contained, but by four it is impossible for a solid angle to be contained,
> for they will again be four right angles.
> By three equilateral and equiangular pentagons the angle of the dodecahedron is contained;
> but by four such it is impossible for any solid angle to be contained,
> for, the angle of the equilateral pentagon being a right angle and a fifth, the four angles will be greater than four right angles:
> which is impossible.
>
> Neither again will a solid angle be contained by other polygonal figures by reason of the same absurdity.
> Therefore, etc.

[75] _Ibid_., Book I, Proposition 18.

As long as these was no conflict between an avowal of constructivist and contemporary mathematics, mathematicians were willing to avow a constructivist philosophy. Such was the case until the last third of the nineteenth century. Until that time the mathematical problems being examined required no use of actual infinity, no highly abstract objects, and little use of proofs of existence by contradiction. Although there was no systematic listing of the constructive techniques that one was permitted to use in mathematics (as Brouwer provided for intuitionistic mathematics) and although there were some developments in analysis that Brouwer would not have accepted as intuitionistically legitimate, nineteenth-century mathematicians did believe it was appropriate to build up all objects which they studied.

However, the almost unconscious acceptance of this moderate version of constructivity began to change in the 1870's. Kronecker provided the first instance of change, but his pattern has been repeated numerous times since then. The conscious avowal of a constructivist methodology has arisen several times when there has been concern over the legitimacy of new results in mathematics, especially if they occur as the product of some new type of mathematical object or new methodology. As a conservative reaction, to protect the soundness of mathematics, mathematicians would revert to the method that had worked so well for Euclidean geometry and which had been the model for classical mathematics in the eighteenth and early nineteenth centuries, constructivism. Then the object(s) or principle(s) in question would either be shown to have the desired properties or be rejected from mathematics since one then would start with primitive objects acceptable to the mind and proceed with steps that were acceptable to the mind. In this way the nature of the object could be more fully understood and be accepted, or it could be branded as incomprehensible and thus inadmissable.

Kronecker, in the 1870's and 1880's, was the first to choose consciously a constructivist position in the face of problems he saw in new developments in mathematics. He was worried about the development of all the new number systems, especially since they did not seem to have much to do with physical reality. In particular, Kronecker was concerned about the development of quaternions by Hamilton in the 1840's, of complex numbers and the generalization of algebra to the study of the manipulation of symbols by DeMorgan in the 1840's, of matrices by Cayley in the 1850's, and of vector analysis by

Maxwell in the 1870's and by Heaviside and Gibbs in the 1880's.[76]

Kronecker's approach was to try to save these various new number systems (from meaninglessness) by showing how they could be built up from the natural numbers and its arithmetic. His theory is summed up in his famous aphorism, "God made the integers; all else is the work of man." By means of definitions in terms of natural numbers, the negative, rational, real, and complex numbers were to be defined and, from there, the remainder of classical mathematics, including the rest of these new number systems, was to be reformulated in terms of these definitions. This was the program he set for himself and his students.[77]

Kronecker insisted that all legitimate mathematics must be capable of reduction to the natural numbers. The force of Kronecker's approach and its variance from the classical

[76] See Hamilton, Lectures on Quaternions (1853); A. DeMorgan, Trigonometry and Double Algebra (1849); A. Cayley, "A Menoir on the Theory of Matrices," Journal für die Reine und Angewandte Mathematik, 50 (1855), pp. 282-285, in Cayley's Collected Mathematical Papers, 2, pp. 475-496, and many following papers; J. C. Maxwell, A Treatise on Electricity and Magnetism, 1873; O. Heaviside, Electromagnetic Theory, 3 vol., (1893, 1899, 1912); and J. W. Gibbs, "Elements of Vector Analysis," 1881, privately published pamphlet, Yale University, in Gibbs' Scientific Papers, 2, 17-90. Also see J. W. Gibbs and E. B. Wilson, Vector Analysis (1901). A general overview of these developments can be found in Michael Growe, A History of Vector Analysis (1967).

[77] See L. Kronecker, "Über den Zahlfegriff," Journal für die Reine und Angewandte Mathematik, 101, 1887, pp. 337-355. Also in Werke, 3, pp. 251-274.

approach is best appreciated by examining Kronecker's controversies with classical mathematicians. Three are of special importance.

During the nineteenth century numerous different proofs were given for the fundamental theorem of algebra.[78] Many of these proofs made use of profound results from the theory of functions of a real or complex variable or were developed from topological facts. Kronecker strongly objected to these proofs. According to Kronecker, since arithmetic was the starting point of all of mathematics, it was a _petitio principii_ to use results from analysis or topology to prove a purely algebraic result, such as the fundamental theorem of algebra.

The second controversy involved Hilbert's proof of Gordan's conjecture of the existence of a finite basis in invariant space theory. Hilbert proved the existence of a finite basis, not by constructing it, but by a purely existential argument. That is, Hilbert showed that logical reasoning from the premises of invariant space theory required a finite basis. It was very difficult to construct finite bases, and Kronecker and Gordan had succeeded only after much work under very special

[78] The fundamental theorem of algebra states that every polynomial equation with complex coefficients has a complex number root.

circumstances. Now Hilbert showed that every one of an infinite number of cases must each have a basis without invoking a single conviction! Kronecker and Gordan refused to accept Hilbert's proof even though they had no objection to the logical steps of the proof. As Gordan put it, "this is not mathematics; this is theology."

The third controversy was Kronecker's famous dispute with Cantor over the transfinite numbers. Kronecker objected to numbers which could not be reached by proceeding from his natural numbers. Kronecker, in fact, was Cantor's earliest and most vehement critic. He was so opposed to Cantor's work that he insured that Cantor's publications would not appear in the best German mathematics journals. Cantor even attributed his unimpressive university position at Halle to Kronecker's influence.

The situation of Brouwer is analogous to that of Kronecker. Brouwer was concerned over abstraction in mathematics, in particular the use of a formalist approach, and also over the use of infinity. He adopted the same sort of conservative methodology as Kronecker along constructivist lines, so as to save mathematics from suspect results. Because both positions were not merely philosophy, but had important mathematical consequences, mathematicians hardened quickly in their opinions either for or against both Kronecker's and Brouwer's results.

Actually, there have been many other constructivist positions besides those of Kronecker and Brouwer. In most cases the positions seem to have been adopted under similar circumstances as those described above. Among the most important are the French intuitionists and their position on set theory, Hilbert's metamathematics for his formalist program, Herbrand's finitism, Skolem's primitive recursive arithmetic, Esenin-Volpin's ultrafinitism, and Bishop's constructive analysis. Esenin-Volpin's[79] and Bishop's[80] work are modern examples of constructibility and are not relevant here. Hilbert, Skolem, and Herbrand are examined in the next chapter. However, the French intuitionist position, which was the direct result of the powerful principles (Axiom of Choice, Power Set Axiom) used to create large and abstruse sets, was a direct reaction to the same problems Brouwer saw in the new mathematics and is discussed in the next section.

[79] A. Esenin-Volpin, "Le Programme ultra-intuitionniste des fondements des mathematiques," Infinitistic Methods, Warsaw, 1961, pp. 201-223. Also, "The untra-intuitionistic criticism and the antitraditional program for foundations of mathematics," in Kino, Myhill, and Vesly, Intuitionism and Proof Theory, 1970, pp. 3-45.

[80] See E. Bishop, Foundations of Constructive Analysis, 1967. Also see E. Bishop, "The Crisis in Contemporary Mathematics," Historia Mathematica, 2 (1975), pp. 507-517.

Other Solutions to the Foundational Crisis

Brouwer was not a lone wolf crying about the problems of abstraction and infinity. The problems did not attract the entire mathematics community because many of the active research areas had no use for any of these suspect principles or objects. However, as discussed above, these principles found heavy use in set theory, topology, measure and integration theory, and functional analysis. The scores of attempts to resolve the set-theoretic paradoxes indicate just how many mathematicians were worried over these issues. There were also numerous attempts to determine the power and the foundation of the axiom of choice. Discussed below are some of the more important attempts to reconcile the problems in set theory, concerning both the paradoxes and the axiom of choice. These can be seen as alternatives to Brouwer's solution to the problems in the new mathematics.

In 1904, Zermelo,[81] in reaction to König's erroneous proof[82] that the set of real numbers could not be well-ordered, provided a proof that any arbitrary set M could be well-ordered. This

[81] E. Zermelo, "Beweis, dass jede Menge wohlgeordnet werden kann," Mathematische Annalen, 59 (1904), pp. 514-516.

[82] J. König, "Zum Kontinuum-Problem," Verhandlungen des Dritten Internationalen Mathematiker-Kongresses in Heidelberg, Leipzig, 1905, pp. 144-147.

required, for an arbitrary set M , that there be an ordering of the elements of M such that every subset of M had a least element. In his proof, Zermelo used the Axiom of Choice[83] in a fundamental way to make a possibly infinite number of choices of these least elements.

A number of objections appeared to the proof in the issue[84] of Mathematische Annalen following the one in which Zermelo published his proof. Among the more interesting of the negative responses were those of the French empiricists,[85] a group of French analysts of a similar mind who were working on the same sort of problems in functional analysis and measure theory. One member of the group, Borel, objected that Zermelo had not proved the Well-Ordering Theorem, but only proved the equivalence, for an arbitrary set M , of two tasks: A. to well-order M ; B. to choose a definite but arbitrary member m' from each non-empty subset M' of M . This, Borel claimed,

> cannot be considered as providing a general solution to problem A. In fact, in order to regard problem B as resolved for a given set M, one needs to give a means,

[83] Remember, the Axiom of Choice states that, for any collection of non-empty sets, one can choose a representative element from each set such that no two representative elements are the same.

[84] Mathematische Annalen, 60 (1905).

[85] This informal group consisted of Borel, Lebesque, Baire, their students, and some of their lesser-known colleagues.

> at least a theoretical one, for determining a distinguished element m' from an arbitrary subset M'; and that problem appears to be among the most difficult, if one supposes, for the sake of definiteness, that M coincides with the continuum [the set of all real numbers].[86]

In fact, Borel took an even harder line against the proof by insisting that "any argument where one supposes an arbitrary choice to be made a non-denumerably infinite number of times . . . [is] outside the domain of mathematics."[87] Hadamard, another of the French empiricists, pointed out[88] that what was required to make the proof legitimate was to insure that the choice function γ, by which these least elements were chosen, was "effectively defined" or, in other words, that the function be constructible.

Baire, also one of the French empiricists, adopted an even more radical line than Borel. For Zermelo to carry out his proof, he had to apply the Axiom of Choice to the collection of non-empty subsets of M . Baire objected even to the admissability of

[86] E. Borel. "Quelques remarques sur les principes de la theorie des ensembles," Mathematische Annalen, 60 (1905), pp. 194-195. Quoted as translated in Gregory H. Moore, "The Origins of Zermelo's Axiomatization of Set Theory," Journal of Philosophical Logic 7 (1978), p. 312.

[87] E. Borel, op. cit., p. 145. Quoted as translated in G. H. Moore, op. cit., p. 313.

[88] See R. Baire, E. Borel, J. Hadamard, and H. Lebesque, "Cinq lettres sur la theorie des ensembles," Bull. Soc. de France 33 (1905), pp. 261-273.

forming the set S of non-empty subsets of M and, therefore, to the admissability of such a choice function $\gamma: S \to M$. For Baire, an infinite set of any cardinality was "virtual," i.e., existing only as it was defined by certain conventions. Thus the structure of such a set was indeterminate. In particular, he believed "it is false . . . to consider the subsets of this set [M] as given."[89]

A third member of the French school, Lebesque, also took a constructivist position. He argued that an object can only be considered to exist when it has been defined in a finite number of words. Yet, Zermelo had not defined the choice function γ uniquely, nor had he shown that the subsets M' of M were defined in a unique way. For these reasons he rejected Zermelo's proof.

During 1907,[90] in reaction to all the controversy over his well-ordering axiom, Zermelo developed an axiomatic system for set theory. Zermelo's hope was that an explicit statement of the principles used and a formal proof within an axiomatic system would convince his critics that he had a legitimate proof of

89 Ibid., p. 264. Quoted as translated and excerpted in G. H. Moore, op. cit., p. 313.

90 See E. Zermelo, "Neuer Beweis fürdie Möglichkeit einer Wohlordnung," Mathematische Annalen, 65 (1908), pp. 261-281.

the well-ordering principle. Zermelo's axiomatization was a significant improvement over Cantor's "naive" set theory, for Cantor's attempt to define a set was vague and inapplicable:[91]

> Definition: By a "set" we mean any collection M into a whole of definite, distinct objects m (which are called the "elements" of M) of our perception [Anschauung] or of our thought.

Zermelo believed that his axiomatic system had another virtue in that the Axiom of Separation[92] prevented the formulation of the set theoretic paradoxes.[93] Thus, Zermelo sought to proscribe set-theoretic procedures in the form of axioms for the formation of sets which would demonstrate that the Axiom of Choice and its uses were legitimate while prohibiting the set-theoretic paradoxes.

Russell used a rather different approach to resolve the set-theoretic paradoxes. His program for mathematics, logicism, like Brouwer's intuitionism, was shaped by a larger philosophical

[91] G. Cantor, "Beiträge zur Begrundung der transfiniten Mengenlehre," Part I, Mathematische Annalen, 46 (1895), p. 482, as translated in Contributions to the Founding of the Theory of Transfinite Numbers, translated by P. E. B. Jourdain, Chicago, 1915.

[92] The Axiom of Separation (Aussonderung) states: For any set S and any predicate P which is meaningful for all elements of S , there exists a set Y that contains just those elements X of S which satisfy the predicate P.

[93] See A. Fraenkel and P. Bernays, Axiomatic Set Theory, for details.

program. But also like intuitionism, logicism had decided significance for the practice of mathematics as well as for philosophy. Russell's logicist program was formulated in his 1903 Principles of Mathematics and carried out, with Whitehead, in the three-volume Principia Mathematica (1910-1913). According to the logicist program, mathematics can be reduced to a mere extension of logic. The plan was to show that all of mathematics could be deduced from a set of axioms describing logical principles without need for extra axioms of a specifically mathematical character. An important part of the logicist program was the theory of types.[94] It was intended to resolve the set-theoretic paradoxes, whose significance Russell had made apparent to the mathematical and philosophical communities with his Russell set in Principles of Mathematics. The idea underlying the theory of types was to categorize all propositions in such a way that variables in a type n proposition must all be of type less than n. In this way, formulation of the Russell set is impossible.

> These fallacies [set-theoretic paradoxes] . . . are to be avoided by what may be called the "vicious-circle

[94] The theory of types was first formulated in 1906. See Russell's Autobiography, I, pp. 194 ff. for the history of the development of the theory of types. For a technical description, see B. Russell, "Mathematical Logic as Based on the Theory of Types," American Journal of Mathematics, 30 (1908), pp 222-262.

> principle"; i.e., "no totality can contain members defined in terms of itself." This principle, in our technical language, becomes: "Whatever contains an apparent variable must not be a possible value of that variable." Thus whatever contains an apparent variable must be of a different type from the possible values of that variable; we will say that it is of a higher type.[95]

Problems with the Solutions

Thus there were a number of conservative reactions to the problems created by abstraction and infinity in mathematics. Brouwer's intuitionism sought to avoid the problems by restructuring mathematics with a constructivist methodology. The French intuitionists called for a constructive restriction in the use of the axiom of choice and the power set operation. Zermelo sought to avoid the set-theoretic paradoxes and demonstrate the legitimacy of the axiom of choice by carefully proscribing the rules which were allowed in the formulation of sets. Russell sought to avoid the set-theoretic paradoxes by reducing mathematics to logic and then restricting the logical principles used in the construction of sets. There were many other attempted solutions,

95 B. Russell, "Mathematical Logic . . .," as quoted in Robert C. Marsh, ed., Logic and Knowledge, 1956, p. 75.

too numerous to discuss here. A more typical response by a practicing mathematician was that of Hausdorff,[96] who simply prohibited as self-contradictory any set containing all the ordinals or all the cardinals.

Unfortunately, all of these approaches were doomed to failure: some because they were only fregmentary solutions with no apparent extension to a complete solution; some because they merely replaced the problems of abstraction and infinity with other problems.

The French intuitionists' and Hausdorff's solutions were of the fragmentary variety. Although Borel could explain his objections to the use of the axiom of choice and did suggest that a constructivist approach be adopted, he admitted[97] that the actual working out of a constructivist approach was a difficult problem--one he never attempted. In fact, as Hadamard[98] and later Peano[99] pointed out, Borel was inconsistent, using the very techniques in his own research to which he objected in Zermelo's proof of the well-ordering principle.

[96] See F. Hausdorff, "Grundzüge einer Theorie der geordneten Mengen," Mathematische Annalen, 65 (1908), pp. 435-505.

[97] See the quotation associated with footnote 86.

[98] See Baire, Borel, Hadamard, and Lebesque, op. cit.

[99] See G. Peano, Formulaire de mathematiques, Turin (1895-1901), 3 vol.; also G. Peano, "Additione," Revistade mathematica 8 (1906), pp. 136-143.

Similarly, although Hausdorff avoided the operations which led directly to the set-theoretic paradoxes, his solution was suspect in several ways. First, his procedure was ad hoc. There was no reason to exclude these particular operations except that experience had shown they led to difficulties. Second, there was no way of telling how much such a prohibition would hurt research in set-theory. Just how much mathematics was Hausdorff eliminating by prohibiting perhaps harmless uses of all ordinals or all cardinals? Third, and most important, there was no reason to believe Hausdorff had eliminated all of the set-theoretic difficulties simply by banishing sets including all the ordinals or all the cardinals.

Although Zermelo's axiomatization ultimately led to an axiomatization that is accepted today, it was not until the end of the 1920's that all of the technical details of the axiomatization were worked out by Skolem and Fraenkel. Beside this, there remained the problem of objections to the axiom of choice, which was included in Zermelo's axioms for set theory. While Zermelo thought that the new version of the choice axiom which he introduced was of a "purely objective character [which] is immediately evident."[100] his critics were not convinced. The

100 E. Zermelo, "Neuer Beweis . . .," quoted as translated in G. Moore, op. cit., p. 320.

fact that there is still concern today over the highly abstract objects capable of definition in Zermelo-Fraenkel set theory indicates that the axiomatic approach has not answered entirely the problems of the new mathematics.

There were a number of problems with Russell's logicist program as well. First, the original version of the theory of types was attacked as arbitrary. Russell replied that, based on their naure, the theory of types was the only way to avoid the paradoxes. Second, despite the theory of types, Russell's critics were able to show that the paradoxes could still be formulated. Russell responded with his ramified theory of types, but the critics were not satisfied with the technical manipulations of the theory and answered that his solution was *ad hoc*. Third, the Axiom of Reducibility which Russell had added to insure that all of classical analysis could be formulated within the logicist program was even more objectionable to the critics. The Axiom of Reducibility stated, in effect, that for any proposition of type n, with $n>1$, there was an equivalent proposition of type one. There was no reason for adopting this particular axiom other than that it was what was needed to save the mathematics. The axiom was labelled incorrect, *ad hoc*, and extralogical. Fourth, two other axioms of Russell's system, the multiplicative axiom and the axiom of infinity, were also criticized as being extralogical. The multiplicative axiom is equivalent to the axiom of choice, which has been discussed above. The axiom of infinity asserts the existence of an infinite number of distinct objects, which was criticized as a material rather than logical assertaion. An extralogical axiom, of course, would wreck the reduction of mathematics to logic. Fifth, in fact, without criticizing any axiom in particular, there were many objections in general, both mathematical and philosophical, to the possibility of carrying out this reductionist program. Finally, logicism was criticized for never working out all the details of the program--even after the three-volume Principia Mathematica was completed.

Unfortunately, there were also difficulties with Brouwer's proposed solution to the problems of the new mathematics. The difficulties did not involve the intuitionist's solution to the problems with infinity and abstraction. It is clear that the set-theoretic paradoxes could not be formulated in intuitionistic mathematics. It is also clear that intuitionism resolved any problem with abstract objects, either by giving them meaning through a construction or by banishing them altogether from mathematics. Rather, the difficulty lay in the poverty of intuitionistic mathematics research programs, due to the restrictiveness of the constructivist method. This poverty of intuitionistic versus classical mathematics was shown in four ways:

(1) Sometimes the intuitionists obtained the same results as

the classical mathematicians but had to resort to different (and often more difficult) arguments. This was the case, for example, in proving the existence of the transcendental numbers. Cantor provided the only straight-forward proof by using a diagonal argument to show that more real than algebraic numbers existed. This proof was not admissible to the intuitionists. However, a long and tedious argument can be given to prove the theorem in a way acceptable to the intuitionists if one is careful about orderings.[101]

(2) Sometimes the intuitionists could only give a result similar to the classical result. For example, the intermediate value theorem of classical mathematics states that, for any function f continuous on the closed interval from a to b such that $f(a)<0$ and $f(b)>0$, there is a number n such that $f(n)=0$ and $a<n<b$. The strongest intuitionist version of this theorem has the same hypotheses, but concluded that for any positive integer m there is a c, with $a<c<b$, such that $-\frac{1}{m}<f(c)<\frac{1}{m}$.

(3) Sometimes there was more than one intuitionist analogue of a classical result. Such is the case with the theory of

[101] Neither is there a direct intuitionistic analogue of F. Lindemann's proof (Mathematische Annalen 20 (1882), pp. 213-25), which is based on the structure rather than the cardinalities of the real and algebraic numbers.

infinite series. Brouwer claimed that the notion of convergence should be split into a "positive" and "negative" theory of convergence. Belinfante[102] showed that the "positive" and "negative" approaches lead to quite different theories, with only the "positive" theory resembling the classical theory.

(4) Sometimes there was no intuitionist analogue to a classical theorem, or worse, there was an intuitionistic result which was the negation of the classical result. This is especially true in analysis, at the heart of classical mathematics. The Bolzano-Weierstrass theorem, the convergence of a bounded, monotone sequence of real numbers, the theory of Dedekind cuts, the existence of a least upper bound for a bounded set of real numbers, and the existence of a maximum for a continuous real function in a closed interval are all either false or meaningless under intuitionist mathematics.

It is no wonder that most practicing mathematicians were vehemently opposed to Brouwer's intuitionism. In fact, many mathematical practitioners were not concerned about the problems

[102] See H. Belinfante, "Zur intuitionistishen Theorie der unendlichen Reihen," Sitz. Berlin, 1929, No. XXIX; "Über eine besondere Klasse von non-oszillierenden Reihen," Proc. Amsterdam Soc. 33 (1930), pp. 1170-1179; "Das Riemannsche Umordnumgsprinzip in der intuitionistischen Theorie der unendlichen Reihe," Composito Math., 6 (1938), pp. 118-123.

of abstraction and infinity because these problems never surfaced in their research areas. But what about those researchers who studied logic, set theory, topology, measure and integration theory, or functional analysis? How were they to reconcile the power of the new methods of abstract and infinity which were so crucial in their research with the difficulties that these methods introduced? Hilbert, in particular, was worried by this problem. He was sensitive to the problems of the new mathematics. He was enamored with the constructivist solution proposed by Brouwer. But he would not relinquish mathematical results for the sake of constructivity as Brouwer had been willing to do. Hilbert phrased his position in two well-known aphorisms:

> No one shall expel us from the paradise which Cantor created for us.[103]
>
> Forbidding a mathematician to make use of the principle of excluded middle [as Brouwer does] is like forbidding an astronomer his telescope or a boxer the use of his fists.[104]

The next chapter will show how Hilbert attempted to resolve this dilemma with his formalistic program.

[103] D. Hilbert, "Über das unendliche," Mathematische Annalen, 95 (1926), p. 170.

[104] See H. Weyl, Bulletin American Mathematical Society, 50 (1944), p. 637.

Chapter II: From Hilbert's Program to Recursive Function Theory

Like Brouwer, David Hilbert was seriously concerned about the problems abstraction and infinity had caused for the foundations of mathematics. He believed that the discovery of a secure foundation was among the most important tasks awaiting the mathematician. However, unlike Brouwer, Hilbert was determined to provide a foundation for all of mathematics--not just that fragment where security was easily won. Although Hilbert severely criticized Brouwer's intuitionism for excluding results from mathematics because they were problematic, he was enamored nevertheless with the constructivist approach. His solution for the foundational crisis,commonly known as formalism, was an ingenious merger of constructivism with his previously well-developed interest, axiomatics.

This chapter will describe Hilbert's formalist program to solve the foundational crisis and Gödel's incompleteness theorem, which spelled the total collapse of this program. It will be shown that at the root of both Hilbert's and Gödel's work was the use of recursive functions, which are the formal, mathematical analog of those functions one intuitively knows how to compute. The roots of recursive function theory are traced, ending with the precise, mathematical characterization by Alan Turing and Emil Post of the recursive functions as those functions which can be computed by an idealized machine. The chapter concludes with a description of the first plans for the use of this mathematical theory in the construction of physical computing machinery.

Hilbert's Program

Like Brouwer, Hilbert adopted a highly philosophical approach to the foundational crisis. His philosophy of mathematics explicitly acknowledged Kant as a forerunner. The lineage was not as direct as from Kant to Brouwer, however, because Brouwer was willing to relinquish most of analysis for his philosophical position, whereas Hilbert's overriding concern was with preserving all of classical mathematics. Thus, while Hilbert wanted to believe, with Kant, that mathematical knowledge depended ultimately on a priori intuitive insight, he also wanted to accept certain portions of classical mathematics which apparently could not be constructed from such intuition. It was through his attempt to reconcile these apparently contradictory positions that he arrived at his philosophy of mathematics, which conflated a purely formalist approach to mathematics with a purely intuitionist approach to metamathematics.

Hilbert was thoroughly acquainted with, and had at least moderate sympathy for, Kant's philosophy of mathematics. For instance, in fulfillment of his final doctoral requirement, Hilbert defended the proposition "that the objections to Kant's theory of the _a priori_ nature of arithmetical judgements are unfounded." This was at a time when Kant's views of mathematics were in general disrepute owing to the discovery of non-Euclidean geometries. Although these discoveries discredited Kant's views of the nature of geometrical axioms, Hilbert chose to open his _Foundations of Geometry_ with an epigraph from Kant:

> All human knowledge begins with intuitions, then passes to concepts, and ends with ideas.[1]

Although qualifying his agreement, Hilbert explicitly acknowledged his indebtedness to Kant:

> I admit that even for the construction of special theoretical subjects certain _a priori_ insights are necessary. . . . I even believe that mathematical knowledge depends ultimately on some kind of such intuitive insight. . . . Thus the most basic thought of Kant's theory of knowledge retains its importance. . . . The _a priori_ is nothing more or less than . . . the expression for certain indispensable preliminary conditions of thinking and experiencing. But the line between that which we possess _a priori_ and that for which experience is necessary must be drawn differently by us than by Kant--Kant has greatly overestimated the role and the extent of the a priori.[2]

[1] Quoted from Constance Reid, _Hilbert_, p. 62.

[2] From a Fall, 1930 Königsberg address to the Society of German Scientists and Physicians, as quoted in _Ibid._, p. 194.

The motivation for Hilbert's philosophy of mathematics, however, was not a direct reaction to Kant. Instead, it was a direct reaction to arguments by Brouwer and Weyl, reminiscent to Hilbert of earlier arguments he had heard from Kronecker. These arguments concerned imposing limitations on mathematics in order to provide it with a constructive character. Kronecker was among the most powerful mathematicians in Germany during Hilbert's early career, and Hilbert was attracted to Kronecker, basing a number of early papers on his methods and problems. Hilbert disagreed, however, with Kronecker's beliefs that the only mathematical objects which exist are those which can be constructed from a finite number of positive integers, and that existence proofs are meaningless unless they actually specify the object asserted to exist. Hilbert deplored the imposition of what he saw as Kronecker's restrictive personal prejudices, and retorted concerning existence proofs:[3]

> The value of pure existence proofs consists precisely in that the individual construction is eliminated by them, and that many different constructions are subsumed under one fundamental idea so that only what is essential to the proof stands out clearly; brevity and economy of thought are the reason d'etre of existence proofs. . . . To prohibit existence statements . . . is tantamount to relinquishing the science of mathematics altogether.

[3] *Ibid.*, p. 37.

Hilbert saw this same restrictive attitude reappear in Brouwer and his student, Weyl. Once again mathematics of the continuum was being severely limited by the mandate that a mathematical object must be actually constructed to exist. Hilbert angrily responded:

> What Weyl and Brouwer do comes to the same thing as to follow in the footsteps of Kronecker! They seek to save mathematics by throwing overboard all that which is troublesome. . . . They would chop up and mangle the science. If we would follow such a reform as the one they suggest, we would run the risk of losing a great part of our most valuable treasure![4]

Hilbert was faced with countering Brouwer. But how was he to formulate a philosophy of mathematics which preserved both the constructive character and the classical results, which often took the Platonist attitude that objects exist without our providing a construction of their existence?

Hilberts' philosophy of mathematics was based upon a distinction between mathematics and metamathematics. Mathematics studies mathematical objects, such as sets and numbers,and their properties. Metamathematics studies the methods of reasoning used in mathematics and the objects, such as proofs, used to carry out this reasoning. According to his scheme, mathematics was to be carried out in a purely formalistic manner, while metamathematics was to be carried out in an intuitive or, as Hilbert called it,

[4] At a 1922 meeting in Hamburg, as quoted in Reid, p. 155.

"finitary" manner. The idea was to formalize each statement and proof of classical mathematics in an exact, symbolic language and to take this collection of formalized statements and proofs as the object of mathematical study. The motivation for this approach stems from Hilbert's concept of meaningfulness.

Mathematics, Hilbert thought, consisted of statements about symbols--strokes (numerical symbols) in particular--which have no significance in themselves.

> . . . each numerical symbol is intuitively recognizable by the fact it contains only 1's. These numerical symbols which are themselves our subject matter have no significance in themselves. But we require in addition to these symbols, even in elementary number theory, other symbols which have meaning and which serve to facilitate communication, for example the symbol 2 is used as an abbreviation for the numerical symbol 11, and the numerical symbol 3 as an abbreviation for the numerical symbol 111. Moreover, we use symbols like +, =, and ÷ to communicate statements. 2+3=3+2 is intended to communicate the fact that 2+3 and 3+2, when abbreviations are taken into account, are the self-same numerical expression, viz., the numerical symbol.[5]

Only particular statements about particular symbols are admitted as meaningful. Any true, and thereby meaningful, statement must be intuitive (anschaulich), i.e., finitely cognitively graspable. This provides a means for classifying mathematical statements according to meaning. Particular arithmetical formulas, e.g.

[5] Hilbert, "On the Infinite," as reprinted in Benacerraf and Putnam, Readings in the Philosophy of Mathematics, p. 143.

2+3=3+2, are meaningful. The statement "there are no prime numbers between 100 and 200" is meaningful, for it can be written as a finite disjunction with disjuncts of the form "a can be further factored", where a is a number between 100 and 200. Some statements are partially meaningful.[6] For example, the statement "there is no prime number greater than 100" is partially meaningful. It has no meaning by itself, for it presupposes grasping an infinite disjunction. However, the statement can form part of a meaningful statement, e.g., if "and less than 200" is added to it. Finally, there are meaningless statements.[7] For example, the statement "there are an infinite number of twin primes" is meaningless, for there is no finite way to cognitively grasp it.

Thus, individual statements of mathematics are either

[6] This is an interpretation of an argument of Hilbert as reprinted in Benacerraf and Putnam, pp. 143-144.

[7] There are other sorts of examples which are, somewhat oddly, thought of as meaningless. General statements (containing unbounded quantifiers), e.g., for all a and b , a+b=b+a, are meaningless. This attitude reflects Hilbert's concern with mathematics as content (for here there is a different content for each of an infinite number of choices for a and b), and not with form (for there is only one form). However, once instantiation occurs throughout the general statement, it becomes meaningful. Also, negations of certain meaningful statements are not meaningful. See Benacerraf and Putnam, p. 144, where Hilbert says that $1+a \neq a+1$ is meaningless for a particular numerical symbol a .

meaningful or meaningless. What distinguishes them is that the meaningful ones are inhaltlich (material, of content). Instead of restricting mathematics to inhaltliche statements (similar to the ploy of Brouwer), Hilbert relinquished entirely the concept of meaning. The meaningless statements were added, as ideal statements,[8] to the inhaltliche statements in order to form the realm of mathematics. Once the ideal statements were added, however, the problem emerged. How is the truth value of an ideal statement to be determined? How is the truth value of an inhaltliche statement whose proof relies on ideal statements to be determined?

Hilbert's solution was to return to his prior experience with axiomatics and choose an exact, symbolic (formal) language which was powerful enough to express the statements and proofs of classical mathematics, but which still had precise rules for determining the meaningful statements. Hilbert then identified mathematics with the set of provable formulas within a formal, axiomatic system, thereby reducing the truth of classical

[8] The addition of ideal elements was a common method of mathematics during the period. Ideal elements, e. g., points at infinity, or ideals in algebra, were added to make the work simpler and more complete. They obeyed the same rules but were distinguished from the "real elements." Hilbert had used ideal elements previously in a paper on invariant theory.

mathematical statements to the provability of their representative formulas within the formal system.

Hilbert required, however, that the formal system represent classical mathematics, i.e., it must meet the criterion that a formula representing a meaningful theorem is to be provable in the formal system just in case the theorem is true in classical mathematics.[9] In fact, since Hilbert was so restrictive in what statements of classical mathematics he considered meaningful, that is, since all meaningful statements could be reduced to finite disjunctions of inhaltliche statements, it followed that there was no difficulty in proving the formal analogs of all true meaningful statements. The question arose as to whether the formal system was too powerful, whether the formal analogs of false, but meaningful, statements of classical mathematics could also be proved in the formal system. This meant that all Hilbert had to do was to show that the formal system was consistent, i.e., did not contain any provable contradictions. Since the formal analogues of true meaningful statements were provable, if the system were consistent, then the formal analogues of the false meaningful statements could not be provable.

[9] He also required that the negation of the formula be provable if the theorem is false. Hilbert, in fact, chose the system of Principia Mathematica for his formal system.

If he could just prove the consistency of the formal system, he could save all of classical mathematics to the extent that it would never lead to a contradiction and, in particular, never contradict an inhaltliche statement.[10]

> If contradictory attributes be assigned to a concept, I say that mathematically the concept does not exist. . . . In the case before us, where we are concerned with the axioms of the real numbers in arithmetic, the proof of the consistency of the axioms is at the same time the proof of the mathematical existence of the complete system of real numbers or of the continuum. Indeed, when the proof for the consistency of the axioms shall be fully accomplished, the doubts which have been expressed occasionally as to the existence of the complete system of real numbers will become totally groundless.[11]

But how is the consistency of the formal system to be established? Hilbert could not use provability in another formal system to prove the consistency of the first, for this would only lead to a regress ad infinitum. The consistency proof had to be carried out using statements and methods which were meaningful or, as Hilbert termed them, "finitary." "Finitary" was

[10] The argument that the formal system should actually be identified with mathematics was that, once the system was shown to be consistent, it would resemble mathematics in not leading to any contradiction or contradict any of the (evident) elementary statements of mathematics and could be further tested by how well it solved problems for which it was not specifically designed, e.g., the continuum hypothesis. In any case, the meaningless part of mathematics was saved only to the extent that it was shown that 0=1 could never be proved from it.

[11] Quoted in Reid, p. 71.

defined as meaning that

> . . . the discussion, assertion or definition in question is kept within the boundaries of thorough-going producibility of objects and thorough-going practicality of methods and may accordingly be carried out within the domain of concrete inspection.[12]

Statements about formal proofs, which are concrete statements about concrete objects, could be checked in a finite number of steps, and were hence finitary. This finitary reasoning, unlike the axiomatic approach of mathematics, was based upon intuition. The formal system was considered as a formal model of the way we do mathematics. Which rules of deduction were admissable was to be decided by one's intuitions as to which were logically valid. Consistency was to be determined by how the formal operations adhere to the admissable operations of thought.

> For this formula game is carried out according to certain definite rules, in which the <u>technique of our thinking</u> is expressed. These rules form a closed system that can be discovered and definitively stated. The fundamental idea of my proof theory [metamathematics] is none other than to describe the activity of our understanding, to make a protocol of the rules according to which our thinking actually proceeds. . .[13]

Thus while the mathematics was purely formal, the metamathematics was purely intuitive. In fact, Hilbert turned to the

[12] <u>Ibid</u>., p. 156.

[13] <u>Ibid</u>., p. 186.

constructivist approach of Brouwer and Kant to make his metamathematics meaningful and intuitive. Weyl summarizes Hilbert's position:[14]

> It must have been hard on Hilbert, the axiomatist, to acknowledge that the insight of consistency is rather to be attained by intuitive reasoning which is based on evidence and not on axioms. But after all, it is not surprising that ultimately the mind's seeing eye must come in. Already in communicating the rules of the game we must count on understanding. The game is played in silence, but the rules must be told and any reasoning about its consistency, communicated by words. Incidentally, in describing the indispensable intuitive basis for his Beweistheorie Hilbert shows himself an accomplished master of that, alas, so ambiguous medium of communication, language. With regard to what he accepts as evident in this "metamathematical" reasoning, Hilbert is more papal than the pope, more exacting than either Kronecker or Brouwer. But it cannot be helped that our reasoning in following a hypothetic sequence of formulas leading up to the formula $0 \neq 0$ is carried on in hypothetic generality and uses that type of evidence which a formalist would be tempted to brand as application of the principle of complete induction. Elementary arithmetics can be founded on such intuitive reasonings as Hilbert himself describes, but we need the formal apparatus of variables and "quantifiers" to invest the infinite with the all important part that it plays in higher mathematics. Hence Hilbert prefers to make a clear cut: he becomes strict formalist in mathematics, strict intuitionist in metamathematics.

Hilbert realized that he had several difficulties to overcome in carrying out his program. First, he had to show that he could find an axiomatic system powerful enough to prove all of classical mathematics. This meant establishing a set of axioms

[14] Ibid., pp. 269-270.

and then manipulating them to demonstrate that the results of these axioms corresponded with the classical results of mathematics. Unfortunately, the body of classical results was logically intractable, since it did not correspond to any complete logical theory (in the technical, logical sense of completeness). Therefore the project of demonstrating that any axiomatic system corresponded with the theory of classical mathematics was purely empirical. One attempted to show that a particular axiomatic system adequately dealt with all the fundamentals of classical mathematics and with any portion of classical mathematics that appeared _a priori_ difficult to axiomatize.

However, Hilbert had to contend with a second and even more serious problem: the consistency of the system. An inconsistent set of axioms would imply all the theorems of classical mathematics; but it would imply all of their negations as well! Hilbert knew from his previous work on the foundations of geometry that, to prove the consistency of an axiomatic system, one had to go outside the system and discuss the developments in a metasystem. But how was one to be certain of the proof in the metasystem? As discussed above, Hilbert sought a constructivist metasystem which was intuitively consistent. However, the possibility of finding a metasystem which was constructive enough to be intuitively acceptable, but powerful enough to prove the consistency of a formal system for all of mathematics, was far from clear.

Thus Hilbert was left with the following technical tasks in order to carry out his formalist program. The list here is compiled by John von Neumann, one of those working on the project.[15]

1. To enumerate all the symbols used in mathematics and logic. Those symbols, called "primitive symbols," include the symbols '∼' and '→' (which stand for "negation" and "implication" respectively).

2. To characterize unambiguously all the conbinations of these symbols which represent statements classified as "meaningful" in classical mathematics. These combinations are called "formulas." (Note that we said only "meaningful," not necessarily "true." '1+1=2' is meaningful but so is '1+1=1,' independently of the fact that one is true and the other false. On the other hand, combinations like '1+→=1' and '++1=→' are meaningless.)

3. To supply a construction procedure which enables us to construct successively all the formulas which correspond to the "provable" statements of classical mathematics. This procedure, accordingly, is called "proving."

4. To show (in a finite combinatorial way) that those formulas which correspond to statements of classical mathematics which can be checked by finitary arithmetical methods can be proved (i.e. constructed) by the process described in (3) if and only if the check of the corresponding statement shows it to be true.

Hilbert found that it was too large a project to carry out by himself, so he delegated work to his graduate students and other colleagues. Throughout the 1920's and 1930's there were many

[15] John von Neumann, "Die formalistische Grundlegung der Mathematik," Erkenntnis 2 (1931), as translated in F. Benacerraf and H. Putnam, pp. 50-54.

partial results suggesting that the program would soon be successfully completed. Principia Mathematica, published by Russell and Whitehead in three volumes between 1910 and 1913, accomplished the first three tasks to the satisfaction of most mathematicians. The difficult part of the program to establish was point four, a constructive proof of the consistency of the formal system. While fragments of the system were shown to be consistent,[16] no one was able to show the entire system consistent. There was tremendous optimism, however, and everyone believed it was simply a matter of time before the consistency of the system was demonstrated and mathematics was given a secure foundation once and for all.

Godel's Incompleteness Theorem

The mathematics community was stunned in 1930 by Kurt Gödel's incompleteness theorems for first order logical systems. The first incompleteness theorem stated that, for any formal system S which is powerful enough to express elementary number theory, if S is consistent, then there is a formula F in S such that neither F nor its negation is provable in S. The effect of

[16] For example, such work by Skolem and Herbrand is discussed later in this chapter.

Gödel's incompleteness theorem was to show that there was no consistent formal system which had all the truths of arithmetic as provable theorems. This ended any possibility of successfully completing Hilbert's program. First of all, the condition governing Gödel's theorem that the system S be powerful enough to express all of elementary number theory did not provide a loophole for Hilbert's program, because any system in which one could deduce all the theorems of mathematics would have to allow expression of elementary number theory.

A second, and more germane, point is related to Gödel's second incompleteness theorem. This theorem states that if S is consistent, then the formula in S which states that S is consistent is unprovable in S ! The important point here is that the proof of the second incompleteness theorem is finitary in Hilbert's sense. Thus, using the stringent intuitive methods Hilbert required for metamathematics, Gödel showed that the consistency of such a formal system as Hilbert needed to express all of mathematics could be demonstrated to be unprovable! This was the final blow to Hilbert's program.[17]

[17] For a discussion of attempts to continue Hilbert's program even after Gödel's incompleteness theorems, see Charles Parsons, "Mathematics, Foundations of," in Encyclopedia of Philosophy, Vol. V, pp. 188-213.

The method Gödel used in proving his incompleteness theorems is of great importance here. How was Gödel able to use the intuitive metamathematical methods that Hilbert required and yet prove so powerful a theorem? Gödel developed the concept of recursiveness, the precise formal equivalent of the constructivity Hilbert required of metamathematics. Gödel then showed that all the formulas involved in discussing the consistency of a formal system were recursive. While others had worked with the recursive functions before (as will be discussed below), no one before Gödel had given a precise definition of the recursive functions and no one had shown so many functions to be recursive. Moreover, the importance and stunning character of Gödel's result attracted many mathematicians to study in detail the techniques Gödel used in his proofs.

Technically, here is what Gödel did in his paper[18] proving the incompleteness theorems. He began by developing what today is called Gödel numbering or coding. Natural numbers were assigned to sequences of signs and to sequences of sequences in his formal system P (the system that had been developed by Russell and Whitehead for Principia Mathematica together with Peano's axioms

[18] Kurt Gödel, "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I," Monatshefte für Mathematik und Physik, Vol. 38 (1931), pp. 173-198.

for arithmetic). These assignments were one-one and effective, i.e., given a number, it could be decided effectively whether that number corresponded to a sequence and, if it did, the sequence could actually be written down; and conversely, given a sequence, the number corresponding to it could be effectively calculated.[19] Coding was developed to reduce metamathematical problems to number-theoretic ones. In that way, description of the formal system could be reduced, to some extent, to representative statements within the formal language. This technique of coding non-numerical symbols into the integers is the crucial method for the discussion of formal systems and enters into all later decidability results. The method is also important for extending recursive definition to non-numerical objects.

Gödel then provided the first precise definition of the constructable functions. He called them "recursive functions." However, his initial definition was too restrictive and only included a subclass of what we now call the recursive functions. The class he defined is now called the "primitive recursive" functions. In his definition he made use of the notion of

[19] This coding was based on the unique decomposition of numbers into prime factors.

relative recursiveness, which had first been suggested by Hilbert:[20]

> A number-theoretic function $\phi(x_1,\ldots,x_n)$ is said to be <u>recursively defined in terms of</u> [relatively recursively defined by] the number-theoretic functions $\psi(x_1,\ldots, x_{n-1})$ and $\mu(x_1,\ldots,x_{n+1})$ if $\phi(0,x_2,\ldots,x_n) = \psi(x_2,\ldots,x_n)$, $\phi(k+1, x_2,\ldots,x_n) = \mu(k,\psi(x_2,\ldots,x_n), x_2,\ldots,x_n)$ hold for all $x_2,\ldots,x_n,k$.
>
> A number-theoretic function ϕ is <u>recursive</u> if there is a finite sequence of number-theoretic functions $\phi_1,\ldots,\phi_n$ that ends with ϕ and has the property that every function ϕ_k of the sequence is recursively defined in terms of two of the preceding functions by substitution, or finally, is a constant or the successor function $x+1$.
>
> A relation $R(x_1,\ldots,x_n)$ between natural numbers is said to be recursive if there is a recursive function $\phi(x_1,\ldots,x_n)$ such that, for all $x_1,\ldots,x_n$, $R(x_1,\ldots,x_n)$ if and only if $\phi(x_1,\ldots,x_n)=0$.

From this definition Gödel proved the following theorems about how to form (primitive) recursive functions and relations:[21]

> I. Every function (relation) obtained from recursive functions (relations) by substitution of recursive functions for the variables is recursive; so is every function obtained from recursive functions by recursive definitions according to schema (*).
>
> II. If R and S are recursive relations, so are $\neg R$ and $R \vee S$ (hence also $R \& S$).
>
> III. If the functions $\phi(x)$ and $\psi(y)$ are recursive. so is the relation $\phi(x) = \psi(y)$.

[20] Quoted in van Heijenoort, p. 602.

[21] <u>Ibid</u>., p. 602.

IV. If the function $\phi(x)$ and the relation $R(x,y)$ are recursive, so are the relations S and T defined by
$S(x,y)$ iff $(Ex)[x\leq\phi(x)\&R(x,y)]$ and
$T(x,y)$ iff $(x)\ [x\leq\phi(x)\rightarrow R(x,y)]$
as well as the function ψ defined by
$\psi(x,y) = \varepsilon x[x\leq\phi(x)\&R(x,y)]$,
where $\varepsilon xF(x)$ means the least number x for which $F(x)$ holds and 0 in the case there is no such number.

Gödel used the four theorems and the fact that $x+y$, $x\cdot y$, x^y , $x<y$, and $x=y$ are (primitive) recursive, with the recursion schema, to show that forty-five number-theoretic predicates are (primitive) recursive. Most of these predicates, such as Ax(x), which is by definition true just in case x is the code number of an axiom, were associated with metamathematical notions to be used in describing formal systems. By induction, with the application of the forty-five predicates, all primitive recursive number-theoretic predicates were shown to be numeralwise representable in the formal system P ,[22] i.e., a number-theoretic predicate holds of some given numbers just in case there exists a formula of P which is provable whenever the symbols in P representing the numbers are substituted for the free variables, those variables not restricted by a quantifier. This machinery was sufficient

[22] The formal system P was obtained essentially by superposing the logic of Principia Mathematica upon the Peano axioms, allowing constant symbols for the integers, and the successor relation as a primitive notion. Actually, any of a certain broad class of formal systems would have sufficed.

to prove the theorem, and Gödel showed that in any formal system P_k, where P_k was the system P together with any ω-consistent (primitive) recursive class k of additional axioms,[23] there is a proposition such that neither it nor its negation is provable in P_k.

Gödel made several remarks concerning the incompleteness proof which are relevant to the theory of recursive functions. He first noted that the proof of the incompleteness of P_k is constructive. Although only a few of the researchers were direct followers of Brouwer or Hilbert, there was a genuine concern for constructive procedures. In some sense, the recursive functions were supposed to be a formalization of the constructible functions. Throughout the incipient period of recursive function theory, there was a manifest concern with constructible definitions, proofs, and operations. Hilbert was especially interested in recursion, for he believed that it was a finite, constructible way of generating a broad class of the functions necessary for the development of mathematics. The early work on recursive functions is discussed in the next section of this chapter.

Gödel's second remark is even more important. He defined

[23] The notation "P_k" comes into use only later. Gödel gives the first definition of ω-consistency. A formal system is ω-consistent if there is no formula A(x) for which you can prove both $\neg\forall x A(x)$, "not for all x, A(x)", and each proposition A(0), A(1), A(2),....

a relation $R(x_1,\ldots,x_n)$ between natural numbers to be decidable if there is an n-place relation symbol r such that $R(x_1,\ldots,x_n)$ implies $r(\underline{x}_1,\ldots,\underline{x}_n)$ is provable in P_k, and not-$R(x_1,\ldots,x_n)$ implies $\neg r(\underline{x}_1,\ldots,\underline{x}_n)$ is provable in P_k, where $\underline{x}_1,\ldots,\underline{x}_n$ are symbols in P_k for the variables or constants $x_1,\ldots,x_n$ in classical mathematics, repsectively. He then asserted that it suffices for the existence of undecidable propositions in P_k that the class k be ω-consistent and decidable. In a note of 1934,[24] Gödel observed that the decidable predicates are just those which are general recursive.[25] This constituted the first definition of the general recursive functions. A more precise definition was given by Gödel in his 1934 Princeton lectures, which are discussed below.

The following section of Gödel's 1931 paper provided additional incompleteness results relevant to recursive function theory. Gödel first established that primitive recursive number-theoretic predicates are arithmetic, i.e., can be expressed as a

[24] "On the Length of Proofs," 1934, translated in Davis, The Undecidable, pp. 82-83.

[25] The general recursive functions are those which correspond in some formal way to the effectively computable functions. There are several alternative characterizations of the general recursive functions--all of them equivalent. For more detail, see the next three sections of this chapter on the 1936 papers by Church, Kleene, Post and Turing.

formula of first-order number theory.[26] In fact, he showed that if $F(x)$ is primitive recursive, then it is provable in P_k that $(x)F(x)$, "for all x, F(x)," is equivalent to an arithmetic formula. Since the previously demonstrated undecidable formula of P_k was of the form $(x)F(x)$ with $F(x)$ primitive recursive, P_k contains undecidable arithmetic propositions. Thus, Gödel had shown that in P_k there are undecidable number-theoretic predicates.

The Early Development of Recursive Functions

Gödel was not the first to be concerned with recursive functions. The idea of building up mathematical objects in a series of stages goes back to the Greeks. Almost every branch of mathematics in the nineteenth century used some algorithms or iterative procedures. However, there are differences between (i) using recursive functions in some particular context, (ii) basing a theory on exclusive use of recursive functions, and (iii) studying the properties of recursive functions themselves. Use (i) of recursive functions was commonplace in nineteenth

[26] As von Heijenoort points out, p. 594, this is stronger than showing that the number-theoretic predicates are numeralwise-representable.

century mathematics. However, it was not until the twentieth century and the concern over the foundations of mathematics that the second use of recursive functions developed. The third use was developed in the 1930's, partially as the result of Gödel's incompleteness theorems. No more need be said here about the first use of recursive functions, but the other two uses are relevant to the discussion and are considered in detail.

Even before Gödel's incompleteness results, the primitive recursive functions[27] had been used in foundational research. In 1919 Skolem wrote a paper[28] developing arithmetic in a new way in an attempt to avoid the difficulties of the paradoxes without adopting the cumbersome complexities of the theory of types. This arithmetic, now known as primitive recursive arithmetic, eschewed unbounded quantifiers and allowed bounded quantifiers as abbreviations only. Thus, he would not allow such propositions as

[27] The class of primitive recursive functions is the smallest class ξ of functions such that:

(1) all constant functions are in ξ;
(2) The successor function $f(x)=x+1$ is in ξ;
(3) All identity functions $f(x_1,\ldots,x_n)=x_1$ are in ξ;
(4) If $f,g_1,\ldots,g_K$ are in ξ, then so is $f(g_1(x_1,\ldots,x_n),\ldots,g_k(x_1,\ldots,x_n))$;
and (5) If h, g are in ξ, then so is f defined by $f(0,x_2,\ldots,x_k)=g(x_2,\ldots,x_k)$ and $f(y+1,x_2,\ldots,x_k)=h(y,f(y,x_2,\ldots,x_k),x_2,\ldots,x_k)$.

[28] "The Foundations of Elementary Arithmetic Established by Means of the Recursive Mode of Thought Without the Use of Apparent Variables Ranging Over Infinite Domains," 1923. Reprinted in van Heijenoort, pp. 302-333.

"for all s, B(x)," and such propositions as "for all x less than k, B(x)" were allowed only as abbreviations of B(0) or B(1) or ... or B(k-1) . Theorems were free-variable formulas (in which all variables occurred free of quantifiers), and new functions and relations were introduced by means of primitive recursive definition, with proofs following by mathematical induction. This schema is what Skolem called "the recursive mode of thought." Taking the notion of natural number, the successor function, and the recursive mode of thought as basic, Skolem recursively constructed addition, multiplication, subtraction, division, and the inequality relation, and then established elementary properties of arithmetic.[29] Next he showed that the greatest common divisor and the least common multiple are also recursive functions of two variables and that the relation of primeness is recursive.

In the course of Skolem's paper, two additional results concerning (primitive) recursive functions were given. First, the (ordinary) recursive schema of defining U(1) and U(n+1) in terms of U(n) was shown to be equivalent to the (course-of-values) recursion schema of defining U(1) and U(n+1) in terms of U(m) , for $m \leq n$. Second, Skolem provided an heuristic argument that

[29] For example, addition was considered a function of two variables, a and b , such that when b=1 , a+b is a+1 , i.e., the successor of a , and a+(b+1)=(a+b)+1.

Min(U,n) is recursive if U is recursive, where Min(U,n) means the least number k between 1 and n such that U(k) and has no meaning if not-U(k) for all k from 1 to n .[30] However, Skolem provided no formal definition of the class of recursive functions, nor were any of the properties of the class discussed.

Two of Hilbert's papers[31] on the foundations of mathematics provided an account of recursive functions as part of an attempt to prove the continuum hypothesis. The continuum hypothesis states, in effect, that there are no sizes of infinity between the sizes of the rational numbers and the real numbers. Besides being an important (then) unsolved problem of mathematics, Hilbert was also interested in it in order to use constructive techniques to prove something important about infinite sets. In his version of the continuum hypothesis, Hilbert attempted to show that there is a mapping of the ordinals of the second number class[32] onto the set of number-theoretic functions (instead of onto the real

[30] Skolem avoids Min(U) which is the least k such that U(k) since such a function presupposes the existence of an "actual infinity."

[31] "On the Infinite," 1925, and "The Foundations of Mathematics," 1927, reprinted in van Heijenoort, pp. 367-392 and 464-479, respectively.

[32] The second number class is the set of ordinals which are countable, but infinite.

numbers). His proof began with a metamathematical lemma based more upon his philosophical presuppositions than upon reasoned argument:[33]

> Lemma I. If a proof of a proposition contradicting the continuum theorem is given in a formalized version with the aid of functions defined by means of the transfinite |choice function| symbol ε (axiom group III), then in this proof these functions can always be replaced by functions defined, without the use of the symbol ε, by means merely of ordinary and transfinite recursion, so that the transfinite appears only in the guise of the universal quantifier.

Ordinary recursion defines a function of a number-theoretic variable by indicating what value it has for $m=0$ and how the value for $m+1$ is obtained from that for m. "The generalization of ordinary recursion is transfinite recursion; it rests upon the general principle that the value of the function for a value of the variable is determined by the preceding values of the function," and further remarks indicate that transfinite recursion also had for Hilbert the present meaning.[34]

Hilbert demonstrated next that substitution (of a new

[33] Reprinted in van Heijenoort, p. 385. The lemma arises because Hilbert is attempting to metamathematically justify the transfinite portions of mathematics. The attempted solution to the continuum hypothesis is apparently to demonstrate that every mathematical problem is solvable.

[34] Quoted in van Heijenoort, p. 386. Definition by transfinite recursion is given by defining what value the function has at $n=0$, how the value at $a+1$ is determined from the value at a, and how the value at b is determined from the values at c for $c<b$, where b is a limit ordinal.

variable or function) for an argument and (ordinary) recursion are sufficient to generate a broad class of new functions and subsume such function generating operations as defining a function up to a certain value, beyond which it is constant, or by definition in terms of elementary processes obtained from arithmetic operations such as the remainder in a division. Once this was completed, he was able to concentrate on the properties of these two operations, substitution and recursion. This study represented the first attempt to analyze the properties of "recursive" functions.

He first showed that there is no standard form for the recursion schema if one is limited to number-theoretic variables. He gave the following example. Let $\phi_1(a,b)=a+b$, $\phi_2(a,b)=a\cdot b$, $\phi_3(a,b)=a^b$, and so on for $\phi_4(a,b)$, $\phi_5(a,b)$, etc. That is, each subsequent function takes as its operation the next higher operation in the hierarchy: addition, multiplication (repeated addition), exponentiation (repeated multiplication), repeated exponentiation, etc. Then the function $\Phi(a)=\phi_a(a,a)$ (usually known as Ackermann's function) provided an instance of a function of an ordinary number-theoretic variable which can be generated by substitution and manifold, simultaneous recursion (on different variables at once), but cannot be generated by the standard techniques of ordinary, step-wise recursion on a single variable. He showed, however, that the Ackermann function $_n(a,b)$ could be defined recursively, if one allowed the following function variables:[35]

$$\iota(f,a,1) = a$$

$$\iota(f, a, n+1) = f(a, \iota(f, a, n));$$

$$\phi_1(a,b) = a+b$$

$$\phi_{n+1}(a,b) = \iota(\phi_n,a,b).$$

Another example provided an even more intractable recursion:[36]

$$\phi_0(a) = A(z)$$

$$\phi_{n+1}(a) = f(a,n,\phi_n(\phi_n(n+a))),$$

where A represented a known expression containing one argument, and f was a known expression containing three arguments. The difficulty here was that the value for $n+1$ could be obtained from the value for n. One must utilize the range of ϕ_n to determine ϕ_{n+1}. These difficulties were overcome by using variable types in a more general recursion schema. Variable types are variables of different types in the Russell sense of the theory of types. Variable types of height zero are constants, of height one are number-theoretic variables, of height two are variables of number-theoretic properties, and so on. The general

[35] Reprinted in van Heijenoort, p. 388.

[36] Reprinted in van Heijenoort, p. 389.

recursion schema then reads:[37]

$$p(g,a,o) = a$$

$$p(g,a,n+1) = g(p(g,a,n),n)$$

where a is a given expression of arbitrary variable type; g is a given expression of two arguments, of which the first is of the same variable-type as a and the second is a number (thus, the variable type of g must be of the same variable-type as a); and p , the expression defined by recursion, is also of the same variable-type as a . Specific recursions are obtained through substitutions. In effect, through these examples, Hilbert has demonstrated that there are recursive functions which are not primitive recursive and has exhibited a normal form for a wide class of "general recursive functions."[38]

The variable-types were the link that purportedly allowed Hilbert to correspond the number-theoretic functions to the numbers of the second number class. A natural correspondence arises since the way in which the numbers of the second number class increase in number is exactly analogous to the way in which the variable-types increase in height. A one-one correspondence between the numbers of the second number class and the number-

[37] Reprinted in van Heijenoort, p. 389.

[38] Hilbert does not use the term "general recursive function," but he does use the term "general recursive schema."

theoretic functions can be constructed in a complicated way which (roughly) involves constructing, for each ordinal β of the second number class, a recursive function p which subsumes all recursions of height β. These proposed functions p are the earliest known predecessors of universal functions.[39]

Unfortunately for Hilbert, his proof of the continuum hypothesis utilized transfinite recursion,[40] while his general recursion schema only provided for ordinary recursion. Thus, Hilbert postulated Lemma II on no better grounds than he had postulated Lemma I:[41]

> Lemma II. If by adducing a higher recursion or a corresponding variable-type we have formed a function that has only an ordinary number-theoretic variable as argument, then this function can always be defined also by means of ordinary recursion and the exclusive use of Z-types [ordinary number-theoretic variables].

In the 1927 paper, Hilbert admitted that Lemmas I and II had not been justified in the 1925 paper and so tried to provide a heuristic explanation of them. Lemma I, he claimed, was dispensable for the proof of the continuum hypothesis, although

[39] A function $\psi(x,y)$ is said to be universal if for all x and y, $\psi(x,y)=\gamma_x(y)$ if the γ_x constitute the (partial) recursive functions. More will be said of universal functions later in this chapter.

[40] Transfinite recursion has the steps of the operations indexed by all the ordinals, finite or infinite, while ordinary recursion is indexed by the finite ordinals alone.

[41] Quoted in van Heijenoort, p. 391.

useful in fixing the train of thought. However, he hoped to retain Lemma II. The difficulty, he pointed out, was in showing that when a sequence $a(n)$ of numbers of the second number class is given by a recursion

$$s(n+1) = \phi(a(n)) ,$$

where ϕ is defined by transfinite recursion, this transfinite recursion can be eliminated. Certain cases where this elimination had been effected were exhibited, and he showed that the ε numbers, those numbers α of the form $\alpha = \omega^{\alpha}$, while normally defined by transfinite recursion, can be defined by ordinary recursion. However, the number of cases examined was far from exhaustive.

In the 1927 paper, Hilbert also gave examples of recursively defined functions. He first mentioned the well known facts that sum, product, and factorial could be defined recursively. Without proof, he added that $\min(a,b)$ also could be defined recursively. Finally, he provided examples of two more complicated functions

$$\tau(a) = \begin{cases} 1 \text{ if } a \text{ is prime} \\ 0 \text{ otherwise} \end{cases}$$

and

$$\pi(a) = \text{the number of primes} \leq a$$

which can be defined recursively in terms of the functions

$$\phi(a,b,c) = \begin{cases} 0 \text{ if } b \text{ is equal to one of the numbers } 1\cdot a, \\ \qquad 2\cdot a,\ldots,c\cdot a \ (b>0), \\ 1 \text{ otherwise} \end{cases}$$

and

$$\psi(a,b,c) = \begin{cases} \text{the least of the numbers } 1,2,\ldots,a \text{ that are factors of } b \text{ and } >c, \\ b \text{ if none of these numbers has this property.} \end{cases}$$

Thus, Hilbert added to the list of recursively defined functions the functions min(a,b) and $\pi(a)$ (the others having been shown recursively definable by Skolem and others). In addition, Hilbert was the first to formally suggest relative recursive definition by showing $\tau(a)$ and $\pi(a)$ to be defined recursively in terms of $\phi(a,b,c)$ and $\psi(a,b,c)$.[42]

In 1928 Ackermann finally published his long-awaited paper defining the Ackermann iteration function.[43] In this paper he adopted Hilbert's general recursion schema and the variable-type distinction, with number-theoretic functions being typed according to the type of the variable. Functions formed according to the Hilbert schema of level one are primitive recursive. Ackermann

[42] Actually, the procedure of showing functions recursively definable by constructing them from other recursively defined functions was accepted practice among all those working with recursion. However, the practice of reducing the recursive definability of one function to the recursive definability of another without giving the actual recursive definition later became a common and convenient procedure. This notion of relative recursive definition could easily be used in demonstrating that if ψ is recursive, then so is θ, for functions ψ and θ, without knowing whether ψ is actually recursive.

[43] "On Hilbert's Construction of the Real Numbers," reprinted in van Heijenoort, pp. 493-507. This is the function given in Hilbert (1925).

showed, however, that not all functions recursively definable according to the Hilbert schema, when using functions of type one, are capable of recursive definition by exhibiting a function which required variables of type two. He defined $\phi(a,b,n)$ by

$$(a,b,0) = a+b \ ,$$

$$(a,b,n+1) = p_c(a,c,n),A(a,n),b) \ ,$$

where p_c , a function of type two, is defined by

$$p_c(f(c),a,0) = a \ ,$$

$$p_c(f(c),a,n+1) = f(p_c(f(c),a,n)) \ ,$$

such that c is a dummy variable indicating that f is a function of one variable and $A(a,n)$ is the type one function given by

$$A(a,0) = 0 \ ,$$

$$A(a,1) = 1 \ ,$$

$$A(a,n) = a \text{ for } n>1 \ .$$

If $\phi(a,b,n)$ were of level one, $\phi(a,a,a)$ also would be of level one—which he proved is false by demonstrating that it increases faster than any level one function. In the course of the argument he showed, however, that the Ackermann function is effectively calculable by the equations:

$$\phi(a,b,0) = a+b$$

$$\phi(a,b,n+1) = A(a,n)$$

$$\phi(a,b+1,n+1) = \phi(a,\phi(a,b,n+1),n) \ .$$

Nonetheless, this recursion did not fall under Hilbert's general schema because the recursion proceeded on two variables simultaneously. Thus, Ackermann's proof established that such multiple recursions need not be reducible to primitive recursion. The discovery of Ackermann's function prompted the classification of recursion schemas and, in particular, guided Herbrand in his definition of the notion of effectively calculable function.[44]

By 1930 recursive functions had been used in various foundational studies. A number of functions, especially of arithmetic, had been shown to be (primitive) recursive, but it had been demonstrated that there are effectively computable functions which are not primitive recursive. A general recursion schema had been given, which was later[45] shown to be equivalent to the present primitive recursion schema (of Gödel) if the functions were taken to be of type one. Multiple recursive definition had been shown to be more powerful than recursion on one variable. The notion of relative recursion had been introduced. Partial completeness results had been given for formal systems capable of expressing mathematics, and complete success was thought to be imminent. This is the foundation which Gödel shook in 1930

[44] See van Heijenoort, p. 494, for details. Evidently, Herbrand's definition was a generalization of Ackermann's multiple recursion definition of the Ackermann function.

[45] See R. Peter, 1934.

with his incompleteness result.

Attempts to Formulate a Discipline of Recursive Functions

Hilbert could possibly be given credit for being the first to examine the properties of the recursive functions. In his paper on the continuum hypothesis described above, he discussed in great detail the relationships between various recursion schemas and provided examples of recursive functions not really relevant to the main subject of the paper. Yet his research was intended only to solve a particular mathematics problem, not to characterize the properties of recursive functions. It was only in the mid-1930's that the properties of the recursive functions were studied in detail. Four independent characterizations of these functions were given by Kleene, Church, Turing, and Post. These were shown to be equivalent (mainly) by Kleene. A philosophical stance towards these functions was forwarded by Church. Research problems and techniques were identified and developed primarily by Church and Kleene. All of these developments turned the study of recursive functions into a mathematical discipline of its own. While the origins of the work of these four researchers are rather disparate, their research was related to Gödel's work on the incompleteness theorems.

The story, as it relates to Gödel, begins with a paper written by Herbrand in 1931 which provided a partial consistency result for the Hilbert program. Herbrand demonstrated the consistency of that fragment of arithmetic which can be developed if formulas in the induction schema are not permitted to have bound variables. In the course of the paper Herbrand defined the class of general recursive functions, illustrated the motivation that intuitionism provides to the study of recursive functions, and showed the relation of his consistency result to Gödel's incompleteness theorems.

In order to avoid a full induction schema, Herbrand allowed for the construction of a class of functions, provided that: the hypotheses governing the construction of the functions contained no variables bound by quantifiers; considered intuitionistically, the hypotheses made the actual computation of the functions possible for every given set of numbers; and it was possible to prove intuitionistically that a well-determined result could be determined. This provided another characterization of the general, rather than primitive, recursive functions. For example, Herbrand showed that the multiple recursion discussed by Hilbert (above) falls under his schema.

The important point abcut the above definition of the general recursive functions is its intuitionistic character. Herbrand was a follower of Hilbert and apparently only knew of intuitionism secondhand, through Hilbert.[46] As was common among Hilbert's followers at the time, the term "intuitionist" was used as much in the Hilbert sense of "finitary" as in the Brouwer sense of "intuitionist." Herbrand used the term "intuitionist" for the methods he considered admissable in metamathematics. Herbrand noted:[47]

> By an intuitionistic argument we understand an argument satisfying the following conditions: in it we never consider anything but a given finite number of objects and of functions; these functions are well-defined, their definition allowing the computation of their value in a universal way: we never state that an object exists without giving the means of constructing it; we never consider the totality of all the objects x of an infinite collection; and when we say that an argument (or a theorem) is true for all these x, we mean that, for each x taken by itself, it is possible to repeat the general argument in question, which should be considered to be merely the prototype of these particular arguments.

Thus the general recursive functions were intended by Herbrand as a model of the "intuitionist" functions.

In the addendum to his paper, Herbrand discussed his (partial) system of arithmetic in light of Gödel's incompleteness result. He first reconstructed Gödel's argument in order to demonstrate that it is "intuitionistically" admissable, and then proceeds to show that Gödel's method did not apply to Herbrand's system. This was

[46] See the discussion in van Heijenoort, p. 618.

[47] Quoted in van Heijenoort, p. 622.

accomplished by a diagonal argument which showed that there is no "intuitionistic" (general recursive) means of enumerating the "intuitionistic" (general recursive total) number-theoretic functions of one variable.

The importance of Herbrand's work surfaced in a series of lectures given by Gödel during 1934 at the Institute for Advanced Study in Princeton. While the lectures were intended to provide a precise and succinct account of the incompleteness results, they were important as well for providing Gödel's first clearly developed, precise account of both the primitive and general recursive functions. The definition of a general recursive function used in these talks was explicitly credited by Gödel to Herbrand.

Gödel began with a discussion of the (primitive) recursive functions. These functions were defined, for the first time, in terms of initial functions and operations on previously given (primitive) recursive functions.

> The function $\phi(x_1,\ldots,x_n)$ shall be compound with respect to $\psi(x_1,\ldots,x_n)$ and $\chi_i(x_1,\ldots,x_n)$ $(i=1,2,\ldots,n)$ if, for all natural numbers $x_1,\ldots,x_n$,
> (1) $\phi(x_1,\ldots,x_n) = \psi(\ _1,\ldots,x_n),\ldots,\chi_n(x_1,\ldots,x_n))$.
> (2) $\phi(x_1,\ldots,x_n)$ shall be said to be *recursive with respect to* $\psi(x_1,\ldots,x_{n-1})$ and $\chi(x_1,\ldots,x_{n+1})$ if, for all natural numbers $k, x_2,\ldots,x_n$,
> $\phi(0,x_2,\ldots,x_n) = \psi(x_2,\ldots,x_n)$
> and $\phi(k+1,x_2,\ldots,x_n) = \chi(k,\phi(k,x_2,\ldots,x_n),x_2,\ldots,x_n)$.
> We define the class of *primitive recursive* functions to be the totality of functions which can be generated by substituting, according to the scheme (1), and recursion, according to the scheme (2), from the

> successor function x+1, constant functions $f(x_1,\ldots,x_n)=c$, and identity functions $U^n_j(x_1,\ldots,x_n)=x_j$ $(1\leq j\leq n)$. In other words, a function ϕ shall be recursive if there exists a finite sequence of functions $\phi_1,\ldots,\phi_n$ which terminates with ϕ such that each function of the sequence is either the successor function x+1 or a constant function $f(x_1,\ldots,x_n)=c$, or an identity function, $U^n_j(x_1,\ldots,x_n)=x_j$, or is compound with respect to preceding functions, or is recursive with respect to preceding functions. A relation R shall be <u>recursive</u> if the representing functions is recursive.[48]

The (primitive) recursive functions, Gödel added, have the important property that, for any given set of arguments, the value of the function can be computed by a finite procedure. Similarly, (primitive) recursive relations are decidable, in the sense that, for a given set of natural numbers, it can be determined by a finite precedure whether the relation holds or does not hold.

Later in the paper, Gödel proved that the class of (primitive) recursive functions is not closed under multiple recursion. In particular, he showed that if $\psi(y)$ and $\chi(x)$ are given (primitive) recursive functions, then the function $\phi(x,y)$, defined inductively by the relations

$$\phi(0,y) = \psi(y)$$

$$\phi(x+1,0) = \chi(x) ,$$

$\phi(x+1,y+1) = \phi(x,\phi(x+1,y))$ is not, in general, (primitive)

[48] Kurt Gödel, "On Undecidable Propositions of Formal Mathematical Systems," Institute for Advanced Study Monograph, Princeton, NJ, 1934, pp. 2-3.

recursive. This left open the question of what is meant by "the class of all recursive functions." First, Gödel proposed a notion proffered to him in a letter by Herbrand:[49]

> If ϕ denotes an unknown function, and $\psi_1,\ldots,\psi_k$ are known functions, and if the ψ's and the ϕ are substituted in one another in the most general fashions and certain pairs of the resulting expressions are equated, then if the resulting set of functional equations has one and only one solution for ϕ , ϕ is a recursive function.

In order to make Herbrand's definition more formal and precise, Gödel made two restrictions. The first was that the left hand side of each of the functional equations defining ϕ shall be of the form:

$$\phi(\psi_{i1}(x_1,\ldots,x_n),\psi_{i2}(x_1,\ldots,x_n),\ldots,\psi_{iL}(x_1,\ldots,x_n)) .$$

The second was that, for each set of natural numbers $k_1,\ldots,k_L$, there shall be one and only one m such that $\phi(k_1,\ldots,k_m)=m$ is a derived equation, where the derived equations are defined inductively from the given set of functional equations by:

> (1a) Any expression obtained by replacing all the variables of one of the given equations by natural numbers shall be a derived equation.
> (1b) $\psi_{ij}(k_1,\ldots,k_n)=m$ shall be a derived equation if $k_1,\ldots,k_n,m$ are natural numbers, and $\psi_{ij}(k_1,\ldots,k_m)=m$ is a true equality.
> (2a) If $\psi_{ij}(k_1,\ldots,k_m)=m$ is a derived equation, the equality obtained by substituting m for an occurrence

[49] See op. cit., pp. 26-27.

> of $\psi_{ij}(k_1,\ldots,k_n)$ in a derived equation shall be a derived equation.
> (2b) If $\phi(k_1,\ldots,k_L)=m$ is a derived equation where $k_1,\ldots,k_L,m$ are natural numbers, the expression obtained by substituting m for an occurrence of $\phi(k_1,\ldots,k_L)$ on the right hand side of a derived equation shall be a derived equation.

This definition had the advantage that if $\phi(x_1,\ldots,x_L)$ is recursive, there is an arithmetical expression $A(x_1,\ldots,x_L)$ such that

$$\phi(x_1,\ldots,x_L) = y \text{ if and only if } A(x_1,\ldots,x_L,y).$$

In a later letter, Gödel remarked on Herbrand's contribution to the definition of "general recursive function":[50]

> I have never met Herbrand. His suggestion was made in a letter in 1931, and it was formulated exactly as on page 26 of my lecture notes, that is, without any reference to computability. However, since Herbrand was an intuitionist, this definition for him evidently meant that there exist a constructive proof for the existence and unicity of ϕ. He probably believed that such a proof can be given only by exhibiting a computational procedure. . . . So I don't think that there is any discrepancy between his two definitions as he meant them. What he failed to see (or to make clear) is that the computation, for all computable functions, proceeds by exactly the same rules. It is this fact that makes a precise definition of general recursiveness possible.

Thus, out of Herbrand's attempt to create an intuitionistic metamathematics arose Gödel's precise and formal definition of

[50] In a letter from Gödel to van Heijenoort in 1963, quoted in van Heijenoort, p. 619.

the notion of general recursive function.

Kleene and Rosser, graduate students at Princeton University at the time, were assigned the responsibility of taking notes from Gödel's 1934 lectures for the purpose of publishing a monograph. Rosser went on to strengthen Gödel's incompleteness theorems, while Kleene was responsible for developing Gödel's definitions of recursive functions into a mathematical theory. In 1936 Kleene published his first paper[51] on general recursive functions based on the Herbrand-Gödel definition.

The first section of Kleene's paper included several equivalent definitions of general recursive functions by specifying the forms of the equations and the admissable kinds of steps in the computation of a value, a normal form for general recursive functions and related results, and some general theorems about recursive enumerability. The second section of the paper considered which systems of equations define recursive functions under the general schema. It was demonstrated that the systems which do define recursive functions can not be recursively enumerated (under an appropriate Gödel coding of the systems into the natural numbers). This fact then was utilized to demonstrate, in a way different from Gödel's, the existence of undecidable number-theoretic propositions in formal logic satsfying certain conditions.

[51] Stephen C. Kleene, "General Recursive Functions of Natural Numbers," Mathematische Annalen, 112 (1936), pp. 727-742.

Kleene's definitions of primitive and general recursive were similar to those of Gödel.[52] A function which can be defined from the initial functions (successor function, constant function 0, identity functions) by a finite number of applications of substitution and ordinary recursion was said to be <u>primitive recursive</u>. Of the three equivalent definitions of general recursive functions given by Kleene, the one most similar to Gödel's read:

> The functions $\lambda_1,\ldots,\lambda_n$ are <u>defined recursively</u> by E if E is a system of equations in $\lambda_1,\ldots,\lambda_n$ such that for each i (i=1,...,n) and each set of numerals $k_1,\ldots,k_{s_i}$ there is exactly one numeral k (called the value of $\lambda_i(k_1,\ldots,k_{s_i})$) for which $E\vdash_{1;3}\lambda_i(k_1,\ldots,k_{s_i}) = k$.[53] A function λ_n is <u>recursive</u> if there is an E of this description.

Using this definition Kleene defined and proved several fundamental theorems concerning recursive enumerability.[54] He continued by

[52] Kleene uses the term "primitive recursive" for what Gödel calls "recursive."

[53] $A\vdash_{1,3}B$ means that B is provable from A , allowing members to be substituted for free variables and equals to be substituted for equals.

[54] A set is recursively enumerable if it can be effectively listed, i.e., if it is empty or is the range of a recursive function. See the Kleene paper cited in note 50 for details of theorems on recursive enumerability proved by Kleene.

proving the normal form theorem[55] for recursive functions, which characterized their structure by showing how they could be expressed in a certain logical "normal" form. This was undoubtedly the seminal paper in the theory of recursive functions, for it introduced the techniques and types of results that would be used in the field for the next thirty years.

However, one serious problem remained unresolved which precluded much interest in the theory of recursive functions. Just what was the substance of the theory? It was clear from Gödel's work that the recursive functions were useful in proof theory (metamathematics). It was also known that there was some connection between the recursive functions and the functions Brouwer would allow. But what precisely was the connection? The answer was first suggested by Alonzo Church, professor of logic at Princeton University (and advisor to Kleene and Rosser). To fully understand Church's answer, his work on the lambda-calculus must first be described.

[55] Theorem IV in the paper cited in note 50 states: "Every recursive function is expressible in the form $\psi[\varepsilon y(R(x,y))]$, where $\psi(y)$ is a primitive recursive function, $R(x,y)$ is a primitive recursive relation, and $(x)(Ey)R(x,y)$." $\varepsilon y(R(x,y))$ means the least y such that $R(x,y)$.

In papers of 1933 and 1934[56] Church attempted to formulate a set of postulates for the foundation of formal logic which would at once avoid the set-theoretic paradoxes and also avoid the artificial encumberences of the two other known solutions of the paradoxes, Russell's theory of types and Zermelo's axiomatic set theory. Church's plan was to develop a formal system which avoided the use of free (unbounded) variables and limited the use of the law of the excluded middle,[57] the two features he believed were reponsible for being able to construct the paradoxes in a formal system. In part, the reason for the restriction on free variables was that Church wanted every proposition and every function to have a precise, unambiguous denotation. For example, in the case of Russell's paradox, the set of all sets which are not elements of themselves has no precise, unambiguous denotation

[56] Alonzo Church, "A Set of Postulates for the Foundation of Logic," Annals of Mathematics, 33 (1932), pp. 346-366, and 34 (1933), pp. 839-864.

[57] The law of the excluded middle states that every proposition is either true or false--that there is no middle position. Although this position seems innocuous enough, Brouwer's intuitionism made a big point of denying this principle. According to Brouwer, a proposition was true only when you could provide a construction of its truth, and a proposition is false only when you can provide a construction of its falsity. Thus, for Brouwer, there are proposition (for which there is no construction) which are neither true nor false. See Chapter I for details. The Law of Excluded Middle is called the Principle of Excluded Third by Brouwer.

because it is impossible for S to be a member of S , but yet it is also impossible for S not to be a member of S . As Church stated:[58]

> One reason for avoiding use of the free variable is that we require that every combination of symbols belonging to our system, if it represents a proposition at all, shall represent a particular proposition, unambiguously, and without the addition of verbal explanations. That the use of the free variable involves violation of this requirement, we believe is readily seen. For example, the identity
>
> (1) $$a(b+c) = ab + ac$$
>
> in which a, b, and c are used as free variables, does not state a definite proposition unless it is known what values may be taken on by these variables, and this information, if not implied in the context, must be given by a verbal addition. The range allowed to the variables a, b, and c might consist of all real numbers, or of all complex numbers, or of some other set, or the ranges allowed to the variables might differ, and for each possibility equation (1) has a different meaning. Clearly, when this equation is written alone, the proposition intended has not been completely translated into symbolic language, and, in order to make the translation complete, the necessary verbal addition must be expressed by means of the symbols of formal logic and included, with the equation, in the formula used to represent the proposition. When this is done we obtain . . . [some formal symbolic expression] . . .. And in this expression there are no free variables.

Church recognized and explained the relationship of his work to Brouwer's intuitionism. Both placed restriction on the use of the law of excluded middle. Both insisted that each proposition

[58] _Ibid._, 33, p. 346.

be given a single, precise definition. However, Church differed from Brouwer in formulating his approach in a formal symbolic logic.[59] Church was particularly concerned about the free use made in mathematics of formulas and equations without concern for their domain of definition. Part of his solution involved the introduction of a new notation, $\lambda \underline{X} | \underline{M} |$, which denoted the function whose values are given by the formula M (which includes a precise statement of the domain of definition). This was the sole way of introducing a new function in Church's system. Church then provided a list of five rules of procedure and thirty-five postulates which were the only ways in which functions and propositions could be related in order to do mathematics.[60]

As Church began to work with his λ-calculus, as the new formal system was called, he was surprised at how many functions

[59] Brouwer argued that all formal systems are necessarily inadequate to express the truths of mathematics. Thus Brouwer was not impressed by Hilbert's formalist program.

[60] Church had to change the system in the 1933 paper because the 1932 system still admitted a form of the paradoxes. Church was not completely certain that the weakened 1933 system was adequate for the expression and proof of all of classical mathematics. Church's intention was to provide a secure logical system in which all of mathematics could be expressed and proved. He had no illusions, however, that mathematics would ever be carried out in his system. His system was just intended to show that there was a secure logical foundation for the mathematics already done in a more intuitive, less precise, classical framework.

could be defined essentially "from scratch" within his λ-calculus. Such functions he called λ-definable. After working with these λ-definable functions for some time, he came to believe that they provided him with an adequate characterization of those functions which one informally thinks of as effectively computable, computable in a series of steps that can be computed mechanically without ingenuity for all values of the variable once a calculation schema had been provided. In order to provide a more precise characterization of λ-definability, a series of definitions had to be given. The well-formed sequences of symbols in the λ-calculus were defined by the following rules.

(i) A variable x standing alone is well-formed.
(ii) If F and X are well-formed, F{(X)} is well-formed.
(iii) If M is well-formed, so is $\lambda x[M]$.

({F}(X) is, heuristically, a formalization of the function F(X) , while $\lambda x[M]$ is supposed to represent that function of x, M. Bound and free variables were defined then in the formal modern fashion by induction.) $S^x_N M|$ stood for the result of substituting N for x throughout the sequence M. There were then three admissable types of operations on sequences of symbols:

I. Replace $\lambda s[M]$ by $\lambda y[S^x_y M|]$ where y does not occur in M.
II. Replace $\{\lambda s[M]\}(N)$ by $S_N M$, provided that bound variables of M are distinct from x and from the free variables of N.
III. Replace $S^x_N M|$ (not immediately following λ) by $\{\lambda x[M]\}(N)$, provided that the bound variables of M are distinct both from x and from the free variables of N.

Any finite sequence of these operations was called a <u>conversion</u>. If B were obtainable from A by a conversion, then A was said to be convertible into B, or, in short, "A conv B". A formula was said to be in normal form if it was well-formed and contained no part of the form $\{\lambda x|M|\}(N)$. If the variables occurred in linear order (under the Gödel coding) without repetition after the signs in a formula of normal form, the formula was said to be in principal normal form. The normal form of a formula was unique up to applications of operation I. The positive integers were then represented in the λ-calculus in the following way:

→ab.a(b),
→ab.a(a(b)),
→ab.a(a(a(b))) , and so on.

A function F of one positive integer was said to be λ-definable if it were possible to find a formula F such that, if F(a)=r, then F(m)conv r . Kleene had already shown, in 1935, that each of a large class of important functions is λ-definable. In the case of any λ-definable function of positive integers, the process of conversion of formulas to normal form provided an algorithm for the effective computation of particular values of the function. The question of whether a particular formula F of the λ-calculus could be converted into normal form is the analogue to the question of whether F is recursive or not.

The problem with the characterizations of Kleene and especially of Church was that, while they were technically precise and accurate, they were not at all intuitive. Church was surprised when he was able to demonstrate that function after function was λ-calculable. Eventually, Church came to believe that his definition of a λ-calculable function and Kleene's various definitions of a general recursive function were co-extensive and that they both should be identified with the intuitive notion of an effectively computable[61] function. This belief became known as Church's Thesis and became a powerful methodological tool in the study of constructible functions.

No longer did one have to worry about the relative powers and interrelations of the various formal systems, as all were equivalent. Since they were equivalent, all of the theorems proved in the various formal systems could be grouped together as theorems about one common discipline. However, when a particular problem was broached, one could attempt to solve it, with equanimity, in the most convenient formal system. But, perhaps most important of all, Church's thesis provided logicians with an intuitive way of thinking about these functions. It was hard for one to have any intuition of the properties of the λ-calculable functions. Indeed, the originator of the definition was often

[61] An effectively computable or algorithmic procedure is a general procedure applicable to a certain class of symbolic inputs such that from each input will eventuate a symbolic output. The procedure must be deterministic (or mechanical) in the sense that the process of execution must be completely specified and unambiguous, and that there must be no need for creative imagination on the part of the "computer."

surprised by which functions were λ-calculable, and it was only after a good deal of work with the λ-calculus that Church's Thesis was voiced. However, researchers had fairly good intuitions about effectively computable functions. This allowed them to formulate ideas and proofs informally and only later to verify them in some convenient formal system. Much recursion theory was done in this manner then, and much continues to be done in this way today.

The Turing and Post Machine Characterizations of the Recursive Functions

Unfortunately, there was no way to prove Church's Thesis. The informal notion of an effectively computable function is imprecise, and there was no hope of giving it any precision without turning it into one of the specialized formal systems with which the logicians wished to prove it co-extensive. But unless it were fairly precise, there would be no *a priori* way to demonstrate that an effectively computable function has exactly the same properties as, say a λ-calculable function. Church believed that there were already ample empirical grounds for believing in Church's Thesis and that these grounds would be improved each time independent formal characterizations of these functions could be proved equivalent.[62] (Over time a number of independent and equivalent characterizations were formulated.)

Perhaps the whole issue would never have arisen and everyone would have accepted Church's Thesis on the basis of those functions Church and Kleene had shown to be or shown not to be formally computable (in their systems) had it not been for Gödel. Although he was a young man, Gödel's authority in logic was overwhelming, due to the stature of his completeness and incompleteness theorems. He was a stubborn, conservative man, not inclined to accept statements without adequate mathematical justification. Witness the formulation of the incompleteness theorem in light of the almost universal optimism over the possibility of successfully completing Hilbert's program. Not surprisingly, Gödel was not convinced of Church's Thesis. Of all the formal characterizations, Church's came under special attack. Being λ-calculable seemed to Gödel to have nothing to do with a function being computable in a series of stages. Gödel demanded a formal characterization which formally modeled the informal process of computing a function. It is in this regard that Alan Turing's work on computable numbers was so important. Turing provided a clear, intuitive precise

[62] Church, Turing and especially Kleene wrote articles providing laborious mechanical combinatoric procedures for translating the proofs and definitions in one formal system to those in another in order to establish the equivalence of the various formal systems.

formalization of the process of mechanically computing a number; and it was only with Turing's characterization that Gödel became convinced of Church's Thesis. In this light it is understandable why Church was anxious to have Turing come to Princeton when he heard about Turing's paper on computable numbers.

Turing, while an undergraduate at Cambridge, was fascinated by Riemann's conjecture and became interested[63] in actually calculating the real parts of the zeroes of the zeta function.[64] He designed a machine for the task and actually cut the teeth for the gearing mechanism himself in a Cambridge laboratory, although he never completed construction of the machine. This project made him wonder just which numbers in mathematics are mechanically computable. Upon reflection, he concluded that the mechanically computable numbers are exactly those which can be computed by a theoretical machine which he designed, now known as a Turing machine. Alonzo Church heard about Turing's work on computable numbers, recognized its importance to his own work on calculable functions, and offered Turing a fellowship to Princeton to work for a Ph. D. Although Turing had not done any significant work in

[63] This is mentioned in Gandy's obituary note on Alan Turing and is discussed briefly in Sara Turing's biography of her son.

[64] Apparently the machine was designed to test Poincare's famous conjecture that all of the real parts of the zeroes of the zeta function are equal to one-half.

logic prior to this, he saw the possibilities of his Turing machines for logic and matriculated in the fall of 1936. Turing was happy to go to Princeton, for at the time Princeton was the best research center in logic outside of Göttingen and perhaps the best mathematics research center in the United States. The distinguished logic faculty included Church at the university and Gödel and von Neumann at the Institute for Advanced Study.

While at Princeton, Turing was thoroughly trained in mathematical logic. He used this training to advantage in his famous 1936 paper, "On Computable Numbers,"[65] in which he described his Turing machine and demonstrated certain of its consequences for mathematical logic. This paper provides a rather different formal account of effective computability than those of Church and Kleene. The object of study for Turing was the set of computable numbers, those numbers which are computable by a machine--in particular, by his Turing machine. At first the goal was simply to characterize those numbers which could be computed in some mechanical fashion. However, the final published version of the paper, completed in the presence of the logicians at Princeton, contained many results important to logic, such as a proof of the existence

[65] Alan M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," Proceedings of the London Mathematical Society, series 2, volume 42 (1936-1937), pp. 230-265.

of a universal Turing machine, a proof of the negative solution of the Entscheidungsproblem,[66] and many observations relating his work to other work in logic by Church and Gödel.

Turing's machines consist of a one-dimensional tape, broken into squares, and a mechanical device capable of scanning and performing various operations on the tape. The machines operate in discrete time units. At any moment there is only one square being scanned by the machine. Each square is capable of bearing at most one symbol. Depending on the internal state (called the m-configuration) at the moment and on what symbol, if any, is in the scanned square, the machine can move the tape one square to the left or right, print or erase a symbol, or effect any (non-contradictory) combination of these operations. Given the m-configuration of the machine and the scanned symbol, the behavior of the particular machine is completely determined.

There are two kinds of symbols. 0 and 1 constitute symbols of the first kind, while all other symbols are of the second kind. Symbols of the first kind are used for input and output, while symbols of the second kind are reserved for internal use of the machines. If a machine never writes down more than a finite number

[66] The Entscheidungsproblem is the decision problem of interest in Hilbert's program, an effective procedure for deciding the provable theorems of a formal system. It was formulated explicitly by Hilbert and Ackermann in 1931.

of symbols of the first kind, it is called circular. Otherwise, it is called circle-free. Circle-free machines are those which are able to complete their computation and express their output in a sequence of symbols of the first kind. Circular machines are ones which result in an infinite repitition of the same finite set of symbols (a loop in modern parlance) or which stop before completing their computation.

Turing machines are designed to produce appropriately coded sequences of numbers as output, corresponding to the decimal expansion of the fractional part of a number. A sequence is said to be computable if it can be computed by a circle-free machine. A number is said to be computable if it differs by an integer from a computable sequence.

Turing believed that any number which is computable in this formal sense is computable in the intuitive sense of being able to write down the computation of the number with paper and pencil. Thus, the computable numbers comprised Turing's version of the recursive functions. Although Turing concentrated on the computability of numbers, his characterization amounted to the same as those describing computable functions. Just consider the computable number as the value of the computable function. For example, if Kleene were to show a function f to be recursive, Turing would have to show that f(b) is a computable number for suitable values of b.

For each different number to be computed, there was a different Turing machine. Thus, different Turing machines would be requi . to compute f(b) and f(c). Computable numbers were characterized by a description of the machine which computed the correlative computable sequence. Descriptions of the machine were expressed in a standard form as a sequence of quintuples. A quintuple $q_i s_j s_k L q_m$, for example, stated that, whenever the machine is in m-configuration q_i with scanned symbol s_j , it should erase symbol s_j , print symbol s_k , move one square to the left, and assume m-configuration q_m . These quintuples could then be coded into the integers (Gödel numbered) in such a way that each machine M(n) had a description number M . A number which was the description number of a circle-free machine was called a <u>satisfactory number</u>. The coding was such that, to each computable sequence, there corresponded at least one description number; while to no description number did there correspond more than one computable sequence. This proves that the computable sequences and numbers were enumerable.

Turing gave examples of several machines by specifying their quintuples. The most important of these was the universal computing machine U . If this machine U is supplied with a tape on the beginning of which is expressed (in code) the standard description of some computing machine M , then U will compute the same sequence as M . This was an elaboration of the notion of universal functions which was first hinted at by Hilbert in 1925. This idea, that an effectively computable function $\psi(x,y)$ can be found such that $\psi(x,y) = \phi_x(y)$, where the ϕ_x are the effectively computable functions of one variable, became a standard and important research tool in recursion theory.

One of the most important results of Turing's paper was the affirmation that a general (algorithmic) procedure for determining whether a given number is the description number of a circle-free machine does not exist. This was a direct analogue of Church's and Kleene's earlier discovery that the set of Gödel numbers for the recursive functions is not recursive, that is, that there is no recursive function which will give output 0 if b is in the set of Gödel numbers and 1 if b is not in the set of Gödel numbers, for input b. Like Church's and Kleene's, Turing's proof relied on a diagonal argument.[67] Turing's proof demonstrated similarly the unsolvability by reducing the problem to another problem which could be shown to be recursively unsolvable, i.e., for which there was no general algorithmic procedure for deciding the question. In this case, the reduction was to the

[67] The diagonal argument, originally devised by Cantor, was a powerful and often used tool in logic. The general technique of a diagonal argument was to form a new sequence from given sequences by manipulating the nth entry in the nth sequence, for each positive integer n. Turing's particular use of the diagonal argument is described in detail later in this section.

halting problem: whether there is an effectively computable procedure which will decide whether the computation of the nth decimal place in the nth computable sequence (under the ordering of the Gödel coding) will halt (compute to completion) or not. Reduction to the halting problem became the general procedure for demonstrating the recursive unsolvability of problems both in recursion theory and in the application of recursion theory to problems in other areas of mathematics.

Turing actually gave two proofs of this important result. The first was a strictly mathematical proof relying directly on the diagonal argument.

> If the computable sequences are enumerable, let a_n be the n-th computable sequence, and let $\phi_n(m)$ be the m-th member in a_n. Let β be the sequence with $1-\phi_n(n)$ as its n-th figure. Since β is computable, there exists a number K such that $1-\phi_n(n) = \phi_K(n)$ all n. Putting $n = K$, we have $1 = 2\phi_K(K)$, i.e., 1 is even. This is impossible. The computable sequences are therefore not enumerable.[68]

However, as stated above, Turing had already proved that the computable sequences are enumerable. Thus, there must be a fallacy in this proof, and Turing pointed to the assumption that β is computable.

> The fallacy in this argument lies in the assumption that β is computable. It would be true if we could enumerate the computable sequences by finite means, but the problem of enumerating computable sequences is

[68] Turing, *op. cit.*, p. 246. The following quotation is from the same page.

> equivalent to the problem of finding out whether a given number is the D.N. |description number| of a circle-free machine, and we have no general process for doing this in a finite number of steps. In fact, by applying the diagonal process argument correctly, we can show that there cannot be any such general process.
>
> The simplest and most direct proof of this is by showing that, if this general process exists, then there is a machine which computes β .

Although Turing realized that he had outlined an acceptable mathematical proof that there is no general algorithmic procedure for determining whether a given number is the description number of a circle-free machine, he recognized that he had not given an intuitive proof of the result. He accomplished this in a second proof, which showed exactly what went wrong when one attempted to construct a Turing machine for this task. His argument was as follows:

Suppose there were a machine D which, when supplied with the standard description N of any computing machine M(N) , would test N in a finite number of steps and either print "0" if M(N) were circular or print "1" if M(N) were circle-free. The postulated machine D would then print a sequence of 0's or 1's, each entry being completed in a finite number of steps. Thus D would be circle-free.

By combining machine D with the universal machine U , a circle-free machine H could be constructed to compute the sequence β' , whose <u>nth</u> entry was the <u>nth</u> entry of the <u>nth</u> computable sequence (under the Gödel coding). H would carry out its computation as follows: 0 would be entered into H . After a finite number of steps, H would print either "0" or "1." If H printed 0 , then 0 (the number entered in H) was not the description number of a circle-free machine. So, the next number, 1 , would be entered in H . This process would continue until some number n_1 was entered in H such that H printed "1." Then n_1 would be the description number of a circle-free machine $M(n_1)$. Next the universal machine would be used to compute $M(n_1)$, which is the same sequence as $U(n_1)$. Then H would print the first entry of $M(n_1)$ as the first entry of β' . $n_1+1, n_1+2, n_1+3, \ldots$ would be entered in H , repeating the process above until another number n_2 was found, such that H printed "1" with input n_2 . U would be used to generate the sequence $U(n_2) = M(n_2)$. The second entry of $M(n_2)$ would then be printed by H as the second entry of β' . $n_3, n_4, n_5, \ldots$ would be similarly located and H would print the j<u>th</u> entry in $U(n_j)$ as the j<u>th</u> entry of β' .

Since D and U were circle-free, so was H . H must have some description number K . Consider what would happen when H ran across the entry K in computing β' . First K would be placed in machine D . Since H is circle-free, D would print a "1" for K and so K would be entered in U . U would start to compute the sequence $U(K) = M(K) = H$. In other words, U would begin to compute sequence β' . U would have no trouble computing the first part of sequence β' , $\beta'_{n_1}\beta'_{n_2}\beta'_{n_3}\cdots\beta'_{n_p}$, where n_p is the largest number smaller than K which is the description number of a circle-free machine. But in order for U to print the next entry in β' , namely β'_k , it would have to go back and print the sequence for M(K) = H . When it tried to do this, it would get as far as printing $\beta'_{n_1}\beta'_{n_2}\cdots\beta'_{n_p}$. In order to print the next entry it would have to go back and first print M(K) . Thus, the machine would enter an infinite loop, repeatedly printing $\beta'_{n_1}\beta'_{n_2}\cdots\beta'_{n_p}$ in the attempt to print the (p+1)st entry of β', namely β'_K . This contradicts the fact that H is circle-free. Therefore the assumption was false that there is a circle-free machine D which would determine whether a description number is the number of a circle-free machine.

This method allowed Turing to demonstrate that Hilbert's Entscheidungsproblem is recursively unsolvable, that is, that there is no Turing machine which will determine which are the provable theorems of a formal theory. Using the above method, he first showed that there is no machine which, when supplied with the standard description of an arbitrary machine M , will determine whether M ever prints a given symbol. Then, for each machine M , a formula Un(M) was to be constructed for which, if there were a general effective procedure for determining whether Un(M) is provable, there would also be a general effective procedure for determining whether M ever prints 0 . Un(M) was to be a statement of the form "in some configuration of M , 0 appears on the tape." It could then be proved that 0 appears on the tape in some configuration of M if and only if Un(M) is provable. If there were a general effective procedure for determining whether M ever prints 0 . Hence, the _Entscheidungsproblem_ was shown to be recursively unsolvable.

Like Church's paper, Turing's is interesting for its attempt to demonstrate that the formal notion is equivalent to the informal notion of effective computability. He addressed the problem by considering what the possible processes are for computing a number. He then applies three types of arguments in order to justify Turing machine computability as the formalization of effective computability:

(a) direct appeal to intuition;

(b) proof of the equivalence of different formalizations; and

(c) examples of large classes of numbers which are Turing machine computable.

Appealing to intuition, Turing argued that his machine performs exactly those operations that the human mind does in the computation of a number, and that it has similar processes and limitations. It was this similarity of Turing machines to the actual human process of computing that convinced Gödel that Turing had an adequate characterization of the computable functions. The machine is the analogue of the mind of a human computer, and the tape is the analogue of his scrap paper. The scanned square is the only one of which the machine is "directly aware." The number of different symbols which may be printed is to be finite, for otherwise the differences between the symbols would fall below the level of human discernment. The internal "m-configuration" represents the "state of mind," and behavior at any moment is completely determined by the "state of mind" and observed symbols. The number of m-configurations is finite for a reason similar to the reason for finite restriction on the number of symbols. The operations of the machine correspond to the simplest operations of which a human computer is capable: writing down or erasing symbols, shifting attention right or left, or changing "state of mind." Thus, Turing contended that the Turing machine provided an adequate analogue to the human computation of a number.

Turing argued that if there were alternative formalizations of the notion of effective computability, a demonstration of the equivalence of these formal notions would provide important evidence that they coincide with the informal notion. Such equivalence preserved the integrity of intuitions; it provided evidence that they have mathematical substance and are free of inconsistency. Moreover, the intuitive appeal of various formal notions may be different, and equivalence maximized the intuitive appeal. In this vein, Turing demonstrated that if the Hilbert functional calculus were to be modified so as to be systematic and only involve a finite number of symbols, there would be a machine K , which would find all of the provable formulas of the calculus. An appendix to the paper sketched a similar result: the equivalence of λ-definability and Turing computability.

Turing's third argument was based on the exhibition of a large class of computable numbers. He demined the computable functions in terms of the computable variables and then showed (among other things) that:

(a) A computable function of a computable function of an integral or computable variable is computable;

(b) If $\phi(m,n)$ is computable and r is an integer, then $\eta(n)$ is computable, where

$\eta(0) = r$

$\eta(n) = \phi(n,\eta(n-1))$;

(c) the real algebraic numbers are computable;

(d) the real zeroes of the Bessel functions are computable; and

(e) the limit of a computably convergent sequence is computable.

Thus, he demonstrated that a large class of the informally computable functions of mathematics were Turing computable. Turing believed that these three arguments were sufficient demonstration that effective computability should be identified with Turing computability.

In 1936 Emil Post, mathematician at New York University, published a paper[69] presenting the new notion of a finite 1-process, which was strikingly similar to Turing's notion of computability. Although Post was aware of Gödel's and Church's results, he had not seen Turing's paper, which was at the publisher's about to be printed, when his own paper was submitted. Although the basic ideas of Turing and Post were so similar as to be redundant, Church, the editor of the journal to which Post's paper was submitted, decided to publish Post's paper anyway, together with a note explaining the independence of Post's and Turing's results.

The notion of finite 1-process consisted of two concepts: a symbol space consisting of a two-way infinite sequence of boxes ordinally similar to the integers, and a fixed, unalterable set of directions which direct operations in the symbol space and determine the order in which those directions are to be applied. The worker could work in but one box at a time, but could move from box to box. Each box was either empty or had a single, vertical stroke in it. The worker was able to perform the following primitive acts:

(a) Marking the box he was in (assuming it to be empty),

[69] Emil L. Post, "Finite Combinatory Processes, I," Journal of Symbolic Logic, 1 (1936) pp. 103-105.

(b) erasing the mark in the box he was in (assuming it to be marked),

(c) moving to the box on his right,

(d) moving to the box on his left, and

(e) determining whether the box he was in was or was not marked.

There was also a set of operations which were the same for each specific problem. They began with:

Start at the starting point and follow direction 1 .

This was followed by a finite number of directions numbered 1, 2, ..., n, with the <u>ith</u> direction having one of the following forms:

(A) Perform operation 0_i $[0_i = (a), (b), (c), \text{or } (d)]$ and then follow j_i ,

(B) Perform operation (e) and, according as the answer is "yes" or "no", correspondingly follow directions j_i' or j_i'' , or

(C) Stop.

Finite 1-processes were intended to solve general problems, each of which consisted of an enumerable number of specific problems ordered by the positive integers. A set of directions was considered <u>applicable</u> to a general problem if, in its application to specific problems, operation (a) was never ordered for marked boxes and operation (b) was never ordered for unmarked boxes. The set of directions was said to determine a <u>finite 1-process</u> if it were applicable to the general problem and if the process terminated for each specific problem. A general problem was called <u>l-given</u> if a finite l-process was determined which, when applied to the positive integers (the integer n given by marking n consecutive boxes to the right of the starting box), yielded, in a one-one fashion, the class of specific problems constituting the general problem. A finite l-process yielded a l-solution to a general problem if the answer it yielded for each specific problem was correct. If a general problem were both l-given and l-solved, a finite l-process could be found which gave the answer to each specific problem when the latter was represented by a number in symbolic form.

Post believed (but did not provide a proof) that his formulation was equivalent to general recursiveness. However, his purpose was not merely to provide any symbolic system with the ability to compute solutions to specific problems, but a system which had "psychological fidelity." Like Turing's, Post's system was supposed to model the way in which humans compute. Unlike Church, Post did not regard his formal notion as one to be identified absolutely with effective computability, but rather accepted this as a working hypothesis which he further confirmed by increasing the class of computations which had been shown to be l-finite.

Conclusions

By late 1936, four formal notions had been presented for the informal notion of effective computability: general recursiveness (Gödel-Herbrand-Kleene), λ-definability (Church), Turing computability (Turing), and 1-finiteness (Post). The question remained whether these notions were equivalent or not. If they were not, which was to correspond to the informal notion of effective computability? Or was this notion even consistent and formalizable mathematically? Papers by Kleene and Turing[70] answered this question. They used tedious combinatorial arguments to show how a definition in any one of these formal systems could be translated into a definition in any of the other formal systems. By the end of 1937, it was known that the various formal notions were equivalent. This confirmed the belief that the intuition of effective computability was sacrosanct, and that these formal notions adequately characterized it.

The papers of Kleene, Church, Turing, and Post constituted the incipient period of recursive function theory. The alternative

[70] Stephen C. Kleene, "λ-definability and Recursiveness," Duke Mathematical Journal, 2 (1936), pp. 340-353, and Emil L. Post, "Recursively Enumerable Sets of Positive Integers and their Decision Problems," Bulletin of the American Mathematical Society, 50 (1944), pp. 284-316.

formal notions of effective computability had been proved equivalent and there was confidence that these notions appropriately formalized intuitions. Precise, formal definitions had been given of primitive, general, and partial recursive functions corresponding to various informal notions of effective computability. The normal form theorem had given the relation between the primitive and general recursive functions and had provided a general characterization of the general recursive functions. General means had been obtained for building partial recursive functions from other partial recursive functions. General relationships had been established between recursive and recursively enumerable sets. There was a basic method of reduction established for showing the recursive unsolvability of specific problems in terms of the halting problem. These basic results and methods were organized into a coherent theory--one which could stand alone as a separate field of mathematics. Subsequent work on recursive functions utilized this basic theory to develop more general and refined techniques to deal with more general and subtle problems of effective computability.[71]

[71] Hartley Rogers, pp. 46-48, has a fairly complete classification of problems considered in the second stage of recursive function theory. A third stage perhaps could be delimited from the second stage by its concern with recursion on structures other than the natural numbers.

The benefits of this work on recursive functions to the development of computer science were already becoming apparent during the period Turing spent at Princeton. His characterization of the recursive functions, in terms of a machine that could do all the computations theoretically possible by a mechanical procedure, was particularly suggestive of the role recursive function theory could play in the development of computing machinery. For Turing's work not only characterized a particular class of functions in mathematics; it also characterized the theoretical possibilities and limitations for actual computations by a physical machine.[72] In fact, it was a question of actual computation (concerning the zeta function) that first had led Turing to consider the more theoretical issues regarding computation of functions. Turing maintained an active interest in actual computation by physical machinery throughout his career.

Even as early as his student days at Princeton, Turing argued vociferously that computing machines could be built which would adequately model any mental feature of the human brain. Von Neumann, who, as a member of the Institute for Advanced Study, had an office in the same building at Princeton, was attracted to Turing because of their common interest in mathematical logic.

[72] Turing's work also provided a characterization of the human mental process as it did computation. See Chapter Six for a detailed discussion of this point.

Turing's view on the computer and the brain was disputed by von Neumann, and the two discussed the issue on many occasions while Turing was completing his dissertation. This is purportedly[73] what inspired von Neumann's interest in computing. Von Neumann and Turing separated when Turing returned to England, leaving both determined to build computers to test the possibility of mechanically modelling the human brain. Both were restrained from beginning personal computer projects by the outbreak of the war, but, ironically, each was provided his entrée to the computer field on a grander scale than either could have managed individually because of the war.

Turing completed his Ph.D. dissertation[74] on ordinal logics at Princeton in 1938, resolving certain foundational problems brought about by Gödel's incompleteness theorem. Von Neumann offered the new Ph.D. a position as his assistant at the Institute. Characteristically, Turing is reported to have stated that, however he was to begin his career, it was not to be as assistant to anyone—even to von Neumann. Besides, Turing was homesick for

[73] In an oral interview with Rosser in April, 1979 by the author. The entire account of the interaction between Turing and von Neumann at Princeton in the 1930's is due to this discussion with Rosser.

[74] The dissertation was published as "Systems of Logic based on Ordinals," Proceedings of the London Mathematical Society, (2), 45 (1939), p. 161.

Cambridge and for the companionship of his mother. With the political situation in Europe steadily worsening in 1938, Turing wanted to be back in England when his country declared war. Thus he hastily completed requirements for his Ph.D. program, which he finished in two years, and returned to King's College, Cambridge, where his fellowship was renewed.

Chapter Three: Turing's Contributions to the Development of Physical Computing Machinery

This chapter will concentrate on Turing's contributions to the development of physical computing machinery. For several reasons, it is a rather difficult task to assess the novelty and importance of his work and sometimes even to determine what events transpired. For one, many of Turing's documents related to the war effort remain classified--even after the twenty-five year declassification of documents by the British government--and there does not appear to be any hope that this information will be made public in the near future. Second, Turing was terribly disorganized in general, an attribute which carried over to his correspondence. The records of his correspondence appear to be scanty and incomplete. Third, there is no universal opinion about the importance of Turing's work to the development of computer science as there is, for example, in the case of von Neumann. Some people consider Turing's work to be the thoroughly impractical efforts of a mathematical logician who know little about engineering problems and design and do not give him his due. Fourth, Turing committed suicide during the most productive period in his career. Thus he was not around to carry through his projects and ideas, nor did he have the time to communicate them to other people in computer science. Fifth, the fact that Turing was working in Britain reduced the effect of his work. Although there was a surprisingly active computer industry in Britain just after the war considering the state of the British economy, very quickly the United States took dominance over the industry. In the United States the plans of Eckert, Mauchly, and von Neumann held dominant, and very little was adopted from foreign computer development. Nevertheless, it is possible to patch together, from the fragmentary evidence, an account of the development of Turing's work and to conjecture on its influence on the development of computer science.

Bletchley Park and Computing

In 1938, after completing his Princeton degree, Turing returned to Cambridge, where his fellowship at King's College was renewed. Not long thereafter war was declared, and Turing immediately volunteered. He was recruited by the British Foreign Office and was sent to Bletchley Park in Buckinghamshire, some fifty miles north of London. Bletchley Park was the center of British cryptological work during the war. Turing was assigned to one of the groups working on breaking the codes produced mechanically by the German military.

Most of the German mechanical code production was done with a machine known as Enigma. In general shape and size, the Enigma machine was similar to an ordinary typewriter. Like a typewriter, it had an alphabetic keyboard similarly situated in front. However, Enigma had as well a plugboard on the front, a set of lights on top to represent the letters of the alphabet, and a set of three rotors in the rear--all of these connected inside by intricate electrical wiring. Striking a key on the ordinary keyboard would create an electrical contact, and an electrical signal would follow an extremely complicated pattern through the machine, through the rotors, and would eventually light up one of the lights (for a particular letter) on the face of the machine. Each time an electrical impulse would go through a rotor, the rotor would automatically advance through one of its twenty-six positions (each position providing a different electrical path). Furthermore, the rotors could be interchanged in position or be replaced by other rotors, or the plug board on the front could be changed to change the electrical circuit. To encode a message with the machine, one would set the plug board and the rotors in the appropriate initial position for the day (for the initial keying positions were changed daily) and type the message on the keyboard. Then another person, on a normal typewriter, could type the letters that appeared on the lights on the face of the machine. To decode a message, the machine need only be set in essentially the inverse position (depending on the day's key) and have the above coding procedure be followed.

In 1919 a Dutchman, Hugo Alexander Koch, registered a patent for a Geheimschrijfmachine (secret writing machine) for protecting industrial secrets.[1] Although these plans embodied the basic idea of Enigma, the machine awaited practical improvements before it could become commercially feasible. A Berlin engineer, Dr. Arthur Scherbius, completed these practical improvements and was placed on the board of directors of Chiffriermaschinen Aktiengesellschaft (Cipher Machines Corporation) of Berlin, a company founded solely to manufacture these machines which Scherbius named "Enigma."[2]

These commercial machines, which were very similar to the later military versions, were well advertised. The company exhibited the machine before the 1923 Congress of the International Postal Union. The following year the company arranged for a demonstration

[1] The most reliable secondary source on the war and Enigma seems to be R. Lewin, Ultra Goes to War. With the declassification of information by the British government after thirty years, beginning in 1974, a slew of books--mostly unreliable--deluged the market.

[2] According to one source, the unreliable Cave Brown's Bodyguard of Lies, 1975, p. 14, Scherbius named the machine after Sir Edgar Elgar's Enigma Variations in which the composer described his friends in a musical cipher.

of the machine, having the German postal office send a message over Enigma to the International Congress. A sales pamphlet was distributed in English, entitled, "The Glow-lamp Ciphering and Deciphering Machine 'Enigma.' What persons is it calculated to serve and how?" The machine was publicized in the United States in Radio News[3] and was described in print by Dr. Siegfried Turkel, scientific director of the Viennese criminological institute. Besides the German patent, Scherbius applied for a British patent in 1927 and for another for improvements on the machine in Britain in 1931.

Unfortunately for Scherbius, the time was not ripe for his machine. World business had slackened dramatically by 1930, and industrial espionage using sophisticated equipment was not yet prevalent. Scherbius' company went bankrupt and lost the patent rights. However, this was not before the governments of Germany, Poland, Japan, and the United States had purchased models of the machine.[4] In 1933, just as Hitler was coming to power, the commercial model was withdrawn from the market.

As early as 1926 the German military began to use a modified version of the Scherbius machine. Poland, sandwiched between

[3] Radio News, 1923.

[4] The United States bought a Scherbius' machine for about one hundred fifty dollars in 1927.

powerful countries, Russia and Germany, realized its vulnerability. It countered its relative weakness by strong intelligence operations designed to determine what its powerful neighbors were doing. Consequently, Poland was uneasy when the Germans employed their new encoding device for messages which the Poles were unable to decipher. This motivated a concerted new effort to break the Enigma coding. Mathematics students from the University of Poznan were recruited and provided with additional instruction in cryptanalysis.[5] This group worked on building a copy of the German military version of Enigma, on a mathematical theory for breaking its coding procedures, and on supplementary machines to assist in the code-breaking computations. Their work was made easier by Polish intelligence:

> During this year [1928] Polish intelligence also had a German military Enigma in their hands for a weekend. A box addressed to the German Embassy in Warsaw was tactfully removed from the Railway Parcels Custom Office one Friday afternoon and returned, after examination, before the next Monday morning.[6]

The Poles were also assisted by the Intelligence Division of the French General Staff. An officer in the German Cipher Bureau in

[5] Ironically, one of the students was sent to Göttingen, the center of Western mathematics, to gain the mathematical background to break the German code.

[6] Quoted in Lewin, op. cit., p. 30, fn., and attributed to Polish Colonel Lisicki.

1932 provided the French Secret Service with documents on the operations and procedures of German military encoding. Included was a list of the settings for the Enigma machine. In turn, the Poles were supplied with copies of these documents. Using the code keys provided by the French and the messages they had intercepted during the previous year, the Poles were able to develop a mathematical technique for breaking the Enigma code. By 1939 the Poles understood the workings of Enigma, had had fifteen Enigma machines built in the Ava radio equipment factory in Warsaw, and were regularly decoding field messages of the German military once again.

Unfortunately, 1939 was the year in which Poland was overrun by Germany. During the invasion, the members of the cryptographical section were secreted from the country and reestablished as a unit in France. However, their effectiveness in France was hampered severely by the oppressive authority the French military insisted on exerting over the Polish officers. However, afraid of being captured, the Poles had already presented to both the French and the British working models of the German military Enigma, plans for other related machinery, and information about the theory for breaking the coding procedure. As France came under siege as well, the responsibility for breaking Enigma lay with the British.

Dillwyn Knox, a brilliant Eton and King's College, Cambridge, man, was head of the British program to break Enigma. The Government Code and Cipher School (GC&CS) was moved from the London headquarters of the secret service to a small Buckinghamshire town, Bletchley Park, fifty miles north of London. Bletchley Park was chosen because it was away from the strategic bombing targets in London, had ample room for expansion, was linked by major highway and railway to London, and was convenient to Oxford and Cambridge, the major hunting-grounds for recruits to work at Bletchley Park.

The "old boy network" was used to perfection in staffing Bletchley Park. There were already strong connections with Cambridge through the older cryptanalysts: Knox from King's, Welchman from Sidney Sussex, Vincent from Corpus Christi.

> On a personal basis, therefore, suitable academics at Cambridge would be approached by friends already at BP: or, by private arrangement, someone would go down to one of the colleges to interview a mystified group of undergraduates picked out as possibles by a reliable tutor. . . . Cambridge serves as an exact example of the word-of-mouth method that was employed--and perhaps inevitably employed--to staff Bletchley Park in secrecy.[7]

At the beginning of the war, Bletchley Park had the pick of the country's talent. They chose people skilled in solving abstract puzzles: mathematicians, classicists, translators of languages,

[7] _Ibid._, p. 56.

chess masters. Turing was recruited personally by Knox at the beginning of the war to be trained as his assistant.

The British continued in the three directions first traversed by the Poles: the construction of additional Enigma machines, an elaboration of the mathematical theory for breaking the code, and the improvement and construction of devices for the actual breaking of the Enigma code. Before the details of this enterprise can be described, a description of Enigma and its operation must be provided.

There were three difficulties to overcome in breaking the Enigma coding procedure. First, essentially a different code was used for each letter of an Enigma coded message for, each time a letter was struck, the rotors advanced one position, thus providing a different permutation of the alphabet and so a different code. This meant that the ordinary statistical distribution of letters technique for decoding was useless until some technique was derived for reducing an Enigma cipher into a cipher consisting of a simple permutation of the alphabet. By 1943, each machine had six possible rotors, with three being placed on the machine at any one time. Thus, there were 120 possible rotor permutations. Then each rotor had twenty-six possible positions. Many changes were also possible through the plug-board. One worker at Bletchley Park estimated that there were on the order of 10^{20} possibilities to check for each message! Although this could be done, it would take days or months to check all possibilities by brute force and, by this time, the messages no longer would be of intelligence significance. Thus, it is not surprising that at no time during the war did the Germans believe that there was any chance of the Allies breaking Enigma.

The work at breaking the code continued at Bletchley Park. Enciphered Enigma messages were intensely studied for clues. Attention was focused especially on Luftwaffe messages, as the members of the Luftwaffe tended to be more arrogant and less concerned with security in their enciphering.[8] However, an even more fruitful route was the improvement and development of devices to assist with computations in the breaking of the codes. The British inherited from the Poles a machine called the "bomba" which they radically improved, and from which they generated a new series of electro-mechanical machines, known as the Robinson series, and finally an electronic machine known as Colossus. The details concerning these machines as well as their exact functions remain secret, although a vague picture can be drawn.

The Polish bomba was significantly improved at Bletchley Park

[8] Humorously, the greatest boon for the British cryptographers was the use of "four letter words." As part of the encoding procedure, the Germans were instructed to transmit four random letters. However, altogether too often, their "random letters" were four letter invectives--a fact of great utility in the breaking of the code.

and called a "bombe."[9] It was probably the first large, fast cryptanalytic machine.[10] The bombe is best described as a special purpose electromechanical data processor. In a sense, the machines were like universal Turing machines which could imitate any of the various Enigma machines. Harold Keen, chief engineer for the construction of bombes for Bletchley Park, stated the function of the bombes:[11]

> What it did was to match the electrical circuits of Enigma. Its secret was in the intermal wiring of Enigma's rotors, which 'the bombe' sought to imitate.

The Polish bomba had parts of six rotors wired into it which enabled it to imitate the permutations created by the three

[9] For some obscure reason, the machine was named after the Italian frozen dessert. A number of people at Bletchley Park objected to the name "bombe" on the grounds that messages might be intercepted by the enemy and they would confuse "bombe" with "bomb" so that Bletchley Park might become a target of German bombing raids.

[10] I. J. Good, "Early Work on Computers at Bletchley," Annals of the History of Computing, 1 (1979), p. 43.

[11] The contract for the construction of these machines was given to the British Tabulating Company of Letchworth. Keen is quoted in Lewin, op. cit., p. 58.

rotors of Enigma.[12] The bombes provided a way much faster than was possible by the human unaided by machine "to test all the possible wheel or rotor orders of the Enigma, all the possible wheel settings and plug or Stecker connections to discover which of the possible arrangements would match a prescribed combination of letters."[13] The end result would presumably be tested to determine whether the code had the correct statistical distribution to be in German, for it would be impossible for humans to examine each of the 3 x 10^8 messages. However, the fact (that made the Germans so confident) was that, in a reasonable time, even the bombe or a series of bombes was unable to test all the possibilities. Thus a slightly different strategy had to be used—a strategy partially due to Turing.

> It obviously could not be sufficient merely to simulate the Enigma and to try all possible setups for a message, because no machine even now would be capable of running

[12] There seem to be two alternative reasons why there must be parts of _six_ rotors wired into the bombe. According to one view, it is because there are six ways in which the three rotors can be placed in the three positions in the machine. According to the other view, the way the circuits were actually constructed on the Enigma, the electrical impulse would go through the three rotors, be reflected, and return through the rotors in the reverse order. Thus, the rotors would provide six permutations which the bombe would have to imitate. All parties seem agreed that the bombe was unable to cope with the possibility of there being extra rotors which could be placed in the Enigma.

[13] _Ibid._, p. 119.

> through the 3 X 10^8 possible states[14] in a reasonable time. So there had to be some further ingenuity in the bombe. This I cannot describe, but I can only say that Gerdon Welchman had one of the basic ideas and Turing another one. My impression is that Turing's idea was one that might not have been thought of by anyone else for a long time and it greatly increased the power of the bombe.[15]

Turing's contribution to the technical improvement of the bombe remains a mystery. However, in the vaguest sense, the technique for the use of the machine is apparent. By human observation of the incoming coded messages, using a combination of mathematical theory and rule of thumb methods, a select few messages were chosen to be fed in part to the bombe. Since bombes were scarce, these "menus" to be fed to the bombes had to be chosen carefully because there was no a priori way of determining how long the bombe would run before it would halt with a possible solution. Each time the machine would halt, a jubilant staff would manually check to see that the bombe had identified the appropriate coding.

Besides the bombes, Bletchley Park produced two other series of machines for cryptological purposes, the Robinson series and the Colossus. The work on these machines began late in 1942 when

[14] Also see Good, p. 42, who estimates there to have been between 10^{21} and 10^{23} possibilities to check (depending on the model).

[15] Quoted in Lewin, pp. 58-59, from Good.

M. E. A. Newman joined the Bletchley Park staff and was granted permission to open a new section for the design of new types of cryptological machinery. Newman, who had known Turing at Cambridge and even had collaborated with him on work in logic,[16] looked to Turing for new ideas in the construction of these machines: "Heath Robinson," Peter Robinson," and "Robinson and Cleaver."[17] The machines were built by the Post Office research center at Dollis Hill in North London,[18] which had extensive experience with radar. The purpose of the Robinson series was rather different from that of the bombes.

The Robinson machines were fully automatic machines which electronically counted pre-programmed Boolean functions of two inputs. The Heath Robinson read input from two paper tapes, with five holes across, at 2000 characters per second (a speed still fast for modern readers).[19] The tapes in this machine were driven

[16] Newman was a Fellow of St. John's College, Cambridge, a University lecturer in mathematics and a Fellow of the Royal Society. He was later to sponsor Turing's entrance into the Royal Society and to write Turing's biographical memoir.

[17] There machines were named after Heath Robinson, Britain's version of Rube Goldberg, and two London department stores, Peter Robinson and Robinson and Cleaver.

[18] To keep the purpose of the machine secret, the machine was described as "Transmitter, Telegraph, Mark I" to the people building it.

[19] Good gives these details, p. 45, about Heath Robinson.

by sprocket holes and pulleys. The processing of the information on the tape was accomplished by photoelectric readers and electronic circuits, carried out by between 30 and 80 electronic valves. Output was handled by a primitive automatic line printer. There were difficulties keeping the quickly moving tapes synchronized, and the actual utility of the Robinson series during the war was slight. Their main importance was as precursors to the Colossus machines.

> This machine |Heath Robinson| had been put together with emphasis on speed of completion more that on reliability. In fact, so many things could go wrong, especially with the machine and tape preparation, that the success rate was extremely low and discouraging for some weeks, and the future of the Newmanry was perhaps in the balance. . . . By introducing more checks into the entire system and also by other research carried out by Michie and myself |I. J. Good|, the success rate was improved enough to show the feasibility of Newman's faith in a machine attack on the problem in hand. Thus funds were made available for a more powerful, namely the Colossus.[20]

The actual function of the Robinson and Colossus series remains obscure. However, one writer on intelligence speculates on their use.[21] As the war progressed, the Germans worked on developing a new machine, the Geheimschreiber, which would render the codes for their strategic plans and diplomatic assessments virtually

[20] Good, p. 45.

[21] Lewin, op. cit., cf. pp. 129-134.

unbreakable over any period of time.[22] According to this theory, the Germans entrusted to Enigma tactical messages which would be out of date, they felt, by the time they could be broken. However, strategic and diplomatic plans were of on-going interest and would not be rendered useless by the time span necessary to decode an Enigma message. The extra security of the *Geheimschreiber* over the Enigma derived purportedly from three sources. First, the machine was much more rapid than Enigma, sending messages at a rate of about a word a second. Second, the machine consisted of roughly ten rotors, instead of the three rotors on the Enigma. Third, and perhaps most important, the *Geheimschreiber* was entirely automatic. The operator would type the message he wanted coded and the machine would then completely take over, automatically coding and transmitting the message. On the other end, the decoding was automatic and rapid as well. The security was much better because there was no chance of carelessness on the part of the cipher clerks--carelessness which often assisted the breaking of these codes at Bletchley Park.

Unfortunately, the speed of the *Geheimschreiber* required speed on the part of the machines used to break *Geheimschreiber*,

[22] The existence of the *Geheimschreiber* is not speculative, however. Two models were captured in fighting in North Africa. These machines were made by the famous makers of telephone and telegraph equipment, *Siemens und Halske Aktiengesellschaft*.

and the electro-mechanical Robinson machines were not fast enough. Thus a completely electronic machine, known as Colossus, was developed to match the speed of the Geheimschreiber. The project was jointly assumed by Bletchley Park and Dollis Hill. It was T. H. Flowers, head of the switching group at the Post Office's Dollis Hill research Station in North London, who suggested that in Colossus electromagnetic relays be replaced by fifteen hundred electronic tubes. The electronic complexity of such an undertaking was so overwhelming that no one at Bletchley Park would authorize the project; so Flowers had to go to the Director at Dollis Hill for authorization. The work at Bletchley Park on the machine came under Newman's group, as many of the features of the Robinson machines were incorporated into the Colossus design. Turing was available to Newman in an advisory capacity as he had been for the Robinson series. In the short span of eleven months the first Colossus was built and was in operation by December 1943. A number of others were completed for use, with an increase in power to 2400 tubes by June 1944 for use on D-Day.

According to Randall,[23] the chief historian of the subject, the Colossus incorporated the following technical features: 1500

[23] Brian Randell,"Colossus: Godfather of the Computer," New Scientist 73, (10 Feb. 77), pp. 346-348.

tubes (more than twice as many as most of the early post-war computers), operation in parallel arithmetic mode at 5000 pulses per second, paper tape inputs moving at more than 5000 characters per second, electronic counting circuits, binary arithmetic and Boolean logic operations, "electronic storage registers changable by an automatically controlled sequence of operations," conditional (branching) logic, "logic functions pre-set by patch panels or switches, or conditionally selected by telephone relays," and typewriter outputs.

Flowers, the main engineering designer of Colossus, solved the problem of tape alignment that had plagued the Robinson machines by replacing some electro-mechanical operations by fully electronic ones—thus moving one step closer to the fully electronic machine.

> Flowers' idea was to bypass the tape alignment problem entirely by dispensing with one of the input tapes, and instead generating some of the input data electronically within the machine. Means therefore had to be provided for generating the required sets of data from parameters stored on rings of gas-filled thyratron triode valves. This was done by incorporating plugboards and sets of switches into the machine similar to those which were used in an earlier electromechanical device for preparing input tapes for the Heath Robinson.[24]

However, by resorting more to pure electronics and thereby increasing the number of tubes (to 1500), probably the largest number used in any machine up until that time, a difficult

[24] Randall, p. 347.

problem arose over reliability. Flowers worked out a design so that switching circuit networks with even a large number of tubes could be made reliable. The design involved: leaving the equipment on permanently to avoid the power surge when the tubes were turned on (a main cause of their blowing out); use of a clock pulse to synchronise and time operations so as to eliminate cumulative timing errors within such a large machine; and use of binary tube circuitry on a large scale by having all tubes (other than in the tape reader photo cell amplifier) operate at either zero voltage (to represent binary digit 0) or above a minimal voltage level (to represent 1), rather than operate at a continuous range of voltages.

Later versions of Colossus were five times as fast as the original model, mainly due to the use of shift registers (temporary memory storage) which allowed a degree of parallel processing. The later versions had other additional facilities, and the number of tubes grew to 2400.

Randall provided an assessment of the Colossus as a precursor of the modern electronic digital computer:

> A proper assessment of the Colossi as precursors to the modern general purpose electronic digital computer is hampered by the lack of detailed information concerning the functions they performed, and the facilities that were provided for controlling their operations. However, the official release, and statements made by the designers and users, made it seem quite fair to classify the Colossus as a special purpose computer, with at least a limited form of "conditional branching" within programs.

> It was, however, externally programmed; and there is no question of it being an actual stored program computer. This final step in the invention of the modern computer had to await the development of a practical high-speed memory capable of holding a large number of binary digits. The only variable stores on the Colossus were gas tubes and hard valve trigger circuits.[25]

Randall pointed out as well that the only other computer of the period comparable to Colossus was the ENIAC, since they were the only two electronic programmable computers. It was reasonable only to compare electronic computers since they provide roughly one thousand fold increase in speed over their electromechanical predecessors. ENIAC, although begun in 1943, was not completed until 1946. However, it was a much more powerful machine than Colossus, contained 18,000 tubes, and had other features similar to Colossus. Neither machine was general purpose, although ENIAC's specialization to numerical calculations (for ballistical trajectories) was certainly more generalized than Colossus' specialization to compute certain types of Boolean functions. Good pointed out,[26] however, that Colossus had more general purpose capabilities than the special ways in which it was used because ordinary calculations could be expressed in Boolean terms. As evidence of this, he mentioned that Colossus could carry out base 10 multiplication. ENIAC was a more important

[25] Randall, p. 348.

[26] Good, p. 46.

development in the history of the computer since it was a direct predecessor of the modern electronic digital computer, leading to a genealogy including EDVAC, the Institute for Advanced Study computer, and a large number of commercial computers. However, the importance of Colossus as a training project for the computer scientists of Britain can not be overstated. From the Colossus project, Newman and several of his workers went to Manchester to work on MADAM, Turing went to NPL to work on ACE, and Coombs and Chandler remained at Dollis Hill to design the MOSAIC computer. This accounted for much of the early post-war computer activity in Britain.

It is much harder to assess Turing's role at Bletchley Park than to describe generally what work was accomplished there. Turing was recruited originally by Knox and groomed as his assistant for a leadership position. At first, Turing was delegated head of Hut 8, which worked on decoding German naval messages through the use of the bombe. The hope was to cut down on heavy British ship losses in the North Atlantic due to U-boat attacks. His mathematical knowledge of computing theory and his knack for constructing powerful machinery from available paraphrenalia also made him an invaluable member of the teams actually constructing the bombes, Robinson machines, and Colossus. Newman, the head of these projects at Bletchley Park, was a colleague of Turing's at Cambridge and was said to have been inspired by Turing's work on computable numbers. Newman provided Turing with the opportunity to work on the actual construction of the Robinson series and included Turing among those in determining the features to be included in the Colossus. Turing was only one of many figures involved and, at least in the case of Colossus, his importance in the actual construction of the project seems to have been overshadowed by the work of others.

Turing had other responsibilities beside working on the construction of the cryptanalytic machines. He was purportedly sent to classified conferences in the United States in late 1942 to explain the workings of the Bletchley Park machines and to gain information on the progress of the similar American operations, which were being carried out under the code-name "Magic." I. J. Good, a co-worker of Turing at Bletchley, believes that Turing may have discussed the atomic bomb on his American trip:

> I believe that on his trip to the United States, Turing may have discussed the atom bomb, because soon after he returned he mentioned a problem concerning branching theory and made it clear that he could not refer to the application. The problem was related to a number of barrels of gunpowder placed at the points of a two-dimensional lattice. The question was, if one of the barrels exploded what was the probability that the explosion would die out on an infinite two-dimensional lattice.[27]

When Turing returned from America, he was relieved of the administration of Hut 8. At about the same time, in 1943,

[27] Good, p. 42.

Dilwyn Knox died, and Turing assumed some of his cryptanalytic and administrative responsibilities. However, after being relieved of the direct responsibility for Hut 8, Turing was generally more free to spend time on theoretical research involving the coding problems and the mechanical decoding equipment.

It is hard to determine exactly what Turing's contributions were to the military computing machinery at Bletchley Park.[28] As stated above, much of the material remains classified. Besides, these were group projects, and it is somewhat difficult to untangle individual contributions. One might get the impression that Turing was mainly a consultant at Bletchley Park. For example, the Bombes originated with the Polish cryptanalysts, and the Robinson and Colossus machines originated from the group working with M. H. A. Newman. Such a view is probably misleading. There were significant improvements in the British version of the bombe, and there is reason to believe the similarities to the Polish <u>bomba</u> were not as great as the similarity of names suggests. Besides, Turing's high level position at Bletchley suggests that he was among the most valued of consultants. Newman admits that his machines were inspired by the machines in Turing's computable

[28] Certainly more would be known if the author had had access to the following papers: Brian Randell, <u>The Colossus</u>. Report No. 90, Computing Laboratory, University of Newcastle upon Tyne (June 1976), and I. J. Good's article on computing machinery at Bletchley Park forthcoming in the first issue of the <u>Annals of the History of Computing</u>.

number paper, although he does not explicitly state what he took from Turing's paper. Perhaps it was the resemblance between the bombes and Colossus and the universal Turing machine in that one of their tasks was to mimic other machines. This meant that the logical design of the universal Turing machine could be used as the starting point for the logical design of the Bletchley machines--although no evidence is available that directly verifies that it was used in this way.

If it is unclear what effect Turing had on the development of the Bletchley Park machines, it is easier to conjecture what benefits accrued to Turing from his wartime experiences:

(1) Experience in practical electronics.

For the typical Cambridge graduate not majoring in physics, his education in physics consisted of school physics and some theoretical work in mathematical physics, but did not include any experimental or practical work of the type important in designing electronic machines. This was experience Turing was to gain while at Bletchley Park and to use so productively in his design of ACE.

(2) Experience working on actual engineering design problems rather than just on theoretical machines.

Turing developed a great deal of theory about the logical design and programming of computing machinery in his 1936 paper, but his concern with practical considerations such as reliability, durability, and especially speed of computation were the result of his wartime experiences.

(3) <u>Further time to examine the possibilities of a physical computing machine</u>.

Turing had intended to pursue the construction of a physical computing machine when the war intervened. Later in the war Turing was relieved of some of the pressure of day-to-day code breaking and allowed to work more on research-oriented problems. The working of the powerful computing machines at hand gave Turing a chance to compare the workings of physical machines with his theoretical machines.

(4) <u>Contact with engineers and mathematicians interested in problems of computers</u>.

Up until this time there is no evidence that Turing had discussed computing and computing machines other than with a few logicians, mainly at Princeton (most notably von Neumann), and perhaps with a few people at Cambridge when he started to build his "zeta" machine. Bletchley Park provided Turing with the acquaintances among whom were many of his future colleagues.

(5) <u>An entré to a position with capital where he could actually design and construct his own machine</u>.

Ironically, although the war preempted Turing's attempt to construct a computer at the completion of his Princeton degree, the war also provided Turing with the contacts and experience to secure the position at the National Physical Laboratory after the war. It is unclear whether Turing would have found someone to provide the substantial financial backing necessary for the construction of a computer had the war not intervened. For then Turing would have sought the support directly after the Princeton Ph.D. without any practical experience and without the knowledge from the war that such machines could be built.

Computing at the National Physical Laboratory

At the end of the war Turing was offered a lectureship in mathematics at Cambridge. Surprisingly, for a man who so loved Cambridge, he refused it. His activities at Bletchley Park had provided him experience with electronics and special purpose computing devices and had whetted his appetite for building a universal electronic computer. As the war concluded, Turing sought positions where he could be responsible for the actual design and construction of a computer. With private enterprise not yet involved in the computer industry, there were only four places to seek computer employment in Britain: the Cambridge Computing Laboratory, the National Physical Laboratory, the University of Manchester, and Birkbeck College, London.

Presumably, Turing's first choice was the newly founded computing laboratory at Cambridge, for he was most comfortable in academic environments and had many friends remaining in or returning to Cambridge. Cambridge had already had experience with mechanical devices, in particular with a differential analyzer, so they had decided to use the existing war technology in electronics, mainly from developments in radar, to develop an electronic calculating device. In the final decision on a director for the laboratory, Maurice Wilkes, another Cambridge man who had a distinguished war career in radar, was chosen over Turing. Although part of the reason for this choice is undoubtedly due to Wilkes' experience with radar and delay lines, it has been suggested[29] that Turing was not chosen because of psychological problems that had surfaced during the war and because of his stubborn insistence to design machines in a way inconsistent with the history and aims of the Cambridge laboratory. In any event, Wilkes was chosen to direct the activities that led to the construction of the EDSAC computer.

M. H. A. Newman was hired immediately after the war to develop a computer (which Turing was later to program) at the University of Manchester. Presumably, Newman intended to continue the design of machines based on his work at Bletchley Park on the Robinson and Colossus series. Similarly, Professor A. Booth

[29] See R. Malik, And Tomorrow the World.

was given control of the project at Birkbeck College of the University of London. Since Turing desired a position where he would have control over both the design and the construction of a machine, neither Manchester nor London was an appropriate position for him at the time.

The Mathematical Division of the National Physical Laboratory was established in 1945 to coordinate the scientific computing activities that had developed during various military projects throughout the war. The charter called for coordination, research, and development in numerical mathematics and computing equipment. J. R. Womersley was made Superintendent of the Division. Divisions were formed for desk computing, statistics, punch cards, differential analysers, and electronic computing. Upon recommendation of M. H. A. Newman, a logician and colleague of Turing at Cambridge before the war and the man responsible for the Heath Robinson series of machines at Bletchley Park, Turing was given the position of director of the electronic computer section at NPL.

As other members of the Mathematics Section at NPL were involved in their own projects, Turing was left entirely on his own to design an electronic computing machine. At the advice of Womersley, the machine Turing designed was named Automatic Computing Engine (ACE) in recognition of Babbage's seminal work on an Analytic Engine. While Turing was in the process of writing a preliminary proposal for ACE, he received an advanced copy of von Neumann's famous report on the logical design for EDVAC. Upon receiving von Neumann's report,

> . . . being Turing, his immediate reaction was to rethink the proposed design and draft his own version. It was eventually to be a most ambitious version, for Turing's principle was that anything anyone in this field could do, he could do better. Thus, for instance, where the EDVAC design had fifty storage delay lines, the Turing design proposed five hundred.[30]

After a few months, Turing submitted a detailed proposal[31] to the Executive Committee of NPL for the construction of ACE which was duly accepted.

Turing continued to develop his ideas in isolation and was just completing Version V of ACE when he was given his first assistant, for in May 1946 J. H. Wilkinson was assigned to work half time for Turing and half time in the desk computing section. In late 1946 and 1947 a number of people[32] were added to the ACE project. Their task was to assist Turing in developing "the logical design of the ACE in the light of experience gained in trying to program the basic procedures in mathematical computation."[33] With the help of these people, Version V was abandoned,

30 Ibid., p. 26.

31 "Proposal for the Development of an Electronic Computer," National Physical Laboratory Report Com Sci 57, 1972.

32 M. Woodger, D. W. Davies, B. Curtis, and J. H. Norton.

33 J. H. Wilkinson, "The Pilot Ace at the National Physical Laboratory," p. 337.

Versions VI and VII, which included new addressing systems, were designed, and a great deal of detailed coding was done between 1946 and 1948.

The original decision, made in 1945, was to subcontract the actual construction of the hardware to some other government department. For about a year Turing kept in touch with former wartime colleagues at the Post Office Research Station at Dollis Hill who had been involved with radar and the "bombes" during the war, intent on having them apply their electronic experience to the actual construction of ACE. However, there was concern among the ACE staff that they would get either disinterested performance or an attempt to take over their project if the actual construction were farmed out to another agency.[34] So, finally, in 1947, the policy of farming out the construction was abandoned and plans were made for construction within NPL. A new Electronics Section was established under the auspices of the Radio Division of NPL, and H. A. Thomas was hired to direct the section. Most of the personnel assigned to the Electronics Section were transferred in from other sections of NPL. Although most of these transferees had some experience with electronics, almost all had to gain specific experience of pulse techniques on the job itself.

[34] This was certainly Wilkinson's view. See <u>Ibid</u>., p. 337.

Unfortunately, Turing had trouble cooperating with his staff. The head of the Section, Thomas, was "an energetic man, but unfortunately his chief interest was in industrial electronics rather than in the construction of an electronic computer."[35] From the date of his hiring, it was apparent that Thomas had nothing in common with Turing and that fruitful collaboration between them was not to be expected. It is hard to classify a man as idiosyncratic as Turing. However, the conflict amounted for the most part to the differences in methodology and outlook between an academic scientist and an industrial engineer. The situation was further exacerbated by the cool relations between Turing and H. D. Huskey. Earlier in 1947, H. D. Huskey had joined the ACE Section for a sabbatical year.[36] Huskey, who had worked on ENIAC and was thoroughly knowledgable of electronics, was a valuable member of the group. It was apparent that Turing was not happy with the intrusion of an outsider interfering with the construction of his machine. Huskey did nothing to ingratiate himself with Turing, by making it known from the beginning that he believed Turing's plan for outside construction of the hardware mistaken and by actively campaigning for construction within NPL itself.

Huskey, who worked well with the other members of the ACE

[35] Ibid., p. 337.

team, convinced the members of the Mathematics Section to build a pilot machine based, for simplicity, on Version V. This machine was known as "The Test Assembly." Turing, disillusioned with developments at NPL, asked for and was granted a sabbatical for the 1947-48 academic year to return to King's College, Cambridge, where he remained a fellow. Meanwhile, Thomas, apparently jealous of the work on Test Assembly, had Womersley rule that all actual construction should be left to the Electronics Section.

With the Test Assembly project denied, with Huskey disillusioned at the progress as his year at NPL ran out, and with Turing off at Cambridge unenthusiastic about the developments on his pet project, the Mathematics Section reached emotional nadir late in 1947. This was followed in 1948 by a series of departures from NPL and a subsequent regrouping which perhaps was what saved the ACE project. At the end of 1947, Huskey's sabbatical ended, necessitating his departure from NPL. Thomas decided to take a position in industrial electronics and was replaced by the more cooperative F. N. Colebrook as head of the Electronics Section. Turing returned in May, 1948, from Cambridge but, remaining disillusioned over the state of his project, quit to join the group building a powerful electronic computer at Manchester under the general direction of his Cambridge and wartime friend. M. H. A. Newman.

Despite these departures, work continued on an electronic computer at NPL. Colebrook, the new Electronics Section head, persuaded Womersley to allow the members of the Mathematics Section who had worked on the Test Assembly to work jointly with the Electronics Section to build a machine. Early in 1949, detailed design for the Pilot ACE--as the machine was to be called--began, using the basic design in Turing's most finished version (V) for the ACE. Assembly of the machine began in the Fall of 1949, and the machine was successfully operated for the first time in May, 1950.

ACE is perhaps the most important physical machine in Turing's career, for it was his sole attempt to develop logically, from first principles, a physical computing machine. When discussing the ACE project, care must be taken to distinguish between the various plans for the machine (five plans by Turing and two more in conjunction with others), the Pilot ACE, DEUCE, the full-scale ACE, and the various minor additions and changes to these versions made along the way. Although all of these plans share a number of features, Pilot ACE varies from the others in that it was built with minimal features so as to minimize cost and time and was intended originally only as a model for the full-scale machine. DEUCE variea from the other machines in that it was designed for specific commercial uses, and its features and capabilities reflect this. Although it was closely modelled on Pilot ACE, its hardware was refined over a long period of time before being constructed--a delay which Turing always feared.

Turing designed ACE to be a very large computer with ultrasonic, mercury delay line storage. The proposed delay lines were the kind developed for radar during the war, in which both numbers and instructions could be recorded as a series of sound pulses in a trough of mercury. When the wave reached the end of the trough, it was electronically recycled to the beginning. There were to be 200 delay lines holding 6400 thirty-two digit words. Many features of Turing's proposal were similar to, and possibly taken from, the early design of EDVAC. Both machines were to be serial and binary. Memory location was similar in the two machines, with information being located in both by specifying both the address of the delay line and the position of the word in the line. Probably because of his experience at Bletchley Park, Turing used Hollerith punched cards for input and output. The processor was organized around temporary storage registers, some used for arithmetical operations and others used only for temporary storage. Arithmetic was effected by including adding circuits in the recirculation paths of certain of the delay lines. Control was effected by a control circuit which would pass

[36] See Wilkinson for details.

on encoded instructions to flip-flop tubes[37] which would then temporarily retain the information in static form until all the requisite switching operations could be accomplished. Turing's plans included a highly original code. In order to increase speed of computation, the plans replaced a central accumulator by instructions which called information from a specific source location and guided it after processing to a specific destination--in effect a sophisticated addressing system. Each version of ACE included plans for a detailed set of logical operations. Perhaps most original in the logical design were plans for a rotator, a circuit underlying the idea of the modern parallel shifting network.

The NPL staff decided to build a Pilot ACE as a model of the full-scale ACE. By modelling the function of the machine with pencil and paper (similar to the way in which the Turing machines were designed to operate), the staff decided on the features for such a skeletal machine. The main store was to consist of mercury delay lines which held 300 words of thirty-two binary digits each. The design included punched cards for input and output, an automatic multiplier capable of operating at a rate of two milliseconds,

[37] Flip-flop tubes were tubes stable in either of two positions and which could be changed form one position to the other by an electrical impulse.

machine operation with digit repetition rate of one megacycle, and arithmetical operations carried out at a rate of sixteen per millisecond. As an afterthought, Wilkinson developed a multiplier which turned this test model into a valuable computing device which, in actuality, was so used for a number of years. In 1954, after Turing had gone to Manchester, the store was improved by the introduction of a magnetic drum[38] of thirty-two tracks (and soon replaced by one with 128 tracks). Pilot ACE served faithfully until 1956, when it was dismantled and donated to the Science Museum in South Kensington.

The Pilot ACE was originally intended as an experimental machine built only to test the design of the ACE computer. A full-scale ACE was planned from the beginning. However, the utility of the Pilot ACE and the DEUCE delayed the construction of the full-scale machine. The British government felt the need to keep its only electronic computer in operation, so once initial testing procedures were completed, and automatic multiplier and an improved control unit were added, the Pilot ACE was drafted for regular government service. In fact, for several years, Pilot ACE was the most important computer in

[38] It is unclear whether Turing had a role in the transfer of technological improvement from Manchester to his pet project at NPL. The drums used at NPL were tested and constructed at NPL itself however.

Britain since it was faster than its rivals and was used as a finished machine rather than as a test model for a number of important computing projects.

Meanwhile, the English Electric Company became interested in electronic computers, and a small contingent was sent to NPL to study Pilot ACE. In 1951, after three years of association with the ACE project, English Electric Company decided to build an engineered version of the Pilot ACE, called DEUCE. The design was completed in 1953. It included a larger magnetic drum (256 tracks), an additional automatic divider, and additional short delay lines. The earliest DEUCE machines were completed in 1955, and NPL received one that year. Many were sold to large industrial companies, particularly aircraft manufacturers. Although DEUCE was used primarily for industrial work, pure scientific research was also carried out on these machines.[39]

In designing the Pilot ACE, the objective was to construct a machine capable of solving eight simultaneous linear equations. The first program was operated on Pilot ACE in May, 1950. By the end of the year the group was confident enough to present a demonstration to the press. This demonstration included using two

[39] M. Woodger, "The History and Present Use of Digital Computers at the National Physical Laboratory," p. 443. mentions as an example Dr. J. S. Rollett of Oxford University and his work on X-ray crystalography.

long delay lines to trace the passage of rays through a compound optical lens. Soon the mathematical capabilities were being tested. Among the earliest of these tests were programs for the quadrature of $1/(1+x^2)$ by Simpson's rule, integration of Bessel's equation, prime factorization of large integers. Subroutine programs were written for addition, multiplication, and square roots because, according to Turing's plans, only logical and not arithmetical operations were wired into the machine. During 1951 a program for solving simultaneous linear equations was devised. By June of that year the machine had shown its utility by solving a system of seventeen linear equations in a few hours--a feat that rivalled any alternative computing method. By the end of the year Pilot ACE had demonstrated the expected superiority of digital over analog computing devices in terms of accuracy when a program was written which could caluclate e to 306 decimal places.

The real strength of Pilot ACE was its facility in solving problems in numerical linear algebra.[40] The machine could easily find solutions to simultaneous systems of linear equations, invert and multiply matrices, and find latent roots and vectors

[40] Some of the results can be found in J. H. Wilkinson, Progress Report on the Automatic Computing Engine," Divisional Report, NPL, MA/ 17/ 1024, April 1948.

of matrices. These techniques were used on bomb trajectories, aircraft flutter calculations, reduction of aircraft survey data, calculating pressure distributions on aircraft wings, determining scattering of atomic particles, and in calculations in hydrodynamics and geomagnetism. Pilot ACE was also used to do a certain amount of pure scientific research. For example, it was used to find and approximate zeroes of Lamé polynomials.

The Pilot ACE was so heavily used for numerical linear algebra problems for two reasons. First, the programming was being carried out by members of NPL's Mathematical Division, which specialized in numerical ayalysis. Thus, they used their standard tools, various techniques of finite differences and matrix manipulations, and transferred them to the computer in a natural way when it was available. Second, and more important of the two reasons, Pilot ACE was well adapted to do such calculations. IN fact, Pilot ACE was the best adapted computer at the time for numerical linear algebraic problems. The punched card input/output system and the rapid card reader seemed to be the decisive features. For example, to multiply a matrix by a vector, one would first place the vector in the stores of the machine and the rows of the matrix on punched cards. The cards could be read at full speed through the machine and, between cards, the machine would have time to do the arithmetic necessary to calculate the entry in the resultant vector. Thus, such a computation could be carried out as rapidly as the card reader could read--600 cards per minute. This was faster than either SEAC or EDSAC could do the computations--Pilot ACE's only rivals in such enterprises.

In fact Pilot ACE's facility in solving numerical linear algebra problems determined to some extent the demand for this facility. The programmers of Pilot ACE tried to reduce every problem to a matrix calculation. The effect was powerful techniques within a limited range of programming. This was appealing to the users of the machines, especially the aircraft industry which was among the first to utilize the facility, for it meant less new programming to go wrong and a quicker return of solutions to their problems.

As early as 1946 there were detailed plans for a full-scale ACE. Although plans were to be modified in light of experience with Pilot ACE, certain features had already been determined for the full-scale model: it would have a sophisticated addressing system, a large number of delay lines as its rapid access store, and punch card equipment for input and output. Nothing more was done towards designing a full-scale ACE until 1953 when testing was carried out on a design for delay line circuits. Over the next several years this and other design work slowly built up. By 1956 decisions had been made on the instruction coding and addressing system and the electronics that went into its design. Decisions were also made on the input nad output equipment, which was to include a fast card reader capable of reading 600 cards per minute, two broadside card readers, and a card punch. Word length was to be forty-eight digits with a digit repetition rate of one and one-half megacycles. The machine was quite a bit more powerful than its already existing commercial brother, DEUCE. ACE had a more elaborate instruction code, allowing it to provide a wider range of facilities than DEUCE. Instructions were carried out at thirty-two per second--twice the rate of DEUCE. Addressing was improved. ACE was able to transfer data simultaneously from all four drums to the delay lines in a few milliseconds. All of this contributed to the better than four-fold increase in speed of ACE over DEUCE. Despite this fact, many computer scientists feel that the full-scale ACE was a mistake, having already been outstripped by other computing machinery.

It is easier to assess the importance to the development of computer science of Turing's work at NPL than at Bletchley. Turing's greatest contribution to computer science is his development of the art of programming. It is not unfair to say that with ACE, Turing established many principles that are used even today in programming. The stress in Turing's ACE report[41]

[41] "Proposals for Development in the Mathematics Division of an Automatic Computing Engine (ACE)," Turing (1945), report before the Executive Committee of NPL, 19 March 1946.

was on the software development. The report discussed at length how programs were to be written and processed. Turing viewed the machine as being designed for obeying programs (although these programs do involve numerical computation), rather than exclusively as machines to perform numerical calculations.[42] Thus Turing proposed to use ACE to play chess, for example, as well as to compute solutions to a system of equations. This interest in programming had begun with the 1936 paper, and the continuity of thought is apparent in the ACE report, where he tried to show how the programming developed for Turing machines could be adapted to actual physical machines.

Turing's view that a computer is a machine for carrying out programs and not just for numerical computations led him to design a computer which, for the first time, required serious consideration of software and which consequently led to the development of a number of novel features which new are in standard use.

(1) Turing's ACE report provided the first <u>complete</u> design of a <u>stored program</u> computer architecture, for Turing allowed

[42] Von Neumann, like most others, holds the narrower view of the purpose of the computer. Because of this, in the EDVAC report Von Neumann concentrates almost exclusively upon the way in which arithmetical operations are to be carried out. This point and most of the others in this section on programming have been made in either Carpenter and Doran, "The Other Turing Machine," or in Wilkinson, "The Pilot ACE at the National Physical Laboratory."

instructions to be encoded as numbers and thereby to be stored in the memory just like other data.[43] Both von Neumann and Eckert had proposed that programs be stored, but Turing was the first to work out all the details.

(2) Turing held an essentially modern view on <u>access to variables</u> and treated memory in a modern fashion as a random access addressable device.[44] For example, to Turing, branch instructions merely specified the number of the next instruction.[45]

(3) Turing included a simple, straightforward type of conditional branching in his proposal for ACE.[46] For example,

[43] The idea of encoding instructions as numbers is an old idea. However, without question, Turing was led to this idea by Gödel's incompleteness theorem in which the sentences of a formal language are encoded in the integers in order that they can be processed. In the 1936 paper Turing calls his programs "tables" (because he listed the steps of the programs in tabular form). Incidentally, von Neumann was the first after Turing's 1936 paper to describe the stored program concept in any detail.

[44] Von Neumann only arrives at this idea later, some time in 1946. In the first draft of the EDVAC report, von Neumann speaks of "transient transfers" where "the place of the minor cycle which contained the transfer order must be remenbered" (quoted in Carpenter and Doran, <u>op. cit.</u>, p. 270).

[45] Von Neumann considers branch instructions as orders from control to transfer from one connection in the memory to another.

[46] Contrast Turing with von Neumann, who employed an instruction to select, on a certain condition, one of two numbers which was then stored into the address field of an unconditional branch.

when Turing wanted to include the branching instruction

Do instruction 10 if a = 0
" " 20 if a = 1 , he would consider instructions

10 and 20 as encoded digits and write the conditional branching as the calculation (A x Code for Instr. 10) + (1 - A) x Code for Instruction 20.[47]

(4) Turing also provided ACE with the capability of program modification. He allowed for the modification of addresses of instructions as the machine executed.[48] He also allowed for the manipulation of instructions as if they were numbers.[49] Finally,

[47] This is the technique of characteristic functions that was used by the recursion theorists to apply recursive functions to sets and propositions. Without doubt, this is where Turing found the idea.

[48] In the draft report von Neumann distinguished instructions and data by a one-bit tag. Only addressed could be modified. This was the only means of conditional branching or array indexing discussed in the draft report.

[49] Turing uses this technique as the basis of his conditional branching. Von Neumann discusses no such technique in his draft report.

he provided for the possibility of one program processing another --treating it as data.[50]

(5) Turing developed a control processer with a novel address register and optimal coding procedure. To increase speed of computation, Turing introduced "optimal coding," where consecutive instructions were stored in relative positions which would allow one instruction to emerge from the delay line just as the previous instruction was completed.[51] Also to increase speed of operation, the usual central processer was replaced by a series of temporary storage registers and a novel address register whereby each instruction represented a transfer of information from a source to a destination. This addressing system,[52] which was due to

[50] This follows from his idea of a universal machine in the 1936 paper, where he explicitly mentions how the standard description number of a machine is put in the universal machine. Although the ability of one program to process another is thought of as a fundamental characteristic of von Neumann's machine, it was suggested independently—and possibly originally--by Turing. Since von Neumann gave each word a non-overrideable tag, he could not manipulate instructions so as to allow one program to process another.

[51] This provided a significant improvement over consecutive storage of instructions. According to Wilkinson, op. cit., p. 336, Turing was obsessed with the speed of operation of his machines. Optimal coding turned out to be very untidy, but also quite powerful. One of its greatest triumphs was the early development of double and triple precision floating point programming and the subsequent development of floating point error analysis. The use of optimal coding was quite controversial at the time.

[52] More about this addressing system can be found in Wilkinson, op. cit., p. 337.

Wilkinson and the rest of Turing's group, replaced the addressing schema

A → Processor, B → Processor, Arithmetical operation, Processor → C

by the addressing schema

A Arithmetical Operation B → C, next instruction D.

(6) Turing fully developed tha art of subroutine programming, which had been used in a primitive way in earlier relay machines. He conceived of instructions as components of basic operations and consequently wrote programs making extensive use of subroutines. A number of subroutines were explicitly written.[53] Turing looked forward to a computer holding an extensive library of subroutines stored as machine instructions with symbolic addresses. Turing even tackled the difficulties inherent in calling up subroutines at various stages in a program, resulting in the design of a

[53] Included in the report were the following subroutines:

Label	Task
INDEXIN	Add a work with a certain address to the register.
PLUSIND	add one to the register.
MULTIP	Perform multiplication.
ADD	Perform addition.
TRANS45	Transfer
BURY	Save the return address of the next subroutine branch.
UNBURY	retrieve the last saved address and branch to it.
DISCRIM	Depending on the output of the logic unit, transfer one of two storages to a particular address.
BINDEC	Transfer number in accumulator to output buffer which converts it to binary words, which in turn punches a card which gives a decimal representation of the number in a Hollerith code.

software stack for subroutine linkage (the modern concepts of PUSH and POP).[54] He also showed how to externally link-edit subroutines rather than use the slower approach of loading each subroutine at a fixed address.

(7) Turing was among the first computer scientists to call for precise program documentation.

> In addition to these we must recognise the 'general description' of a table. This will contain a full description of the process carried out by the machine acting under orders from this table. It will tell us where the quantities or expressions to be operated on are to be stored before the operation begins, where the results are to be found when it is over and what is the relation between them. It will also tell us other important information of a rather dryer kind, such as the storages that must be left vacant before the operation begins, those that will get cleared or otherwise altered in the process, what checks will be made, and how various possible different outcomes of the process are to be distinguished. It is intended that when we are trying to understand a table all the information that is needed about the subsidiaries to it should be obtainable from their general descriptions.[55]

That Turing was concerned with precise program documentation does not seem surprising, for the idea is apparently an extension of his notion of standard descriptions for Turing machines put forth in his 1936 paper. The purpose of these standard descriptions was

[54] Turing used the terms "BURY" and "UNBURY" for "PUSH" and "POP", respectively.

[55] From Turing's proposal for ACE, p. 29, as quoted in Carpenter and Doran, op. cit., p. 272.

to provide a precise formal characterization of informally defined functions. Thus it is not surprising that the logician Turing was concerned about the formalities of the actual programs for ACE as well as for the idealized programs for Turing machines. Other computer scientists, not in logic, were not so fussy about the formalities of writing down that the programs worked--as long as they did indeed work.

(8) Turing was among the earliest to use instructions which did only the most basic of tasks. The advantage, it turned out, of the simplicity of this scheme was that these instructions could be combined in a much larger variety of ways to do a larger variety of tasks than could machines, like EDVAC, which had only a few instructions, each of which did a number of tasks at one time.[56] It is not surprising that Turing designed his instructions each to do a single logical task, for his aim was to design a machine which could carry out a variety of different kinds of programs, presumably which might be carried out in rather different

[56] According to Carpenter and Doran, op. cit., p. 274, von Neumann allowed four types of instructions: load data, unconditional branch, load contents of some addressed location. do arithmetic operation and send the result to a particular location.

combinations of steps.[57] Listed below are the types of instructions used by Turing:[58]

Type	Task
B	Branch Instruction
K	Store in main memory
L	Load form main memory
M	Load from temporary storage
N	Store in temporary storage
O	Punch a card
P	Read a card
Q	Load with data
R	Boolean Operations
S	Arithmetic Operations
T	Turn on valves

Pilot ACE was not the first fully automatic, electronic machine to be built in Britain. Minimal facilities were available at Manchester in 1948, and, by 1949, several machines were completed. The first fully automatic computations were carried out at Cambridge on EDSAC in 1949. Pilot ACE was only completed in 1950. Nor did Turing's work lead to the first commercial, fully automatic, electronic machines, for Fereanti Mark I

[57] It was reasonable for von Neumann's machine to have only a few instructions, for his machine was designed to carry out only one specific type of task, numerical computation. For example, von Nuemann did not need an instruction to carry out Boolean operations because Boolean operations occur in numerical computations only as parts of arithmetic operations. Thus, von Neumann only needed to have instructions to carry out the arithmetic operations. Nor did von Neumann need so many specific instructions about transfer of information within the machine or an instruction to turn on the valves--as did Turing.

[58] Adapted from a more detailed list in Carpenter and Doran, op. cit., p. 275.

(based on the Manchester machines) and LEO (based on EDSAC) appeared in 1951, four years before DEUCE.

Pilot ACE was not ahead of its rivals in the equipment used in construction. All of the first generation of electronic machines owed a great deal technologically to war developments of radar and electronic computing equipment. ENIAC, as well as Pilot ACE, used punched Hollerith cards for input and output. UNIVAC, EDVAC, EDSAC, and SEAC, as well as Pilot ACE, used ultrasonic mercury delay for storage. EDSAC, as well as Pilot ACE, had additional magnetic storage.

What, then, was the importance of the Pilot ACE? First of all, it was the fastest of all the first generation electronic computers. Because of this, it was used for several years for important computing projects, especially by the British government. Second, and the main reason for its impressive speed, was its design as a non-sequential machine with novel programming and addressing systems. Non-sequential machines are harder to design and harder to program; but, they result in substantial increase in operating speed. Turing was the first to arrive at a workable design for a non-sequential machine. It was these contributions to programming and addressing systems that made Pilot ACE important to the development of actual computing technology.

Computing at Manchester

In 1948 Turing decided to leave NPL to work with his long-time friend, M. H. A. Newman at the University of Manchester. Apparently, his hope was to get away form the pedestrian details of computer construction and go somewhere where he could work on the uses of the computer. He was appointed Reader in the Department of Mathematics, under the direction of Newman. Turing's responsibilities were to include programming of the Manchester machines and the devleopment of uses for the computing facilities in solving mathematical problems. This suited Turing, for it gave him great flexibility in the tasks which he assumed and provided him with the time to investigate the computing powers of the computer. Turing found this arrangement agreeable because it was at this time that he became interested in actively investigating the question of whether a computer can think by seeing just what sorts of mental activities could be programmed into the computer.[59]

Manchester provided a welcome environment for Turing. The university had just completed a working prototype of their electronic computer--several years ahead of NPL. Their needs meshed

[59] See Chapter Six for a discussion of Turing's work on computers and intelligence.

with Turing's greatest strength, programming. Turing's affiliation was with the mathematics department and the mathematical side of computing. Thus he did not have to deal with the administrative or engineering problems which he had found so distasteful at NPL. His duties were light and only loosely defined; so he had freedom to investigate problems of his own choosing. Finally, he had the company of a friend and fellow mathematician-logician, Newman, who shared the same outlook on computer science.

Although all of the Manchester machines were called Mark I, there were actually four distinct early machines:[60] a rudimentary machine completed (in June, 1948) roughly three months before Turing came to Manchester; two prototype machines, the improved machine (completed in April, 1949) and the large-scale machine (completed in October, 1949); and finally, a production Mark I (completed in February, 1951) which was marketed by Ferranti Limited, a Manchester manufacturer of electrical and electronic equipment. Turing worked on the programming schemes for all but the rudimentary machine. Turing made the following contributions to programming at Manchester:

(1) Development of a programming system which utilized a teleprinter for typing in input and for receiving output. This

[60] The best account of early Manchester computing is in M. Campell-Kelly, "Programming the Mark I," Annals of the History of Computing, II (1980), 130-168.

replaced a cumbersome system by which programs were entered in binary code by means of a panel of pushbuttons.

(2) Development, together with Newman, of the specifications for the logical design for the production Mark I. G. C. Tootill provided the actual logical design for the machine, but he explicitly worked to the specifications determined by these mathematical logicians.

(3) Development of a system for program organization. Programs were broken into chapters, pages, columns, paragraphs, and lines. This provided a convenient way of breaking the program into parts to deal with the storage limitations of the machines, immediate access but limited electronic store and substantial direct-access magnetic store. This system of organization was characteristic of later Manchester machines as well.[61]

(4) Development of the first input routine (called Scheme A) for automatically loading other programs into the machine. Unfortunately, it was written in teleprinter code and provided only the most basic conveniences of the modern assembler.[62]

(5) Development of a subroutine system for the Mark I machines. Turing had previously developed subroutine linkage for ACE, and the Cambridge computing group had made extensive progress

[61] For details, see Campbell-Kelly, pp. 138-139.

[62] See Campbell-Kelly, pp. 141-143, for details.

in the design and use of sub-programming systems.[63] Thus, Turing's work here was not novel. In fact, one author[64] calls it "a rather pedestrian piece of work for a person of his talents."

(6) Contributions to the subroutine library. In October, 1949, Cicely Popplewell was hired as Turing's programming assistant. Her responsibility was to create a subroutine library for the production Mark I. Turing oversaw this work as well as contributing a number of sub-programs himself.

(7) The development of a novel and primitive operating system (called the "formal mode") for making formal and documenting the interaction of programmer, machine, and operator.

> The advantage of working in the formal mode is that the output recorded by the printer gives a complete description of what was done in any computation. A scrutiny of this record, together with certain other documents [,] should tell one all that one wishes to know. In particular this record shows all the arbitrary choices made by the man in control of the machine, so that there is no question of trying to remember what was done at certain critical points.[65]

(8) Authorship of the first programming handbook for the Manchester computers.

[63] See "Report on the Preparation of Programmes for the EDSAC and the Use of the Library of Sub-routines," from the Cambridge Computing Laboratory.

[64] Campbell-Kelly, p. 131.

[65] From Alan Turing, "Programmer's Handbook for the Manchester Computer" (first handbook), p. 55, as quoted in Campbell-Kelly, p. 146. See Campbell-Kelly, pp. 146-147, for details on the formal mode.

Turing's programming work at Manchester did not match his profound programming accomplishments at NPL. The reason for this seems clear. Turing felt he had solved the interesting programming problems while working at NPL. The programming problems that remained appeared to him as mundane. Besides, there were many important uses of the computer which Turing wanted to work on rather than work on the routine details of programming. Thus he focused attention on these problems instead of on programming.

There is much evidence to support this interpretation. First, the overall quality of Turing's programming work at Manchester did not match the quality of his earlier work at Princeton, Bletchley, or Teddington. Second, an additional person, R. A. Brooker, was hired to assume the more routine programming duties when Turing expressed a lack of interest in them. Third, although his interest in programming flagged, his interest in the computer did not. This is evidenced by the fact that, until his death, Turing remained the heaviest individual user of the Manchester computing facilities. He used them, however, for work on a chess program, in mathematical research on Mersenne primes and the Riemann hypothesis, and on his new theory of biochemical morphogenesis. Fourth, Turing demonstrated a profound lack of interest in specific technical improvements which would have greatly assisted the average user, but would have in no way improved the speed or facility of the machine (e.g., leaving input/output in teleprinter notation, lack of interest in an autocode), thus indicating his lack of interest in the routine problems of programming. Turing did not see the point of expending energy making the facilities easier for lesser minds to use. As Wilkes, head of the Cambridge Computing Laboratory described Turing,[66]

> He had a very nimble brain himself and saw no need to make concessions to those less well-endowed. . . . he could not appreciate that a trivial matter of that kind [writing and multiplying binary numbers backwards for the sake of significant digits] could affect anybody's understanding one way or the other.

Thus, he was unwilling to develop programming systems that catered to the mediocre mind. This is also probably why his programming manual for the Mark I "was not a model of clarity,"[67] and additional manuals were written so soon after his was completed.

However, if Turing's work on programming is not of great importance, his other work at Manchester is. His use of the computer to simulate thought (computer chess, solution of mathematical problems, language translation) was of first importance. However, this work is closely related to his theoretical work on artificial intelligence, so discussion of the topic is reserved for Chapter Six (on Turing's contributions to theoretical computer science).

66 "Computers Then and Now," Journal of the Association of Computing Machinery, 15 (1968), p. 4.

67 Campbell-Kelly, p. 65.

Chapter Four: Von Neumann's Contributions to the Development of Physical Computing Machinery

This chapter examines von Neumann's contributions to the development of modern computing machinery. Assessing von Neumann's contributions to this field shares some of the same methodological difficulties as assessing Turing's contributions. A number of the files in the von Neumann archives at the Library of Congress remain closed to public viewing. Thus, it is difficult to assess the war-time relationship between Turing and von Neumann and how it affected computer developments. It is also difficult to obtain a precise picture of the applications--especially military applications--for which von Neumann was making use of the computing facilities.

Von Neumann scholarship shares another methodological problem with Turing scholarship, and with the study of the history of computers in general. Often, developments are group projects, and it is hard, for both insiders and outsiders, to assess the individual contributions to the project. Extensive patent litigation has demonstrated this point. There was such controversy involving von Neumann's contributions to the ENIAC and especially to the

EDVAC computer projects. Much has been written arguing each side of the issue. The aim here is not to settle these disputes. It is unclear whether they can be settled and doubtful whether settling them would have any significant historical merit. For historical purposes, it does not seem essential to choose one person out of a group to assign credit for a certain idea. Often, the group provides the incubation necessary for an idea, even if it is originally the result of a single individual.

What does seem historically useful is to consider how the particular training and insight an individual brings with him might shape the particular developments of a computer project. To that end, this chapter will concentrate on examining the way in which von Neumann's special training and talents, especially his background in logic, affected the specific developments of the three computer projects with which he was involved. The overall details of these three computer projects, ENIAC, EDVAC, and IAS, will be described as succinctly as possible, citing other references of greater depth whenever possible. An attempt is made, however, to incorporate material from the von Neumann archives which has bearing on the development of these projects and which has not been published before.

Applied Mathematics and Computing

Just as in the case of Turing, the war interrupted any plans von Neumann might have had to work on computing equipment immediately after his interaction with Turing at Princeton. Also like Turing, it was the war that presented the opportunity for von Neumann to become involved in computer construction in a much bigger way than he would have been likely to have become involved through his original plans with Turing.

Von Neumann was better trained than Turing for his work on the construction of actual computing equipment. In fact, von Neumann had a unique background in chemical engineering, physics, and several branches of mathematics that was of great importance to his work on computers. In fact, it might even be argued that von Neumann was able to make the important contributions to computer science that he did because of his unique training and background. As an undergraduate, von Neumann had earned a degree in chemical engineering at Eidegenossische Technische Hochschule in Zurich. This background in chemical engineering stood him in good stead for understanding engineering problems and practices as they arose in the computer projects. In fact, his computer work was marked by a decided concern for engineering details.

He also had extensive experience in physics. In the 1920's he did fundamental work in the mathematical foundations of quantum mechanics and, beginning in the late 1930's, he turned ever more to problems in mathematical physics: fluid dynamics, continuum mechanics for nuclear research, meteorology, ballistics, shock problems. This background in mathematical physics provided him with experience in physics, familiarity with problems suitable for work on the computer, and experience with complicated problems in numerical analysis. It also provided him with the general background in physics to feel comfortable in the quickly expanding field of electronics.

His background in mathematics was extremely wide. He was equally comfortable in applied mathematics and in mathematical logic--a feat accomplished by few mathematicians. Not only was von Neumann familiar with both areas; he had done fundamental research in both areas. Beside his work in physics described above, he had provided the mathematical foundation for much of modern applied mathematics through his work in operator theory and in ergodic theory. On the other hand, in the 1920's and 1930's he did substantial work on symbolic logic, set theory, axiomatics, proof theory, Boolean algebra, and lattice theory. The logic background proved useful in his computer work in the logical design of the equipment, in coding techniques, and in programming. The familiarity with applied mathematics gave him a feeling for the problems that had to be solved, the capabilities that a machine must have, the role the computer was to play in mathematical research, whether pure or applied, and the development of a new numerical analysis suited to high-speed computing. The combined knowledge of mathematical logic and applied mathematics benefited him in the development of algorithms for going from the problem to the mathematical solution, and in the development of the programming to go from the pure mathematical solution to the actual computation. This combined knowledge was also useful in developing a theory of error correction for the computers which was partly numerical and partly logical.

There was a great difference between Turing and von Neumann in this regard. Turing's background was primarily in mathematics and mathematical physics. He had to gain his engineering experience on the job at Bletchley during the war. His work in mathematical logic was not nearly as substantial as that of von Neumann, and his work in applied mathematics was practically non-existent. Turing was just beginning his career as the war began. Von Neumann was already established as one of the most valuable members of the scientific community. His permanent position at the Institute for Advanced Study is only one of many indicators of his established position. In fact, von Neumann used his notoriety to great advantage during and just after the war to lend credence to the ENIAC project, to bring it government connections and support, and especially to lend scientific credence to the discipline of computer science. Turing had only one influential paper and the prospects of an outstanding career to offer to the status of computer science.

It was through his interest in applied mathematics that von Neumann was eventually brought back to the field of computing. Beginning in the mid-1930's von Neumann's interest in applied mathematics increased. His work focused on the problems of fluid dynamics, the theory of shock waves, and the theory of turbulence. By the time the United States was drawn into the war effort, von Neumann was an expert on shock and detonation waves. This expertise in applied mathematics led to his involvement with a number of government agencies, including the Ballistics Research Laboratory, the Bureau of Ordnance, and the Los Alamos project. This interest in applications continued until the end of his life. During this time he worked on problems of ordnance, submarine warfare, nuclear weapons, military strategy, weather prediction, and ICBM's--among others.

Many of the war-time problems of applied mathematics, including a significant portion of the work done by von Neumann, were not readily accessible to the analytical solutions of classical mathematics. Von Neumann includes in this category the work of "continuum dynamics, classical electrodynamics through hydrodynamics to the theories of elasticity and plasticity."[1]

[1] H. H. Goldstine and J. von Neumann, "On the Principles of Large Scale Computing Machines," VN, Collected Works, V, 1.

Typical of the difficulties confronted in these problems were those in hydrodynamics. These problems involved solving non-linear partial differential equations for which there were no known analytical methods of solution and for which there were not even insights into the character of the problems. As von Neumann himself assessed the situation:[2]

> Our present analytical methods seem unsuitable for the solution of the important problems arising in connection with non-linear partial differential equations and, in fact, with virtually all types of non-linear problems in pure mathematics. The truth of this statement is particularly striking in the field of fluid dynamics. Only the most elementary problems have been solved analytically in this field. Furthermore, it seems that in almost all cases where limited successes were obtained with analytical methods, these were purely fortuitous, and not due to any intrinsic suitability of the method to the milieu.

It was not for the lack of effort or the lack of first-rate minds working on the solutions of non-linear problems either:[3]

> The advance of analysis is, at this moment, stagnant along the entire front of non-linear problems. That this phenomenon is not of a transient nature but that we are up against an important conceptual difficulty is clear from the fact that, although the main mathematical difficulties in fluid dynamics have been known since the time of Riemann and of Reynolds, and although as brilliant a mathematical physicist as Rayleigh has spent the major part of his life's effort in combating them, yet no decisive program has been made against them--indeed hardly any progress which could be rated as important by the criteria that are applied in other,

[2] Ibid., p. 2.

[3] Ibid., pp. 2-3.

more successful (linear!) parts of mathematical physics.

Von Neumann recognized the impasse with purely analytical techniques and so turned in the late 1930's to an entirely novel computational technique to assist in the solution of these problems. This new computational technique was the result of von Neumann's realizing the possibilities of high-speed computing equipment and also having a profound understanding of how the mathematical physicist used physical insights to aid his mathematical solutions. This required someone open to new technologies, creative enough to see their utility in a classical field of mathematics which requires proofs more than computations, and with a profound understanding of mathematical physics. Von Neumann's creativity, openness, and background were crucial to the development of this new computational technique.

More specifically, the computational technique involves use of the computer as an heuristic tool, used to provide insight into the analytics of the problem through the computation of approximate solutions of a few crucial cases, rather than a complete computational solution to the problem to replace the analytical solution. Von Neumann arrived at this technique by considering the way in which physical insight and experimentation have provided "heuristic pointers" to the solution of purely mathematical problems. He specifically mentioned calculus and the theory of elliptic differential equations as originating from purely physical insights, and their leading to a cohesive body of purely mathematical formulas and techniques. Such was equally true of the breakthroughs that had been made on non-linear problems:[4]

> Such advances as have been made in the theory of non-linêar partial differential equations, are also covered by this principle, just in what seems to us to be the most decisive instances. Thus, although shock waves were discovered mathematically, their precise formulation and place in the theory and their true significance has been appreciated primarily by the modern fluid dynamicists. The phenomenon of turbulence was discovered physically and is still largely unexplored by mathematical techniques.

Many people recognized the fact that a number of mathematical developments resulted form physical problems. The crucial part of von Neumann's technique resulted from his insight into the precise way in which physics aided mathematics: in its use an an heuristic guide for the purposes of providing insight into the problems. Von Neumann described it in this way:[5]

> At the same time, it is noteworthy that the physical experimentation which leads to these and similar discoveries is a quite peculiar form of experimentation; it is very different from what is characteristic in other parts of physics. Indeed, to a great extent, experimentation in fluid dynamics is carried out under conditions where the underlying physical principles are not in doubt, where the quantities to be observed are completely determined by the known equations. The purpose of the experiment is not to verify a proposed theory but to replace a computation from an unquestioned theory by direct measurements.

[4] Ibid., p. 4.

[5] Ibid., p. 4.

Thus, as von Neumann recognized, the purpose of these experiments was to compute approximate answers to very special cases of the problems so as to provide the researcher with better insight into the nature of the problem and what its solutions might look like, consequently giving him some clues in determining analytic solutions to the problems. In fact, as von Neumann observed, the role of the wind tunnel in these fluid dynamics problems is primarily as an analog device for providing measurements of particular solutions. Thus, the wind tunnel is a crude sort of analog computing apparatus, which integrates the appropriate non-linear partial differential equations.

> Thus it was to a considerable extent a somewhat recondite form of computation which provided, and is still providing, the decisive mathematical ideas in the field of fluid dynamics. It is an analogy (i.e. measurement) |analog| method to be sure. It seems clear, however, that digital . . . devices have more flexibility and more accuracy, and could be made much faster under present conditions. We believe, therefore, that it is now time to concentrate on effecting the transition to such devices, and that this will increase the power of the approach in question to an unprecedented extent.[6]

Thus, von Neumann's method for solving non-linear problems led him to computing. In doing so, he devised a general method for the use of the computer as an aid in mathematical research. The computer was to be used to solve numerically special cases of

[6] *Ibid.*, p. 5.

analytically intractable problems. These results were to be used as an heuristic guide to theorizing about analytical solutions. The use for the computer in mathematics was, to von Neumann, not to provide complete numerical solutions to problems, but was to be used for heuristic purposes and for simulation.

Von Neumann's Contributions to ENIAC

During the war, von Neumann's expertise in fluid dynamics was put to use in work on the Los Alamos project. He joined the group as a consultant in 1943 and worked on the problem of implosion. The aim was to find a technique of making safe radioactive isotopes of plutonium reach a critical state quickly, through some type of implosion. This was one of the major problems confronting the group before finalizing the design for the atomic bomb. Von Neumann was able to model the implosion problem mathematically. The difficulty remained, however, of providing a numerical solution to the problem. The large number of computations necessary were being carried out on desk calculators, but these machines were not fast enough to handle all of the calculations.

It was about the time that it became apparent that the Los Alamos problem outstripped the capacity of the desk calculators to complete the computations in a reasonable period of time with reasonable accuracy that von Neumann first happened to hear about a high speed, electronic computer being built at the University of Pennsylvania. This machine, ENIAC, was the first electronic, high-speed, digital computer. It was designed and built at the Moore School of Electrical Engineering at the University of Pennsylvania with the assistance of the Bureau of Ordnance. The plan for the machine had arisen with John Mauchly, a physics professor at Ursinus College, who had suggested to H. H. Goldstine, of the Ordnance Department, that such a machine be built for the purpose of computing ballistic trajectories. Ordnance was agreeable, and construction began in 1943. ENIAC was designed and constructed by many under the general direction of Mauchly and J. P. Eckert, an engineering professor at the University of Pennsylvania.

Plans for ENIAC had already been settled and construction was in progress when von Neumann first saw the machine. It is generally agreed that von Neumann had no significant influence on the original design of ENIAC. The design of ENIAC was unique. It was significantly different from the earlier electro-mechanical machines by being completely electronic. However, it also differed from later electronic machines in its general design. For an electronic machine, it was crude, primitive, and severely restricted by its small memory and tedious hand programming. There were several reasons for this. First, it was the first such machine to be built, and later machines very clearly learned from the experience of ENIAC. Second, it was being built in haste as part of a war effort. Thus, there was not the leisure for a great deal of reflection before the ideas were put into production or for the testing of equipment before it was used. Third, there were serious drawbacks in computer technology. In particular, the only available device for electronic storage of information at the time was the bulky, inefficient, expensive, unreliable vacuum tube. Otherwise, there was only mechanical and electromechanical storage.

Since ENIAC was a general-purpose computer, each particular project to be computed on the machine had to be programmed individually. One of the most severe limitations of ENIAC was that this programming had to be carried out by hand. It involved setting mechanical switches for each piece of equipment in the machine, interconnecting these by cables, and mechanically setting the function tables. Often this process took time orders of magnitude longer than the computation itself. The technique was laborious. It was hard to check for programming errors. Perhaps worst of all, it was extremely inefficient, since the machine was idle during the entire programming process.

After ENIAC was completed and operating, von Neumann did make one important improvement to facilitate the programming of the machine. He recognized that, since the numerical data and program instructions were coded as electrical pulses of a similar size and shape, the two could be processed in the same way by the machine. This possibility may well have come to von Neumann in considering the design of the universal Turing machine, in which the program information and the numerical data were coded in exactly the same way and processed together by one part of the machine. Of course, central programming such as this is harder to carry out when one has to work out the physical as well as the theoretical details of the problem. In fact, what von Neumann suggested was that the program information could be stored by setting switches on the function tables, that part of the machine where switches were set to compute fixed functions of any input. Upon von Neumann's recommendation, automatic equipment was added which connected the function tables, through the master programmer, to the other units of the machines. When one wanted to program the computer for a specific task, all one had to do was to set the function tables for both instructions and numerical data. This automatic equipment would then set the other units of the machine. Although setting the switches of the function tables had to be completed manually, this still resulted in a considerable improvement in time and effort over having to set every component of the machine by hand. This constituted a significant step toward effecting stored program computing.

ENIAC reached working order near the end of 1945. For a long time, von Neumann had recognized the utility of the machine for scientific research. At his urging, the first substantial program run on the machine was by Stanley Frankel and Nicholas Metropolis, from the theoretical Physical Division at Los Alamos. The project required transferring a million IBM cards from the Los Alamos card library to the Moore School! This project was completed early in 1946.

Von Neumann was successful in convincing a number of Los Alamos researchers to numerically test their theoretical ideas on ENIAC. In addition to this work and the computation work carried out for the host organization, the Bureau of Ordnance, there were also independent scientific utilizations of the machine. The Army allowed university scientists to use ENIAC free of charge. This prompted a wide variety of research projects.[7] For instance, in 1946 it was used by Hans Rademacher and Harry Huskey for the study of round-off errors, by Frankel and Metropolis for a calculation on the liquid drop model of fusion, by Douglas Hartree for a study of the boundary layers in a compressible fluid flow, by Abraham Taub and Adele Goldstine for an examination of the properties of shock waves, by J. Goff on the thermodynamical properties of gases, by D. H. Lehmers on the investigation of some problems in number theory, and by von Neumann in modelling for

[7] See H. H. Goldstine, The Computer from Pascal to von Neumann, pp. 157-167.

weather forecasting.

ENIAC was used through most of 1946 at the Moore School. Late that year it was disassembled and moved to the site of the parent organization, the Ordnance Bureau, at the Aberdeen Proving Grounds, where it was put into operation again in 1947. It remained in operation until 1955, when it was disassembled and presented to the Smithsonian Institution.

Von Neumann's Contribution to EDVAC

Due to technological limitations at the time of its conception, plans were being made to supersede ENIAC even before it made its first computation. The main technological limitation was the use of vacuum tubes as the sole quick access means of storage in the ENIAC. Eckert, one of the heads of the ENIAC project, proposed a new machine based on mercury delay lines, which had been used successfully during the war in radar equipment. The plan was first proposed in 1944, before ENIAC was near completion and just before von Neumann joined the Moore School group. Prospects for a new machine were discussed regularly in the latter part of 1944 among a group of the designers, engineers, and mathematicians who had worked on ENIAC. Included in this group was von Neumann. It was decided that plans should be put forth for the construction of a new type of electronic computing machine, to be called EDVAC.

The heart of the new machine was to be the mercury delay line storage. Mercury acoustic delay lines had been used during the war to delay pulses in radar. The delay lines consisted of long tubes filled with mercury. Pulses were entered at one end and slowly moved to the other end through the mercury medium. The pulse would be weakened, but not seriously distorted, when it reached the other end of the tube. For use in EDVAC the output of the delay line would be sent into a pulse amplifier and a pulse reshaper and then entered back into the input end of the delay line. One such delay line and a few tubes for the amplifier and reshaper provided a circulating memory capable of storing about 1000 bits. This was relatively compact compared with ENIAC, where approximately 1000 tubes were required to store 1000 bits. In fact, EDVAC was designed to contain only about 3000 tubes, instead of the 18,000 tubes in ENIAC--although EDVAC was the more powerful machine! It was finally feasible to have a machine capable of carrying out the detailed computations necessary for the applied mathematical problems von Neumann had in mind.

The use of mercury delay lines in EDVAC mandated changes in other features. In ENIAC, many of the units could work independently of one another, so the machine was designed to work in parallel. On EDVAC, however, since the delay lines were serial, the overall design of the machine was made serial. ENIAC was a partly synchrous, partly asynchrous system. With EDVAC, however, with all the memory automatic in the delay lines, the machine was designed to be completely synchrous. Eckert and Mauchly designed a "clock," a central source of pulses to time the entire workings of the machine.

Perhaps the largest difference between ENIAC and EDVAC concerned the programming of the two machines. There was no automatic programming in ENIAC. Apparently the designers had considered it for the machine, but due to the exigencies of the war and to the fact that the sort of programs used for ballistics work would be used for a long time before being changed, they abandoned the idea. As stated above, in ENIAC it was certainly not an automatic process, and not even a particularly quick process to change the plugboard or the function tables. While ENIAC was designed so that the program information could be encoded in the machine (making it a general purpose machine) and while von Neumann assisted the process toward a stored program computer by showing how the program information could be treated in the function tables in the same way as numerical data, there was still one crucial criterion standing in the way of ENIAC beign a stored program computer in the modern sense of the term. Because of the inability to quickly change the function tables, where the program information was stored, it was impossible to have a program which could read and modify itself in the course of a computation. A stored program computer in the modern sense required an internal memory, capable of quick address in which the program information could be stored. The plan was for EDVAC to have the requisite memory through the use of mercury line technology. Thus, the really curcial difference between the two machines was the stored program concept.

As a result of discussions in 1944 and 1945 about the proposed new machine, EDVAC, von Neumann wrote a "First Draft of a Report on the EDVAC." In this report von Neumann worked out, in fine detail, the logical design for "a very high speed automatic digital computer system."[8] The report was distributed by Goldstine to the members of the Moore School working on EDVAC and to some interested outside scientists. It was the first widely distributed document on electronic computers and was the first written report of a machine in which programs could be stored and modified electronically. The report in fact was used in the late 1940's as the basis for computer and stored program design.

The major thrust of the report involved the logical control of the machine, an issue much discussed by the Moore School group during 1944 and 1945. Von Neumann's intention in the

[8] See "First Draft Report on the EDVAC," University of Pennsylvania, 1945.

report, however, was not to state simply the logical design for the EDVAC, but rather to state the logical structure and functioning of any "very high speed automatic digital computer system."[9] Thus he focused on theoretical issues and mentioned technical details about a particular machine only as illustration. Von Neumann believed that there was a science of computing which transcended the technical details of any particular machine. This report was to outline the first principles of such a science. The report consequently gave general definitions of the fundamental units to be found in any computing equipment, the principles by which computers in general would operate on data, the general theory of programming and control, and the relation of these artificial computing systems to the natural computing systems of the human brain. It was these general principles that could be and were utilized by later designers of computing equipment.

Von Neumann's report began by giving a description of the structure of an automatic computing system, including the purpose of the machine and the process which must be completed to carry out this purpose:[10]

An automatic computing system is a (usually highly

[9] "First Draft of a Report . . .," section 1.1.

[10] Ibid., sections i.i-i.4.

> composite) device, which can carry out instructions to perform calculations of a considerable order of complexity . . .

This purpose is general enough to apply to any general purpose electronic computer. He then explained how any such machine would carry out this task:

> The instructions which govern this operation must be given to the device in absolutely exhaustive detail. They include all numerical information which is required to solve the problem under consideration . . . These instructions must be given in some form which the device can sense . . . All these procedures require the use of some code, to express the logical and the algebraical definition of the problem under consideration, as well as the necessary numerical material.
>
> Once these instructions are given to the device, it must be able to carry them out completely and without any need for further intelligent human intervention. As the end of the required operations the device must record the results again in one of the forms referred to above. The results are numerical data; they are a specified part of the numerical material produced by the device in the process of carrying out the instructions referred to above.

The generality of this definition is apparent. Never did he refer to the specific machine being planned. In fact, he stated, for example, that instructions could equally well be given in a number of different forms, which he listed as illustration:

> punched into a system of punchcards or on teletype tape, magnetically impressed on steel tape or wire, photographically impressed on motion picture film, wired into one or more, fixed or exchangeable plugboards--this list being by no means necessarily complete.

Von Neumann recognized that technological considerations imposed constraints on structure, but that their import in the overall design of the machine was secondary. Thus, he was indifferent, as far as the theoretical purposes of this report, to the particular technology used.

In the report von Neumann also gave a characterization of the five basic units that any automatic computing system must have and the way in which they function. Again, the characterization was completely general, not specific to the EDVAC machine. The five fundamental units, as von Neumann described them, are as follows:

(a) a central arithmetical unit CA. The purpose of this unit is to carry out the fundamental arithmetical operations that occur frequently in using the machine as a computing device. As von Neumann pointed out, it was most typical to have the machines be able to do addition, subtraction, multiplication, and division, but that many other operations could supplement or replace these four operations.

(b) a central control CC. The purpose of this unit is to insure that the machine carries out the proper sequence of operations according to the specific instruction given for a particular problem inserted into the machine--whatever that problem might be.

(c) a memory M. The purpose of this unit is to store intermediate results in arithmetic computations, to store instructions, to store additional numerical information such as boundary conditions, to store tables of specific functions called frequently in the course of computation, and the storage of data for statistical and sorting purposes. These three units, CA, CC, and M, constitute the internal workings of the machine. There are then two additional units which permit interaction between the internal working of the machine and the outside world (the operator of the machine). These two units are:

(d) an input unit I. The purpose of this unit is to transfer information from the outside recording medium of the device R to the internal parts of the machine, CA, CC, and M.

(e) an output unit O. The purpose of this unit is to transfer information from CA, CC, and M to R.

Von Neumann's characterization was the first purely logical characterization of the computing machine. In previous work, each of these units and their functions had been described, but it had always been in the context of a particular machine. More important, in each of these cases the logical design had been confused with the development of the circuit design for the particular machine. Von Neumann realized that there were universal logical functions of these machines that transcended the particular circuit design of any one machine, and he was the first to isolate these logical characterizations from circuit considerations. Von Neumann was drawn to this because of his interest in the science of computing rather than in the engineering of computing. However, the distinction was useful to future computer engineers as well, for it enabled them to separate logical questions from circuit questions. In fact, for a number of years computing machinery logical design followed the logical specifications of von Neumann's report.

The draft report did not discuss in great detail the consequences of or the requirements for the stored programming designed for the EDVAC and described in the draft report. However, von Neumann did devote attention to stored programming in "On the Principles of Large Scale Computing Equipment," which was based on a lecture he gave at the Moore School in May, 1946. He first pointed out that if a machine were to be general purpose, it must not have many connections permanently wired in; otherwise, it would lose all of its flexibility.

> Due to its very nature a general purpose computer had only a very few of its control connections wired in. Apart from certain main communication channels these fixed connections are usually those which suffice to guarantee the device's ability to perform certain of the more common arithmetic processes, such as addition, subtraction, multiplication and possibly division or square roots. It is the function of the control organ and its associated memory to make and unmake the balance of the connections needed to carry out the routine for a given problem.[11]

[11] "On the Principles of Large Scale Computing Machines," Collected Works, V, p. 25.

Thus, there must be some means for making temporary connections for each particular computation. Von Neumann described two alternative methods used in existing machinery for making these connections: the method of making all the connections for each particular problem before the program begins running; and the method of establishing connections as the need arises during the computation, with the instructions for the necessary connections being stored by some means as a paper tape. The first technique, he stated, had the disadvantage of requiring a long preparation time before each computation, long enough to dominate and thus nullify the high speed of the electronic computing. For example, on ENIAC, a machine of this type, it might take eight hours to set up a problem which the machine would compute in five minutes. Another disadvantage was that such a programming scheme required a large number of connections and thus a large number of vacuum tubes, each of which was being used for one purpose only and thus was being used inefficiently. Moreover, under this technique, there was a fixed, limited number of connections that could be made, so there was a limitation in the programming that could be done on such a machine.

The second technique had the advantage of having the flexibility of an unlimited number of connections. However, the first scheme did have one serious advantage over the second in that, once the connections were made, the entire computation could take place at an electronic rate. However, the other one required connections to depend on gaining instructions mid-computation from some source such as a paper tape and making the connection by any of a number of means. This slowed the computation considerably from the electronic rate.

Von Neumann's plan was to modify the second scheme so that the instructions for the machine could be properly encoded as digital information which could then be stored in the electronic memory and be operated on like other stored data in an electronic way. However, the feasibility of such a scheme rested on the possibility of designing a machine which had logical control and transfer of information from memory which was fast enough to not dominate multiplication time, the fundamental unit of computation time on the computer, and also on coding procedures which allowed automatic modification of orders in the midst of a computation.

> It should be added that this technique of automatic substitutions into orders, i.e. the machine's ability to modify its own orders (under the control of other ones among its orders) is absolutely necessary for a flexible code. Thus, if a part of the memory is used as a "function table", then "looking up" a value of that function for a value of the variable which is obtained in the course of the computation requires that the machine itself should modify, or rather make up, the reference to the memory in the order which controls this "looking up", and the machine can only make this modification after it has already calculated the value of the variable in question.[12]

[12] Ibid., p. 32.

However, he gave general arguments showing the feasibility of each of these requirements, pointing to existing technology for illustration or verification of particular points.

In assessing credit for the priority of stored program computing, it would be fair to say that Eckert and Mauchly made one contribution, von Neumann a second, and the Manchester group a third. Eckert and Mauchly were the first to work out all the engineering and design features for a stored program computer. Von Neumann was the first to publish an account of stored programming and to provide it with a theoretical framework, considering both its logical structure and its ramifications for the theory of computing. However, the Manchester group was the first to have in operation (with their rudimentary model discussed in Chapter Three) an actual, stored program computer.

However, none of these statements address the crucial issue of who developed the basic idea of the stored program. Eckert and Mauchly claim they were the first to arrive at the idea. In their dispute over the issue with von Neumann they point out that there had been plans to make ENIAC an automatic computer with the capability of handling its own instructions before von Neumann joined them, but that the exigencies of the war required that they relinquish the plan to design equipment with these capabilities. However, the fact that von Neumann could later join the group and suggest a simple alteration in the function tables which was significantly closer to stored programming belies Eckert and Mauchly's claim. It suggests that they had the general idea for an automatic machine, but that they did not understand the principle behind stored programming--as von Neumann did--of treating instructions as numerical information in the same way data was treated. This principle was clear in Turing's universal machine, and it could be there where von Neumann's idea of stored programming originated.

There was acrimonious dispute over priority and credit concerning other issues involving EDVAC. Eckert and Mauchley intended to patent many of the design features of EDVAC in their own names. To that end, they sent out letters to the engineers on the EDVAC project asking them to waive their patent rights. They further submitted disclosures to the legal branch of the Ordnance Department, describing the patent items they intended to claim. Von Neumann objected that patent rights should not be given to individuals on this project. Although his reasons are not entirely clear, it seems that he felt the development of computer science should be kept as free as possible of patent hindrances, that there should be as much free interchange of information about computers as possible. One historian[13]

[13] See Nancy Stern's dissertation, State University of New York at Stoney Brook, 1978, "From ENIAC to UNIVAC."

credits this to the difference between Eckert and Mauchly's industrial outlook and von Neumann's academic outlook. There is also reason to think that von Neumann was not averse to being given the fame, if not the financial reward, for some of the design details of EDVAC.

> May I say this. As far as we are personally concerned we do not anticipate having any interest--financial interest . . . and as far as that is concerned we are agreeable. We do not want to contribute . . . to our ideas on the thing which will make the field less.[14]

That is, he did not want to have developments credited to him if they would prevent the further development of computer science.

A meeting was called at the Moore School on 8 April 1947 to discuss patent matters pertaining to EDVAC. That von Neumann was not disinterested in the patent issues is indicated by his unsolicitedly bringing a patent lawyer to the meeting to represent his and Princeton's interests. The minutes of the meeting indicate a very strong difference of opinion between von Neumann and Goldstine on the one hand and Eckert and Mauchly on the other.

One interesting fact to come out of this meeting was von Neumann's attitude as to his role in the development of EDVAC. He was willing to concede that there was a period (called

[14] Minutes, p. 7.

"phase I" in the discussion where work was done on EDVAC before von Neumann entered the project, presumably for which he should receive no credit.[15]

> There are certain items which are clearly one man's . . . the application of the acoustic tank to this problem was an idea we heard from Pres Eckert. There are other ideas where the situation was confused. So confused that the man who had originated the idea had himself talked out of it and changed his mind two or three times. Many times the man who had the idea first may not be the proponent of it. In these cases it would be practically impossible to settle its apostle.[16]

Yet, later in the discussion, von Neumann was less generous in giving all this credit to Eckert and Mauchly concerning the acoustic delay lines and related technology.

> Dr. von Neumann: Period I you described was before the acoustic tank came in.
>
> Dr. Mauchly: Including that. First acoustic memories and numbers, then acoustic memory, then the function table to be used as a switch and dated the time before you came in.
>
> . . .
>
> Dr. von Neumann: I think we discussed a number of serial items. Remember very clearly that I proposed one for another type. I think that area is listed [in the joint work and should not be given as individual credit to you].
>
> . . .

15 "We might agree to Phase I," von Neumann, Minutes, p. 24.

16 Von Neumann, Minutes, p. 8.

Dr. von Neumann: I suggested the use of . . . |minutes incomplete| . . . as an output to . . . |minutes incomplete| whether that was a wire or what.[17]

However, his attitude towards "Phase II", the time after he joined the project, was that the general idea for EDVAC was a joint effort and credit should not be given to any individual, but that particular limited, special developments could possibly be credited to certain people. In fact, after the meeting to determine priority rights and after von Neumann realized that Eckert and Mauchly were going to file for patents in their own names, he decided to file his own statemtne concerning his personal contributions to the EDVAC project. They included:[18]

1. A new code for enabling the operation of the EDVAC.
2. The serial performance or progression through the system of the various arithmetical operations required for the solution of a whole problem.
3. The use of the "iconoscope" as a memory device.

He saw, as did the Patent Office, his report as a formalization of a group effort. Moreover, the Patent Office ruled that von Neumann's report was a publication, so part of the public domain, and thus any items discussed in it could not be given individual patent credit. This was a victory for von Neumann, a loss for Eckert and Mauchly.

[17] Minutes, p. 20.

[18] From "Informal Report in re Disclosure of John von Neumann's First Draft of a Report on the EDVAC," written shortly after 2 April 1946, as quoted in H. H. Boldstine, The Computer from Pascal to von Neumann, p. 224.

Work continued on EDVAC until 1949. It was then delivered to the Ballistics Research Laboratory, where it was in operation from 1951 until 1962. However, many of the principal designers of EDVAC left after the initial development period ending in 1946, and many changes were made after they left. Eckert and Mauchly left the Moore School to form their own computer company. Von Neumann returned to the Institute for Advanced Study to develop his own computer. He took several members of the EDVAC staff, most notably H. H. Goldstine and Arthur Burks, with him to the Institute to work on this project.

The Institute for Advanced Study Computer

Von Neumann's aim when he left the Moore School was to build a computer whose memory was based on the iconoscope (cathode ray tube) as used in television, instead of on the acoustic delay line which had been used on EDVAC. The advantage he saw in the iconoscope was its more rapid access to memory than was possible with the acoustic delay line because of the random access to information in the iconoscope as opposed to the cyclic access through the delay line to information in the EDVAC system. Soon after he had arrived at the Moore School he had considered the possiblities of such a system, and, as the war neared its end, he discussed the possibilities of developing such cathode ray tubes for computing with the Radio Corporation of America (RCA) research division in Princeton.

During 1945 von Neumann began making inquiries concerning the possibilities of constructing a computer. His hope was to convince the Institute for Advanced Study to support such a project. The difficulty was that the Institute was an unlikely site for such a project because of its limited facilities and its purely theoretical, as opposed to experimental, bent. There was not a single laboratory facility at the Institute prior to von Neumann's computer project.

In March, 1945, von Neumann received a feeler from Norbert Wiener asking whether he would be interested in accepting a position as head of the department of mathematics at Massachusetts Institute of Technology if facilities were available for working on his computer projects.[19] This provided von Neumann with the initial leverage to convince the Institute that it might lose him if it were not willing to make available there the facilities he needed for his computer project. Von Neumann strengthened his case by soliciting offers from Chicago and Columbia as well. By late 1945 von Neumann, with the help of Ostwald Veblen, had convinced Frank Ayedlotte, director of the Institute, to involve

[19] Letter from Wiener to von Neumann, March 24, 1945, Von Neumann Archives.

the Institute in a computer project, and initial planning began. After that time, Aydelotte was an unflagging supporter of von Neumann's project.

The computer project was designed to be carried out under the joint sponsorship of the Institute for Advanced Study, Princeton University and RCA, which had a research division in the Princeton area and which was to carry out the research and development of the electronic tubes for the machine. There was concern at first that government funding, at least from the military, would be difficult to obtain because of the competition from EDVAC. Thus, the group turned first to the Rockefeller Foundation for support. In the end support did not materialize from the Rockefeller Foundation, but, due to the different nature of the proposed IAS machine, the government eventually did support the project. Initial support came from Army Ordnance. Continuing sponsorship came from the same agency, together with Office of Naval Research, U.S. Navy, U.S. Air Force, and the Atomic Energy Commission.

Despite the heaby government support, von Neumann made it clear in his contractual agreement with the government for the machine that it was to be used as an experimental, scientific device, not as a tool for production jobs. This is clear from a letter[20] from von Neumann to Admiral Bowen, of the Office of Research and Inventions, Navy Department, dated January 23, 1946. In the letter von Neumann stated the principles under which the contract between the Navy and the IAS project were to be drawn:

> The performance of the computer is to be judged by the contribution which it will make in solving problems of new types and in developing new methods. In other words, it is a scientific tool, to be used in research and experimental work, and not in production jobs.

Von Neumann's plan was that additional machines should be built to carry out computations that the IAS machine had shown were possible, thus freeing the Institute machine for strictly experimental problems.

> Thus, if a new type of problem arises which can not be handled on other existing computing machines, and which this computer may seem likely to solve, then the new computer should be used on the problem in question until a method of solution is developed and tested--but it should not be used to solve in a routine manner further problems of the same type. The policy should rather be, to have further electronic computers of this new type built, which will belong, say, to the Navy, and do the routine computing jobs. The Institute's computer should be reserved for the developing and exploring work as outlined on the preceding page [above], and this should be the objective of the envisioned project and the Institute's function.

[20] In fact, the letter was never sent. However, the purpose of the letter (as the letter itself indicates) was merely to put in writing a telephone conversation between the two of the same day. The following three quotations are from this letter. The letter is in the Von Neumann Archives, Library of Congress, Washington, DC.

In fact, three copies were built for the Atomic Energy Commission and another by the University of Illinois for the Ballistics Research Laboratory.

Von Neumann also outlined the criteria for the selection of problems to be worked on by the Institute computer. They are the same sort of problems which he had hoped to solve on the ENIAC: problems of interest to the applied mathematician which can not be handled by more classical techniques. It is clear that the choice of problems is to be governed by scientific considerations, not by consideration of specific government needs.

> It is then clear that the problems to be put on the computer for development and exploration of methods should be judged by fairly general viewpoints, namely: Are they of a new type, not amenable to handling by existing methods and machines? Are they typical of a wide and important evolution of applied mathematics, in the direction of the major interests and needs of the present or the immediately foreseeable future? In other words, the criteria have to be scientific, in the meaning of this term in applied mathematics and mathematical physics.

It was not that von Neumann was uninterested in the government research needs and only interested in funding his scientific project. Von Neumann was a fiercely patriotic immigrant American, who was heavily involved both during and after the war with government research and agencies. He was proud of these accomplishments and pleased to lend his expertise to government research. Rather, he thought that the appropriate role for the Institute for Advanced Study, an institution with a tradition of purely theoretical research, was the investigation of the theoretical, scientific uses of the high-speed electronic computer.

Not only would this theoretical role for the Institute be most efficient, it is likely the only arrangement to which the Institute would agree. Aydelotte, Director of the Institute, was able to convince the Institute Board to approve the computer project on the grounds that it was a scientific project. He likened the computer to the telescope, as an experimental instrument necessary to any research, theoretical or experimental, in the scientific field. This was clever, for the need for the large-scale telescope was clearly appreciated by the scientific community.

> I think it is soberly true to say that the existence of such a computer would open up to mathematicians, physicists, and other scholars areas of knowledge in the same remarkable way that the two-hundred-inch telescope promises to bring under observation universes which are at the present moment entirely outside the range of any instrument now existing.[21]

In fact, Aydelotte did not play on the patriotic feelings of the Board in his attempt to convince them of the need for an Institute computer. Rather, he emphasized the prestige that would accrue from having the premier computing device in the country, on which

[21] Minutes of Regular Metting of the Board of Trustees, Institute for Advanced Study, 19 October 1945, as quoted in H. H. Goldstine, The Computer from Pascal to Von Neumann, pp. 243-244. The next quotation is from the same source.

"scientists from all over the country" would come to do research.

> This means, of course, that it would be the most complex research instrument now in existence. It would undoubtedly be studied and used by scientists from all over the country. Scholars have already expressed great interest in the possibilities of such an instrument and its construction would make possible solutions of which man at the present time can only dream. It seems to me very important that the first instrument of this quality should be constructed in an institution devoted to practical applications.

Details were settled and staffing began in 1946. Von Neumann was named director of the project and H. H. Goldstine was named assistant director. Engineering design was headed by Julian Bigelow, and later James Pomerene. Goldstine and von Neumann worked on the logical design of the machine and on its mathematical capabilities. The extensive work on meteorology programmed on the machine was directed by Jules Charney and von Neumann. Von Neumann proved to be an able administrator, handling personnel problems smoothly and having an amazing capacity for keeping abreast of and contributing to every aspect, whether mathematical, logical, or engineering, in the construction of the machine.

Since the genesis of the idea for the Institute computer was mainly due to von Neumann, and since it was through the abilities, the prestige, and the perseverance of von Neumann that the machine achieved physical reality, the IAS computer gives a better picture than does EDVAC of von Neumann's vision of the computer and of its applications. In a letter of March 19, 1946, von

Neumann described his hopes and plans for the IAS machine to M. H. A. Newman, the Bletchley and Manchester computer scientist.

Von Neumann's description of the aim of the machine as being a "scientific exploration tool" is fairly consonant with the aims for the machine von Neumann had discussed with Admiral Bowen (described above) and will not be mentioned further here. However, von Neumann is more specific with Newman concerning the actual types of questions the Institute staff would be trying to answer with their experimental apparatus.

> . . . we propose to use it as a "scientific exploration tool", i.e. in order to find out what to do with such a device. That is, I am convinced, that the methods of "approximation mathematics" will have to be changed very radically in order to use such a device sensibly and effectively--and to get into the position of being able to use still faster ones. I think that the problem in this respect is partly logical and partly analytical, since finding suitable approximation methods and finding and coding the proper machine "setups" may be the main bottleneck. We want to do a good deal of mathematical and logical work in parallel with the engineering development, a good deal more, with the machine as the "experimental tool", when the machine is ready.

The letter also summarizes the technical characteristics von Neumann hoped to design into the machine. Paraphrasing them from his letter, they include:

(1) Use of the binary system, with precision of approximately 2^{-40}. Equipment would be provided for binary decimal conversion for input and output, so that the machine could be programmed and answer in decimal.

(2) Capability of reaction times on the order of 10^{-6} seconds.

(3) Addition, subtraction, multiplication, division, and square rooting as fundamental operations with the following maximum computation times (in seconds): addition and subtraction, 10^{-5}; multiplication, 10^{-4}; division and possibly square rooting, $2(10^{-4})$.

(4) A substantial memory, capable of storing both instructions and numerical data (such as initial information, functions, intermediate results). There should be capability of storing 4000 40-digit groups. This information should be accessible (located, read out, cleared, or read in) in no more than 10^{-5} seconds.

(5) Input and output in tape medium, presumably magnetic. There should also be means for the direct graphing of output on oscilloscope screens.

(6) A self-checking mechanism, with the capability of at least determining when and where there is a simple failure (one in which there are not two or more compounding, simultaneous failures).

(7) Use of no more than 2000 ordinary, receiving tubes and no more than 100 special tubes.

Von Neumann was also careful in planning the speed and capacity of the various units of his machine before construction so as to insure ability to solve the problems of applied mathematics he wanted his machine to solve. He estimated that a typical ballistics problem required approximately 10^8 multiplications and that the multiplication time in such a problem was roughly one-third to one-eight of the total computing time. Thus, to complete the problem in a 40-hour week, he estimated that multiplications must be carried out in 1/2 to 1/5 of a millisecond.

Von Neumann realized that the overall speed of the machine was determined by its slowest internal part, that a bottleneck would occur at this part which would defeat the advantages of the other units. Thus, he aimed to design a machine where all the units worked at speeds such that there would be no bottlenecks. Since machine multiplication involved (on the average) five transfers of information between the memory and the arithmetical organ, von Neumann estimated that an access to the memory should take about 50 microseconds to be consonant with multiplication speed. Since each such transfer of information involved several orders from the control organ, each such order should take only a few microseconds. He finally argued that, due to the significantly smaller amount of information handled in input and output as compared with the inner working of the machine in the intermediate computations on a problem, the input and output equipment could be much slower than the internal equipment without concern over a serious bottleneck. Thus, he designed the machine so that data could be input or output at the rate of 500 words per second.

Von Neumann organized his design of the IAS machine according to the logical characterization of the machine he had arrived at in the Draft Report for EDVAC: arithmetical unit, control unit, memory, input, and output. The most salient features of each of these are described below.

The arithmetic unit was comprised of ordinary vacuum tubes, with the ability to carry out an elementary operation in approximately a microsecond. The circuits were designed with reliability in mind, and so were set up to respond to any electrical impulse over a certain minimum threshold no matter what were the details of its electrical form. The overall design of the arithmetic unit was modelled after the design of a typical desk calculator. For example, for multiplication the computer had three registers (for temporarily holding the multiplier, multiplicand, and partial products as they accrued), an adder (capable of adding two quantities), and a counter (which stopped the operation when all steps were completed).

One important feature on the IAS arithmetical unit was its ability to add the digits of two numbers simultaneously rather than serially, one at a time. This was possible because the store on the IAS machine had paralled rather than the serial access of EDVAC. Thus each of the digits could be added in paralled and the addition completed in roughly one instead of 40 units of adding time.[22] For each digit there was an adder with three inputs (one for the appropriate digit of each of the two numbers and one for carrying from the next smaller digit) and two outputs (one for the sum and one for the carry to the next larger digit).

The control unit was also comprised of ordinary vacuum tubes with the reliability feature described above for the arithmetic unit. The control functioned by withdrawing the instructions, a pair at a time (as they were encoded as twenty bits per instruction in 40 bit words), and causing the machine to execute them. Facility was included to change instructions, with the aid of the arithmetic organ, in the midst of a computation, so that a given order could attain different values as it was repeatedly used in the computation. Unlike EDVAC, which used a four-address code, IAS used a simple one-address system. Each order consisted of a number expressing the location ("the address") in the memory and a number expressing the operation to be performed on the contents of the memory stored in the location given by that address. These two numbers could be encoded in twenty binary

[22] The addition was completed in several stages. First, each digit of one number was added to the corresponding digit in the other number. Forty registers carried out these digit additions simultaneously. All the carries resulting from the first stage were then added simultaneously in the forty adders to the first sum. Subsequent additions of carries were made until there were no more carries.

digits; so two instructions were stored at one address in the memory in order to economize on space.

The fact that IAS could perform arithmetic operations on the numbers which were codes for instructions set it apart from many earlier machines. For example, the Bell Laboratory relay machines had instructions stored on paper tapes which could not be modified by the machine in the course of a computation. This meant that the IAS machine could carry out in a simple manner computations which were not even possible by its competitors. Given appropriate storage space, IAS could compute all primitive recursive functions. Yet, these paper tape machines could not compute such simple primitive recursive functions as $x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot \ldots \cdot x_n$ for arbitrary integer values of n .

However, in order to gain the ability to modify instructions internally in the course of a computation required that the process of coding of problems be more complicated, as von Neumann, himself, explained:[23]

> It is partly for this reason [internal modification of instructions] that the coding of problems is not a mere transliteration of symbols from those of the mathematical analysis to those understood by the machine. Actually, the steps are as follows: a) the formulation of the physical situation in a purely mathematical terminology; this formulation is usually highly implicit and quite unsuited for digital calculation; b) the reformulation of the equations in

[23] Report on IAS computer project, 1946, pp. 12-13, Von Neumann Archives.

> explicit finitistic form suitable for numerical calculation; c) the analysis of the logical structure of the final formulation, usually in the form of a "flow diagram" showing all inductions; d) "coding" proper, based on the "flow diagram", that is, the final expression of all operations and quantities in machine terms.

In fact, for the logical analysis of the computation procedure and for the purpose of obtaining the appropriate coding, von Neumann and H. H. Goldstine invented the first flow charts.

The memory unit caused the most engineering difficulties in the early computers. There were mutually destructive requirements of the storage in these machines because there was a desire for as near an infinitely large storage area as was possible and, yet, at the same time there was a desire for electronic access speeds to these stores. Von Neumann realized that the best possible compromise between these mutually exclusive demands was a hierarchy of memories, each larger than the preceding one, but each also slower than the preceding one. The first level of storage was by electronic storage tube, or iconoscope, which was designed by the RCA research laboratory. In it information was stored in a dielectric substance inside an electronic tube by charging miniature condensors. The second level of memory was given by magnetic drums. These stored information by magnetizing small areas on the face of a cylindrical drum. Tertiary storage was provided on punched cards and punched tapes outside the machine, with access through the input and output devices.

The iconoscope storage was important because it enabled von Neumann to dispense with the serial mode for the machine. In EDVAC, to access a particular datum of information, one would have to cycle through the delay line until the particular location came up. This caused the overall design of the machine to be serial. However, the electrostatic storage tube provided equally fast access to any location on its surface. Thus, the machine could be designed to run in parallel, since it did not have to wait for serial access to information. The result was quicker access of material, quicker arithmetical operations (since the different digits could be added at the same time, rather than serially a pair added at a time), and in general more rapid computation times than with the serial EDVAC. In fact, with less equipment, IAS was five times faster than EDVAC.

When von Neumann began working on the design of the IAS computer, he considered using the electrostatic storage device designed by the Manchester group. Instead, the Institute designed their own, but similar, device. It was the first electrostatic storage used in the United States. It was also the Institute machine design that was more important to the next generation of computing machinery.

The input-output equipment for the Institute machine was also a new technological development. This unit consisted of several pieces of teletype equipment modified specifically for the

Institute machine by the National Bureau of Standards, some electronic circuitry to connect the various pieces of the input-output equipment with each other and with the remainder of the machine, and a mechanism to move magnetic wire at various speeds. The speed of this equipment was that of electro-magnetic and not electronic equipment. However, von Neumann was careful to prove that this would not vitiate the internal speed of the machine.

To operate the machine, a keyboard operator would type onto teletype tape the instructions to be given to the machine. Once completed, this teletype tape would be entered into a second teletype machine and another operator would again type the instructions for the machine, this time into the second machine. The second piece of teletype equipment was specially wired so that it would only accept the instructions typed in if they compared exactly with the ones typed on the first tape. This technique, known later as multiplexing, of having more than one piece of equipment carry out the same task and only continuing if the results agreed, was an entirely novel method designed by von Neumann for making computations highly reliable given that the users of the machine and each piece of equipment in the machine were not all that reliable. Once the tapes agreed, the second tape was run synchronously with the slowly moving magnetic wire on a wire drive and the data was transferred onto the wire. The magnetic wire could then be run quickly to feed the information to the interior of the computing machine.

The output consisted of the same wire drive as in the input, a relay rack of circuits, and a teleprinter. The results of the computation were delivered to the magnetic wire. When the operator was ready for the results, the wire was run at a slow speed, the message was picked up and run through the circuits, which in turn activated the appropriate keys of the teleprinter.

The Institute computer was invaluable for many military and atomic energy computations. However, one of the most novel and important applications of the Institute computer was its use in numerical meteorology. Von Neumann's interest in the subject was entirely in keeping with his interest in fluid dynamics. Ever since the first world war scientists had been trying to predict the weather by calculating mathematically the fluid flow of the atmosphere. However, the particular differential equations involved required that there be a large number of initial data points. Thus, the computing required an extensive amount of time. A man with a desk calculator, working eight hours per day and thirty days per month, would take two years to complete one twenty-four hour forecast of a quality comparable to a synoptic forecast, extrapolating from the present flow using established rule-of-thumb methods. ENIAC had been used for some weather forecasting by von Neumann and Jules Charney, a well-known

American meteorologist. Unfortunately, the limited memory and speed of ENIAC meant that it took 48 hours of computing time to do the twenty-four hour forecast.

In 1952 von Neumann and Charney completed such a forecast on the Institute computer in three hours and would have been able to complete it in forty-eight minutes had the machine been operating at full speed. This made it feasible for the first time to use the computer to do weather forecasting. In fact, by 1953, the Institute computer could make these calculations in six minutes. For simplicity and shorter computing time, all of these models of atmospheric flow had been two dimensional. The computations had involved only the flow at 18,000 feet above sea level. Von Neumann and Charney, once they were successful with the one level prediction, went on to calculate two and then multi-level models. One of the early three layer models was completed by the Institutein under an hour--still an acceptable time for practical use. Of course, the multi-layer models gave better accuracy than the one layer model. There was an added bonus provided by the three layer model, for one of its predictions showed cyclogenesis. This was the first mathematical model of the formation of cyclones. From the early 1950's on, the computer became a valuable and well-used tool in weather forecasting, and meteorologists looked to the work at the Institute for their guidance.

As important as its applications were, the Institute computer was probably more important as the prototype of the next generation of American computing machinery. Use of the binary system, parallel mode of operation, and one-address coding were all important features which were initiated on the Institute computer and became standard features on subsequent American computing equipment.

The plans for the Institute machine were discussed in a series of related papers by von Neumann, H. H. Goldstine, and Arthur Burks, entitled: "Preliminary Discussion of the Logical Design of an Electronic Computing Instrument," written in 1946 by all three, and a three part "Planning and Coding of Problems for an Electronic Computing Instrument," published between 1946 and 1948 by Goldstine and von Neumann alone. Typical of von Neumann, these papers were intended to contribute to the science of computers. The papers intended to study the general theory of computers, not just a description of the design of the Institute computer. In fact, in these reports, the authors discussed the advantages and disadvantages of all the existing computer technology, how computers should be built, and how computers should be coded and programmed. Besides the Institute machine, the following computers were built according to the specifications in these reports: ORDVAC, ILLIAC, AVIDAC, ORACLE, WEIZAC, BESK, DASK, CSIRAC, and JOHNNIAC.

Von Neumann's draft report on EDVAC had concentrated on the logical character of computing devices and had mentioned technical details only as illustration and in passing. These reports concentrated, however, on the technical details, using as the logical framework as described in the draft report. The various technologies available were considered for each of the five fundamental units of a high-speed computer as described in the draft report. For example, von Neumann compared the advantages and disadvantages of trigger circuits, gas tubes, electro-mechanical relays, before choosing the iconoscope as the best means of storing information. He provided a detailed account of how the adder and the multiplier worked, not on the level of the electrical circuitry, but on the functional level. This engineering report was dominated by von Neumann's mathematical discipline and his desire to develop a science of computers. Thus he made very formal mathematical arguments in his engineering discussions. He provided careful probabilistic arguments as to the speed, time of computations, length of computations, etc. He discussed the problems of round-off errors and the theory of approximation in a mathematical fashion. He provided a detailed mathematical analysis of the use of the binary system in the computer.

In the draft report on EDVAC von Neumann had carried out no analysis of the problems of programming or coding, although he had done some programming and coding in connection with problems he had run on ENIAC. Thus, during 1946 and 1947, von Neumann and Goldstine started to analyse the general problems inherent in the actual coding and programming procedure. As an earlier quotation points out,[24] von Neumann and Goldstine realized that it was not entirely straight-forward to take a mathematical problem and put it into the computer to solve.

There were two major steps in the programming of a problem for the computer. First, one had to decide which of those sequences of finitary operations which could be handled by the computer would effect the appropriate computation. As an aid to this project, the two men developed a crude geometric system for displaying the logical sequence of operations that must take place in the computation. This technique was refined over 1946 into a useful and sophisticated tool, which they called "flow diagramming." This was, in essence, a sophisticated version of our modern system of flow charts.

The second task that must be completed to solve a problem on the computer was to do the "static coding," i.e., to write down a set of rules, entirely analogous to the description rules of the Turing machines, which described the internal working of the machine. These rules were "static," as opposed to the "flow" of operations shown on the flow diagrams, in that they were fixed

[24] See the quotation to which footnote number 23 of this chapter is affixed.

rules only one of which applied at a time depending on the conditions of the machine at the time.

Volume II of the Institute's report, on planning and coding, was devoted to the study of the methods of coding and the philosophy governing it. The stated desire of von Neumann and Goldstine was to develop coding with the following characteristics:[25] (1) simplicity and reliability of the engineering solution required by the coding; (2) simplicity, compactness, and completeness of the code; (3) ease and speed of translation into the code of human language and ease of finding errors; and (4) efficiency of the code in allowing the machine to work near its full intrinsic speed.

As illustration of the problems confronted and the possible solution of coding problems, they provided both the flow diagrams and the static coding for a number of problems. They showed how to compute and store an arithmetic operation like $(au^2 + bu + c)/(du + e)$; how to convert between binary and decimal systems; how to carry out double precision, i.e., work with numbers that required more than 40 binary digits and so could not be stored completely in one location; how to solve analytical problems such as computing definite integrals or interpolating a function of one variable with tabulated values; and how to carry

[25] "Planning and Coding . . .," Von Neumann, Collected Works, V, 81.

out combinatorial problems, such as placing a random sequence of numbers in monotone order or meshing two sequences, where the task is primarily logical rather than computational. Finally, they showed how to carry out subroutine programming, enabling them to use the type of programming they had already demonstrated as whole units in a larger, more complicated program.

In sum, von Neumann contributed many important developments to the early history of computer science. He improved ENIAC, was a major figure among the group that designed EDVAC, and was chiefly responsible for the design and development of the Institute computer. These computers were important both for providing the first high-speed computing ability in the United States and for being protorypes of the next generation of American machines. He was integrally involved in particular important technical developments, including: stored programming, use of the binary system, serial operation, one address coding, flow diagramming, separating logical from circuitry design, development of the science of programming, and characterization of the logical design of computers.

His contributions were not restricted to technical design. He was extremely valuable to the early development of computer science by using his reputation as a first rate scientist to promote computer science, both at the Moore School and at the

Institute for Advanced Study. Until this time, computers represented a technology with an uncertain future and with little apparent usefulness to science.

Perhaps von Neumann's greatest contribution was his efforts to turn the engineering discipline of computers into a science, based on general laws incorporating the principles of mathematics, logic, and physics, and important to scientific research. Part of this von Neumann contributed by showing there was a general theory of computing machines involving principles which held about all computing machines and which could be discussed without detailing the engineering of a particular machine. Part of it involved his development of a plan for the use of the computer as an heuristic device in the solution of scientific problems. Part of it involved his demonstration of the profound utility of the high-speed electronic computer in the solution of many important applications, such as fluid dynamics, atomic energy problems, and numerical meteorology. Without von Neumann's influence, it might be inappropriate to affix the name "science" to the discipline involving computers.

Chapter Five: The Conceptual Revolution in the Information Sciences

The last two chapters have discussed Turing and von Neumann's contributions to the development of electronic computing equipment and to the general design features used in subsequent generations of such machines. This type of research is typical of the history of the computer field. Most tend to equate the history of computer science with the history of computing machinery. The phenomenal growth of a new generation of more powerful machines every few years has overshadowed other important developments in computer science. The remainder of this dissertation will examine a theoretical revolution in the computer and information sciences which has been obscured by this technological bias. This chapter will consider the general development of this conceptual revolution. The remaining two chapters will examine the contributions due to Turing and von Neumann.

In the decade following the end of the second world war a small group of mathematically oriented scientists developed a mathematical theory of information and information processing.

For the first time, information became a concept worthy of study on its own. It was given the status of a physical parameter, such as mass, which could be quantified and studied mathematically. This theory was designed to apply to both machines and to living organisms. The major figures involved included Alan Turing, John von Neumann, Claude Shannon, Warren Weaver, Warren McCulloch, Walter Pitts, Norbert Wiener, W. R. Ashby, and a host of less important figures. They came to the subject from mathematics, electrical engineering, psychology, biology, and physics. The problems they considered included a mathematical theory of communication, mathematical models of the brain, artificial intelligence, cybernetics, automata theory, and homeostasis.

Roots of this work could be found going back as far as the middle of the nineteenth century in physics, mathematical logic, psychology, and biology. In particular, the work on information theory developed primarily out of the following, more traditional scientific problems:

(a) Maxwell, Boltzmann, and Szilard's work in thermodynamics and statistical mechanics, because of the close analogy of the concept of information to the concept of entropy;

(b) the development of control and communication as a supplement to the power field in electrical engineering due to the development of telegraphy, radio, and television;

(c) the study of the physiology of the nervous system throughout the first half of the twentieth century, especially through the work of Bernet on homeostasis and the internal regulation of living organisms;

(d) the development of functionalist and behaviorist theories of the mind in psychology; and

(e) the development of recursive function theory in mathematical logic as a formal, mathematical characterization of the human computational process.

Although these results appear to be, and in many ways are, quite diverse, there was a concerted attempt after the war to unify the various disciplines through a mathematical characterization of the concept of information and information processing. At the heart of this new field was the idea that an interdisciplinary approach could be used to solve problems in both biological and physical settings where the key to the problems was the manipulation or transmission of information and where the overall structure could be studied through mathematical models. For example, both the human brain and the electronic computer were considered as types of complicated information processers, whose similar laws of functioning could be better understood by the abstract results deduced from the mathematical models of automata theory.

Atypical of mathematical modelling, the branch of mathematics most conspicously utilized in information theory was mathematical logic.[1] This is explained by the fact that mathematical logic studies the laws of thought in abstraction; but, more particularly, by the fact that in the 1930's logicians had been especially concerned with a mathematical characterization of the process of computation--whether by human or by machine.

In fact, training in mathematical logic was the most salient tie between the early pioneers in information theory. Claude Shannon completed a doctoral dissertation in electrical engineering at Massachusetts Institute of Technology on the application of mathematical logic to the study of switching systems. Norbert Wiener studied mathematical logic with Bertrand Russell and explicitly admitted[2] its influence on his later work in cybernetics. Alan Turing received his doctoral degree from Princeton University for work in mathematical logic. Walter Pitts was trained by Rudolf Carnap as a mathematical logician, while his colleague, Warren McCulloch, was a physiological psychologist interested in questions concerning the learning of mathematics. The polymath, John von Neumann, made a number of contributions early in his career to two branches of logic, proof theory and set theory.

[1] However, there was also substantial use of probability (ergodic theory) and differential equations (control theory).

[2] See the introduction to Wiener's <u>Cybernetics</u>.

There is more of a similarity in the work of these men right after the war than just their use of mathematical logic to solve problems in diverse fields. There was a strong feeling that the newly discovered concept of information could tie together, in a fundamental way, problems from different branches of science. Although these scientists were coming from widely diverse backgrounds and from the vantage of widely diverse problems, they were in close contact with one another through collaboration, scholarly review of one another's work, and frequent interdisciplinary conferences. For example, Wiener introduced Pitts to Shannon's work and worked with him on problems of electronic computing at M.I.T.; Shannon was the reviewer in Math Reviews of McCulloch and Pitts' work on neural networks; and, typical of many conferences, was the Princeton meeting in 1943 organized by Wiener and von Neumann for mathematicians, engineers and physiologists to discuss problems of mutual interest concerning cybernetics and computing. The introduction to Wiener's book on cybernetics described the sense of community and common purpose among these diversely trained scientists. Perhaps even more telling were Wiener's attempts to develop an interdisciplinary science known as cybernetics around the concept of feedback information, and von Neumann's attempt to unify the work of Shannon, Turing, and McCulloch and Pitts in a general theory of automata.

The timing for the growth of this interdisciplinary information science was not accidental. Rather, it was the product of the massive cooperative and interdisciplinary scientific projects of the second world war--projects that often carried scientists to projects beyond the standard scholarly bounds of their specialties. At no other time had there been such mobilization of the scientific community. Wiener was led to the subject of cybernetics through his participation on Vannever Bush's computing project at M.I.T., his work with Y. Lee on wave filters, and his work on fire control for air-aircraft artillery--all part of war-related projects. Turing and von Neumann used expertise derived from war-related computing activities in their work on artificial intelligence and automata theory. Shannon's work on a theory of communication was the result of the tremendous development of the communications due to the development of radar and electronics during the war. Similar war connections could be drawn to the work of McCulloch, Pitts, and Weaver. The major developments in this new theory of information science are discussed below.

Claude Shannon, Warren Weaver, and the Mathematical Theory of Communication

While working on communication problems relating to telegraphy at Bell Laboratories in the 1940's, Claude Shannon became involved in developing a general theory of communication which would treat the transmission of any sort of information from one point to another in space or time. His aim was to isolate and give specific technical definitions to concepts general enough to pertain to any situation where information was being manipulated or transmitted; concepts such as information, noise, transmitter, signal, receiver, and message.

At the heart of this theory was a new concept of information. To make communication theory a scientific discipline, Shannon had to provide a precise new definition of information which transformed it into a physical parameter capable of quantification. To do this he distinguished the concepts of information and meaning. Meaning he reserved for what was actually included in a particular message. Information he used to refer to the number of different possible messages that could have been carried along a channel of information depending, say, in the case of a spoken message, on its length and on the number of alternative words which could have been chosen at each point in the message. Information in Shannon's sense was a measure of orderliness (as opposed to randomness) in that it told out of how many possible random choices one has chosen to send a particular message. The more possible choices there are, the larger the amount of information transmitted, because the actual message is distinguished from a larger number of random possibilities.

Shannon explicitly admitted the importance of previous work in the communication industry to his interest in a general theory of information:[3]

> The recent development of various methods of modulation such as PGM and PPM which exchange bandwidth for signal-to-noise ratio has intensified the interest in a general theory of communication. A basis for such a theory is contained in the important papers of Nyquist[4] and Hartley[5] on this subject. In this paper we will extend the theory to include a number of new factors. . . .

H. Nyquist was a researcher at Bell Laboratories working on the problem of improving transmission speeds over telegraph wires when he wrote his paper on the transmission of intelligence. His 1924 paper, which influenced Shannon's later work, was concerned with two factors relating to the maximum speed at which "intelligence" can be transmitted by telegraph: signal shaping and choice of codes. As Nyquist stated,[6]

> the first is concerned with the best shape to be impressed on the transmitting medium so as to permit of greater speed without undue interference either in the circuit under consideration or in those adjacent, while the latter deals with the choice of codes which will permit of transmitting a maximum amount of intelligence with a given number of signal elements.

[3] Claude E. Shannon, "The Mathematical Theory of Communication," Bell System Tech. J., 1948, pp. 31-32.

[4] Nyquist, H., "Certain Factors Affecting Telegraph Speed," Bell System Technical Journal, 3 (1924), 324-346.

[5] R. V. L. Hartley, "Transmission of Information," Bell System Technical Journal, 7 (1928), 535-553.

[6] Nyquist, p. 324.

While most of Nyquist's article considered the actual engineering problems involved with transmitting information over telegraph wires, there was one theoretical section of importance to Shannon's work, entitled "Theoretical Possibilities Using Codes with Different Numbers of Current Values." In this section Nyquist proved the first logarithmic rule governing the transmission of information.

Nyquist proved that the speed at which intelligence can be transmitted over a telegraph circuit with a given speed follows the equation $W = k \log m$, where W is the speed of transmission of intelligence, m is the number of current values which can be transmitted, and k is a constant. He argued that if there are n signal elements per character transmitted, then the total number of different characters which can be constructed corresponds to the number of sequences of current values of length n which can be constructed. Thus there are m^n possible characters. If the total number of characters to be transmitted is assumed to be constant, then m^n = constant, or, by taking the logarithm of both sides, $n \log m$ = constant . Since the speed of intelligence transmitted W is directly proportional to the line speed s and inversely proportional to n, the number of signal elements transmitted per character, Nyquist obtained the equation $W = s/n$. Substituting the above equation into this one, he concluded that $W = s \log m/\text{constant}$ -- which is of the form

W = K log m if the line speed is assumed to be constant.

Nyquist also prepared a table (listed below) which illustrated the advantage of using a greater number of current values for transmitting messages:[7]

Number of Current Values	Relative Amount of Intelligence Which Can Be Transmitted With the Given Number of Signal Elements
2	100
3	158
4	200
5	230
8	300
16	400

While Nyquist's work was empirical and concerned mainly with engineering issues, and while he used the term "intelligence," which masked the difference between information and meaning, his work was important for the first statement of a logarithmic law for communication and for the examination of the theoretical bounds for ideal codes for the transmission of information. Shannon was later to give a more general logarithmic rule as the fundamental law of communication theory, which stated that the quantity of information is directly proportional to the logarithm of the number of possible messages. Nyquist's law was a specific case of Shannon's law, since the number of current values was directly related to the number of symbols (bits of information) that could

[7] Nyquist, p. 334.

be sent. Nyquist was aware of this relation, as his definition of speed of transmission indicated:[8]

> By the speed of transmission of intelligence is meant the number of characters, representing different letters, figures, etc., which can be transmitted in a given length of time . . .

By "letters, figures, etc." he meant what Shannon later would have called "bits of information." Nyquist's table, listing the <u>relative</u> amount of intelligence transmitted, illustrated the gain in information consequent upon a greater number of possible choices. The fact that the table listed the <u>relative</u> amount of intelligence transmitted indicates Nyquist's awareness that there was an important relation between the number of figures and the amount of "intelligence" (information) being transmitted. This relation is at the heart of Shannon's theory of communication. Unfortunately, Nyquist did not recognize the significance of this relation, nor did he generalize his concept of "intelligence" beyond telegraph transmissions.

Shannon's other predecessor in communication theory, R. V. Hartley, was also a researcher at Bell Laboratories. Hartley's intention was to set up a quantitative measure whereby capacities of various systems to transmit information could be compared. His hope was to provide a theory general enough to include

[8] Nyquist, p. 333.

telegraphy, telephony, picture transmission and television over both wire and radio paths. His 1928 investigation began by attempting to establish theoretical limits of information transmission under idealized situations. This was important, for it led him away from the empirical studies of engineering towards a mathematical theory of communication.

Before considering concrete engineering problems, Hartley turned to "more abstract considerations." He began by making the first attempt at distinguishing a concept of information capable of use in a scientific context. He realized that any scientifically usable definition of "information" should be based on what he called "physical" rather than "psychological" consideration. By this he meant that information is a concept involving quantity of physical data and should not be confused with the meaning of a message.

> The capacity of a system to transmit a particular sequence of symbols depends upon the possibility of distinguishing at the receiving end between the results of the various selections made at the sending end. The operation of recognizing from the received record the sequence of symbols selected at the sending end may be carried out by those of us who are not familiar with the Morse code. We could do this equally well for a sequence representing a consciously chosen message and for one sent out by the automatic selecting device already referred to. A trained operator, however, would say that the sequence sent out by the automatic device was

> not intelligible. The reason for this is that only a limited number of the possible sequences have been assigned meanings common to him and the sending operator. Thus the number of symbols available to the sending operator at certain of his selections is here limited by psychological rather than physical considerations. Other operators using other codes might make other selections. Hence in estimating the capacity of the physical system to transmit information we should ignore the question of interpretation, make each selection perfectly arbitrary, and base our result on the possibility of the receiver's distinguishing the result of selecting any one symbol from that of selecting any other. By this means the psychological factors and their variations are eliminated and it becomes possible to set up a definite quantitative measure of information based on physical considerations alone.[9]

Thus Hartley distinguished between psychological considerations (involving meaning) and physical considerations (involving information, that is, the number of possible messages--whether meaningful or not). He used this definition of information to give the following logarithmic law for the transmission of information in discrete messages--such as in the case of telegraphy, which included Nyquist's earlier law:

$$H = K \log s^n ,$$

where H is the amount of information, K is a constant, s is the number of symbols, n is the number of symbols being chosen, and thus s^n is the number of possible symbolic sequences. Once the discrete case had been established, he showed how it could be

[9] Hartley, pp. 537-538.

modified to treat the case of continuous transmission of information, as in the case of telephone voice transmission.

Hartley next turned to questions of interference, and described how the distortion of a system limits the rate of selection at which distinctions between transmitted symbols may be distinguished with certainty. His special concern (from which a great deal of engineering research ensued) was with the interference caused by storage of energy through induction and capacitance, and through its subsequent release. He found that the total amount of information which could be transmitted over a steady state system of alternating currents which is limited in frequency to a given range is proportional to the product of the frequency-range on which it transmits and the time during which it is available for transmission.

Hartley had arrived at many of the most important ideas of the mathematical theory of communication: the difference between information and meaning, information as a physical quantity, the logarithmic rule for transmission of information, and the concept of noise as an impediment in the transmission of information. However, Hartley's aim was specifically to construct a theory capable of evaluating the information transmitted by any of the standard communication technologies. Starting with these ideas, Shannon developed a completely general theory of communication, not restricted to the study of technology designed specifically for communication. A colleague, Warren Weaver, described Shannon's theory clearly:

> The word communication will be used here in a very broad sense to include all of the procedures by which one mind may affect another. This, of course, involves not only written and oral speech, but also music, the pictorial arts, the theatre, the ballet, and in fact all human behavior. In some connections it may be desirable to use a still broader definition of communication, namely, one which would include the procedures by means of which one mechanism (say automatic equipment to track an airplane and to compute its probable future positions) affects another mechanism (say a guided missile chasing this airplane).[10]

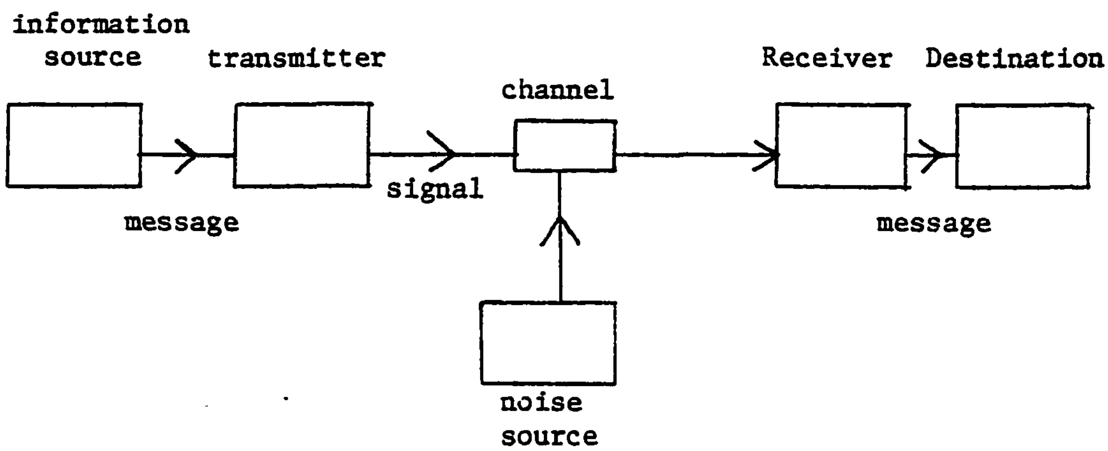
What began as a study at Bell Labs of transmission over telegraph lines ended as a general theory of communication applicable, according to Shannon,[11] to telegraph, telephone, radio, television, and computing machines--and, in fact, applicable to any system, physical or biological, in which information was being transferred or manipulated through time or space.

[10] "Introductory Note on the General Setting of the Analytical Communication Studies," p. 3, in Shannon and Weaver, The Mathematical Theory of Communication.

[11] See Shannon and Weaver, p. 35.

According to Shannon, a communication system consists of five components related to one another as illustrated in the following schematic diagram:[12]

Schematic diagram of a general communication system



These components are:

1. An information source which produces a message or sequence of messages to be communicated to the receiving terminal . . .
2. A transmitter which operates on the message in some way to produce a signal suitable for transmission over the channel. . . .
3. The channel is merely the medium used to transmit the signal from transmitter to receiver. . . .
4. The receiver ordinarily performs the inverse operation of that done by the transmitter, reconstructing the message from the signal.
5. The destination is the person (or thing) for whom the message is intended.

[12] See Shannon and Weaver, p. 34.

The importance of this characterization was that it applied equally well to a wide variety of communication problems, provided the five components were appropriately interpreted. For example, it applied equally well to conversations between humans, interactions between machines, and even to communication between parts of an organism. Through this generality, the similarities were established, for example, between the communication of the stomach with the brain and the target with the guided missile.

Hartley recognized that a distinction must be made between information and meaning. Shannon completed the distinction by giving the first precise definition of information capable of scientific usage. A colleague provided a good account of this definition:[13]

> The word information, in this theory, is used in a special sense that must not be confused with its ordinary usage. In particular, information must not be confused with meaning.
>
> In fact, two messages, one of which is heavily loaded with meaning and the other of which is pure nonsense, can be exactly equivalent, from the present viewpoint, as regards information. . . .
>
> To be sure, this word information in communication theory relates not so much to what you do say, as to what you could say. That is, information is a measure of one's freedom of choice when one selects a message. If one is confronted with a very elementary situation where he has to choose one of two alternative messages, then it is arbitrarily said that the information, associated with this situation, is unity. Note that it is misleading (although often convenient) to say that one

[13] Shannon and Weaver, pp. 8-9.

> or the other message conveys unit information. The concept of information applies not to the individual messages (as the concept of meaning would), but rather to the situation as a whole, the unit information indicating that in this situation one has an amount of freedom of choice, in selecting a message, which is convenient to regard as a standard or unit amount.

More tersely, Shannon stated, "thus in information theory, information is thought of as a choice of one message from a set of possible messages."[14]

Shannon recognized that information in his sense could be measured by any monotonic function, including the logarithmic funcitions, whenever the number of possible messages is finite. He chose the logarithmic function, for the same reason Hartley did, that it accorded well with our intuition of what the appropriate measure should be. For example, we intuitively feel that two punched cards should convey twice the information of one punched card. If one card can carry n symbols, two will carry n^2 combinations. $\log n^2 = 2 \log n$ gives twice the information ($\log n$) carried on one card. Shannon chose logarithm base 2 as the unit for measuring information since it designates one unit of information to a switch with two positions ($\log_2 2$(positions) $= 1$). Then N two position switches could store $\log_2 2^N = N$ binary

[14] Shannon, "Information Theory," Encylcopaedia Britannica, 1965.

digits of information.[15] If there were N equiprobable choices, then the amount of information would be given by $\log_2 N$. Shannon generalized this equation to the non-equiprobable situation, in which case the amount of information H would be given by

$$H = -(p_1 \log_2 p_1 + p_2 + \ldots + p_n \log_2 p_n) ,$$

where the choices have probabilities $p_1 , \ldots, p_n$.

Shannon recognized that this formulation of information was closely related to the concept of entropy, since, as mentioned above, the concept of information measured the orderliness of the communication channel.

> Quantities of the form $H = -\Sigma p_i \log p_i$ (the constant K merely amounts to a choice of a unit of measure) play a central role in information as measures of information, choice and uncertainty. The form of H will be recognized that of entropy as defined in certain formulations of statistical mechanics where p_i is the probability of a system being in cell i of its phase space. H is then, for example, the H in Boltzmann's famous H theorem.[16]

In fact, the concept of entropy had a long history in physics and, during the twentieth century, had already become closely associated with the amount of information in a physical system.

[15] "Binary digits" was shortened to "bits" by John Tukey, a Princeton professor who also worked at Bell Laboratories. The introduction of a new term, such as "bits," is a good indication of the introduction of a new concept.

[16] Shannon refers the reader to R. C. Tobman, Principles of Statistical Mechanics, Oxford, 1938.

Weaver carefully credited these roots of Shannon's work:[17]

> Dr. Shannon's work roots back, as von Neumann has pointed out, to Boltzmann's observation, in some of his work on statistical physics (1894), that entropy is related to "missing information," inasmuch as it is related to the number of alternatives which remain possible to a physical system after all the macroscopically observable information concerning it has been recorded. L. Szilard (Zsch. f. Phys. Vol. 53, 1925) extended this idea to a general discussion of information in physics, and von Neumann (Math. Foundation of Quantum Mechanics, Berlin, 1932, Chap. V) treated information in quantum mechanics and particle physics.

This close relation of information to entropy is not surprising, for information is related to the amount of freedom of choice one has in constructing messages. The fact that there is such a tie between thermodynamics, statistical mechanics, and communications theory suggests that communication theory involves a basic and important concept of the physical universe and is not a frivolous scientific product of modern communications technology.

Shannon used his theory to prove a number of theoretical results about communication systems and to demonstrate applications to the communications industry. He provided a number of results concerning the theoretical limits to practical communications problems. His communication theory had enormous theoretical significance. It isolated a new concept, information, which could be identified in a wide variety of physical settings. It

[17] Shannon and Weaver, fn., p. 3.

provided a mathematical handle to the theoretical problems of information transmission and processing. Shannon and Weaver continued the theoretical study of this subject. Meanwhile, this theory provided the basis for interdisciplinary information studies carried out by many others on such diverse systems as electronic computing machines, physical feedback systems, and biological feedback systems.

Norbert Wiener and Cybernetics

While Shannon concentrated mainly on communications engineering applications of information theory, Norbert Wiener concentrated on its application to control problems which had surfaced during the war effort and to complicated biological phenomena. Wiener's individual scientific projects were intended as illustrations of the power of the interdisciplinary approach of the new science of cybernetics. This new science he created as the result of his recognition that a similar approach was required to solve several diverse problems he had worked on during the war.

Wiener explicitly described the importance of the war-related work to his later work on cybernetics. Speaking of the conviction he shared with the physiologist, Arturo Rosenblueth, "that the most fruitful areas for the growth of the sciences were those which had been neglected as a no-man's land between the various established fields,"[18] he wrote:[19]

> We had agreed on these matters long before we had chosen the field of our joint investigations and our respective parts in them. The deciding factor in this new step was the war. I had known for a long time that if a national emergency should come, my function in it would be determined largely by two things: my close contact with the program of computing machines developed by Dr. Vannever Bush, and my own joint work with Dr. Yuk Wing Lee on the design of electrical networks. In fact, both proved important.

In 1940 Wiener began work on the development of computing machinery for the solution of partial differential equations. One outcome of that project was a proposal by Wiener, supposedly suggested to Vannever Bush,[20] for features to be incorporated into future computing machines. These features included: numerical rather than analog central adding and multiplying equipment, electronic tubes rather than gears or mechanical relays for switching, base of two rather than base of ten, completely built-in logical decisions with no human intervention necessary after introduction of the data, and an included memory with rapid capability of storage, recall, and erasure. The importance of these suggestions was that "they are all ideas which are of

[18] Wiener, <u>Cybernetics</u>, p. 8.

[19] Wiener, p. 9.

[20] Wiener, pp. 9-10.

interest in connection with the study of the nervous system."[21] This was the first attempt to explicitly compare features of the electronic computer and the human brain and was an illustration of the similarity of structure in diverse settings which Wiener emphasized in his cybernetics.

Another war-related program, undoubtedly the most important to Wiener's formulation of cybernetics, involved the development of fire-control apparatus for anti-aircraft artillery. This problem was of great importance at the beginning of the war due to German prowess in aviation and the defensive position of England. Because of the appreciable velocity of the new German aircraft, classical methods for the direction of fire were obsolete. Wiener found that any effective control device for anti-aircraft equipment must incorporate a feed-back system which directs future firings on the basis of the success of previous firings. Thus Wiener and Julian Bieglow worked on the theory of prediction (in this case of the flight of the aircraft) and on how to effectively apply this research to the anti-aircraft problem at hand.

> It will be seen that for the second time I had become engaged in the study of a mechanico-electrical system which was designed to usurp a specifically human

[21] Wiener, p. 11.

> function--in the first case, the execution of a complicated pattern of computation; and in the second, the forecasting of the future.[22]

Bigelow and Wiener recognized the importance of this concept of feed-back in a number of different mechanico-electrical and biological problems. For example, the movement of the tiller to regulate the direction of a ship was shown to involve the same feed-back process used in hand-eye coordinations necessary, say, to pick up a pencil.

Wiener realized that the mathematics of feed-back control was closely associated with parts of statistics, statistical mechanics, and information theory.

> On the communication engineering plane, it had already become clear to Mr. Bigelow and myself that the problems of control engineering and of communication engineering were inseparable, and that they centered not around the technique of electrical engineering but around the much more fundamental notion of the message, whether this should be transmitted by electrical, mechanical, or nervous means. The message is a discrete or continuous sequence of measurable events distributed in time--precisely what is called a time-series by the statisticians.[23]

The feed-back problems often reduced to partial differential equations relating to the stability of the system. The third war-related project, the work with Lee on wave filters, reinforced the close tie to information theory, for the purpose of their

[22] Wiener, p. 13.

[23] Wiener, p. 16

work was to remove extraneous background noise from electrical networks. The most important aspect of the work was the underlying mathematical theory applying to all of these diverse engineering problems. Like Shannon, Wiener was moving from the art of Engineering to the precision of science. Using the statistics of time-series, Wiener was able to show that the problem of prediction could be solved by the established mathematical technique of minimization.

> Minimization problems of this type belong to a recognized branch of mathematics, the calculus of variations, and this branch has a recognized technique. With the aid of this technique, we were able to obtain an explicit best solution of the problem of predicting the future of a time series, given its statistical nature; and even further, to achieve a physical realization of this solution by a constructible apparatus.
>
> Once we had done this, at least one problem of engineering design took on a completely new aspect. In general, engineering design has been held to be an art rather than a science. By reducing a problem of this sort to a minimization principle, we had established the subject on a far more scientific basis. It occurred to us that this was not an isolated case, but that there was a whole region of engineering work in which similar design problems could be solved by the methods of the calculus of variations.[24]

Due to the recurrence of similar problems of control and communication in widely diverse fields of engineering and to the availability of a mathematical theory by which to organize these problems, Wiener decided on the creation of a new interdisciplinary science which he called cybernetics. "We have decided to

[24] Wiener, p. 17.

call the entire field of control and communication theory, whether in the machine or in the animal, by the name of Cybernetics . . ."[25]

Actually, long before Wiener's formulation of the science of cybernetics in 1947, there were results which could be included under cybernetics. The word "cybernetics" derived from the Greek word "kybernetes" which meant steersman. This latter term was used by Plato to refer to careful and prudent public governance.[26] "Kybernetes" in Latin was "gubernator"--from which our word "governor" derived. Both as a steersman of public policy and as a self-regulation mechanism on a steam engine, the connotations are faithful to the word's ancient roots.[27] In fact, the governor on a steam engine is a feed-back mechanism which increases or decreases the speed of the engine depending on the present speed of the engine. James Clerk Maxwell published a paper[28] in 1868 which gave a mathematical characterization of governors. Similar feed-back mechanisms were discussed by the

[25] Wiener, p. 19.

[26] See Plato, Republic, I, 346B.

[27] For a history of the subject, see Otto Mayr, The Origins of Feedback Control.

[28] See J. C. Maxwell, "On Governors," Proceedings of the Royal Society, London, 1868, 270-283.

physiologist Claude Bernard around the turn of the century in his discussion of homeostasis, the means by which an organism regulates its internal equilibrium.[29]

Although Wiener only arrived at the name "cybernetics" in 1947, as early as 1942 there were interdisciplinary meetings to discuss the problems of the new science. The first meeting was held in New York in 1942 under the auspices of the Josiah Macy Foundation. The meeting was devoted to problems of "central inhibition in the nervous system." Bigelow, Rosenblueth, and Wiener read a paper[30] which used cybernetic principles to examine the functioning of the mind. Von Neumann and Wiener called an interdisciplinary meeting at Princeton in 1943-44 for engineers, physiologists, and mathematicians to discuss cybernetics and computing machinery design. As Wiener assessed the situation:

> At the end of the meeting, it had become clear to all that there was a substantial common basis of ideas between the workers in the different fields, that people in each group could already use notions which had been better developed by the others, and that some attempt should be made to achieve a common vocabulary.[31]

In fact, from discussions with computer engineers up and down the

[29] For a discussion of Bernard's work, see W. Cannon, Wisdom of the Body.

[30] Published as "Behavior, Purpose, and Teleology," Philosophy of Science, 10 (1943), 18-24.

[31] Wiener, Cybernetics, p. 23.

east coast, Wiener observed, "everywhere we met with a sympathetic hearing, and the vocabulary of the engineers soon became contaminated with the terms of the neurophysiologist and the psychologist."[32] In 1946 McCulloch arranged for a series of meetings to be held in New York on the subject of feed-back--again under the auspices of the Josiah Macy Foundation. Included at a number of these meetings from 1942 on were the mathematicians Wiener, von Neumann, and Pitts, the physiologists McCulloch, Lorente de No, and Rosenblueth, and engineers, such as H. H. Goldstine (who worked on ENIAC and EDVAC). Thus, there was widespread interaction in the United States of the participants in the new information sciences. A visit to England and France gave Wiener a chance to exchange information on cybernetics and artificial intelligence with Turing, then at the National Physical Laboratory at Teddington, and to exchange mathematical results on the relation of statistics and communication engineering with the French mathematicians at a meeting in Nancy. Thus, there was also international exchange of information on this new information science.

Throughout the development of Wiener's interest in cybernetics, there was a strong bias towards biological as well as

[32] Wiener, p. 23.

electro-mechanical applications of the new discipline. This interest goes back to the 1930's when Walter Cannon led informal monthly discussions on scientific method with a small group from Harvard Medical School. A few members of the M.I.T. faculty, including Wiener, began to attend these meetings. It was here that Wiener met Arturo Rosenblueth, with whom he was to collaborate on biocybernetics for the rest of his career.

Bigelow and Wiener, as a result of their work on anti-aircraft artillery, realized that feed-back is an important factor in voluntary activity. As illustration, Wiener described the process of picking up a pencil: We do not will certain muscles to take certain actions; rather, we will to pick the pencil up.

> Once we have determined on this, our motion proceeds in such a way that we may say roughly that the amount by which the pencil is not yet picked up is decreased at each stage. This part of the action is not in full consciousness.
>
> To perform an action in such a manner, there must be a report to the nervous system, conscious or unconscious, of the amount by which we have failed to pick the pencil up at each instant.[33]

Wiener and Bigelow recognized the importance of feed-back in this particular situation. As evidence, they pointed to the effect of pathological conditions such as ataxia, where the feed-back system is deficient, or purpose tremor, where the feed-back system is

[33] Wiener, p. 14.

overactive, in the inability to carry out such activities as picking up a pencil. In fact, they felt that the cybernetic approach could provide a valuable new view to the understanding of neurophysiology.

> We thus found a most significant confirmation of our hypothesis concerning the nature of at least some voluntary activity. It will be noted that our point of view considerably transcended that current among neurophysiologists. The central nervous system no longer appears as a self-contained organ, receiving inputs from the senses and discharging into the muscles. On the contrary, some of its most characteristic activities are explicable only as circular processes, emerging from the nervous system into the muscles, and re-entering the nervous system through the sense organs, whether they be proprioceptors or organs of the special senses. This seemed to us to mark a new step in the study of that part of neurophysiology which concerns not solely the elementary processes of nerves and synapses but the performance of the nervous system as an integrated whole.[34]

The revelation that cybernetics provided a new approach to neurophysiology resulted in a joint paper by Rosenblueth, Wiener, and Bigelow outlining their insights. As the title indicates,[35] they gave an outline of behavior, purpose, and teleology from a cybernetic approach. They argued that "teleological behavior thus becomes synonymous with behavior controlled by negative feed-back, and gains therefore in precision by a sufficiently restricted connotation."[36] They also argued that the same broad

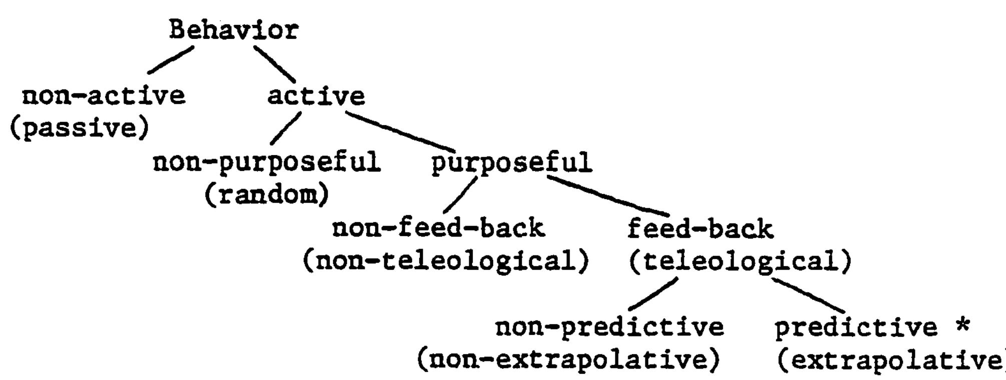
[34] Wiener, p. 15.

[35] "Behavior, Purpose, and Teleology."

[36] Rosenblueth, Wiener, and Bigelow, p. 24.

classifications of behavior (illustrated below) hold for machines as hold for living organisms. The differences, they maintained, are functional differences: collcids versus metals, large versus small differences in energy potentials, temporal versus spatial multiplication of effects, etc.[37] They presented the following chart as illustration of the various classifications of behavior.

<u>The Rosenblueth, Wiener, and Bigelow Classification of Behavior of Machines and of Living Organisms:</u>[38]



* order of prediction (depending on number of parameters)

[37] See Ibid., pp. 22-23 for details.

[38] As given in Ibid., p. 21, with exception that order of prediction is explained here.

For the most part, however, Wiener's work in biocybernetics was less philosophical and more physiological than the joint paper with Rosenblueth and Bigelow would indicate. Typical was a joint project between Rosenblueth and Wiener on the muscle actions of the cat.[39] In this project they used the cybernetic methods of McCall on servomechanism[40] to analyze the system in the way one would study an electrical or mechanical system from data provided with an actual physiological experiment on the cat. For the remainder of his career, Wiener cooperated with Rosenblueth on specific physiological projects showing the utility of the cybernetic approach to the understanding of physiological processes of biological organisms.

Warren McCulloch, Walter Pitts, and the Development of Mathematical Models of the Nervous System

Another application of the new information science was to the study of nervous systems and, in particular, to the study of the human brain. This resulted in mathematical models which were partly physiological and partly philosophical. The most famous

[39] See Wiener, Cybernetics, pp. 28-30.

[40] M. McColl, Servomechanisms, 1945.

application of information science in this area was a joint paper by Warren McCulloch and Walter Pitts in which they provided a mathematical model of the neural networks of the brain based on Rudolf Carnap's logical calculus and on Turing's work on theoretican machines.

The application of the information sciences to psychology fit well with movement within the discipline of psychology itself. The rise of physiological psychology, the development of functionalism, the growth of behaviorism, and the infusion of materialism in the biological and psychological sciences all contributed to the study of mathematical models of the functioning of brains in a very general sense.

Physiological psychology was important to information science because it contributed the idea that one can understand the brain by examining its material functioning; that this functioning is capable of scientific study; and, consequently, that it is capable of mathematical examination. Beginning with the work of Helmholtz, Ribot, and James at the end of the nineteenth century,[41] physiological psychology became equated with the study of the physiological underpinnings of behavior and experience. The subject grew to be heavily dependent on physiological research

[41] See Gardner Murphy, Historical Introduction to Modern Psychology, New York, 1949, for details.

concerning the central nervous system. Of special importance was the work of Waldeyer and Sherrington.[42] Waldeyer's "neurone theory," which argued for the independence of the nerve cells and the importance of the synapses, was quickly accepted by psychologists and was the basis for most later physiological study of the brain. Sherrington's work on reflex arc was the most important research in convincing psychologists they should consider a neurophysiological approach. McCulloch explicitly pointed to Sherrington's work as a precursor to his own research. Their work shared another similarity--its idealized nature. Sherrington realized that his model of simple reflex was not physiologically precise, but only a "convenient abstraction." McCulloch and Pitts made a similar observation about their neuron nets.

Research continued in physiological psychology throughout the twentieth century. However, its importance increased dramatically in the 1930's due to two developments: the implementation of electroencephelegraphy enabled researchers to make precise measurements of the electrical activity in the brain; and the growth of mathematical biology, especially Nicholas Rashevsky's Chicago school, which produced Pitts, contributed a precise, mathematical theory of the functioning of the brain which could

[42] For example, see C. S. Sherrington, The Integrative Action of the Nervous System, New Haven, CT, 1906.

be tested experimentally. Thus, the concentration on the material properties of the brain, the emphasis on its functioning rather than on its states of consciousness, and the mathematical approach of Rashevsky all set the stage for a mathematical theory of the functioning brain as an information processor.

An attitude held generally by the physiological psychologists, but more generally held among psychologists as well, was the functionalist position. James argued in Principles of Psychology that mind should be conceived dynamically as opposed to structurally. At the end of the nineteenth century there was a strong belief that psychology should concentrate on mental activity rather than on states of experience.[43] E. B. Holt took a radical, and almost cybernetic, feedback position[44] towards psychology when he argued that consciousness is merely a servomotor adjustment to the object under consideration. As one historian of psychology assessed the importance of functionalism:[45]

> Functionalism did not long maintain itself as a school; but much of the emphasis lived on in behaviorism . . . and in the increasing tendency to ask less about consciousness, more about activity . . .

[43] See William James, Principles of Psychology, New York, 1890.

[44] See E. B. Holt, The Freudian Wish, New York, 1915.

[45] Gardner Murphy, p. 223.

The importance of functionalism to information science is clear: it concentrated on the functional operation of the brain; and it conceived of the brain as a processor (or information)--as a doer as well as as a reflector.

Behaviorist psychology concentrated on behavior rather than consciousness. Thus it helped to break down the distinction between the mental behavior of humans and machines. This assisted the acceptance of a unified theory of information processers--whether they be men or machines. American behaviorism was a revolt against the old-style, introspective psychology of Wundt and Titchener. In the attempt to make psychology scientific, there was a movement toward materialism at the end of the nineteenth century. Behavior was observable and therefore capable of scientific study. Watson, the leader of American behaviorism, thus concentrated on "scientific" concepts such as effector, receptor, and learning as opposed to the old concepts of sensation, feeling, and image.[46] In fact, Watson also conceived of mental functions as a type of internal behavior. One sees much the same attitude taken by Turing in his conception of thinking machines--whether human or mechanical. In fact, although Watson's behaviorism did not convince the majority of American

[46] See J. B. Watson, Psychology from the Standpoint of a Behaviorist, New York, 1919.

psychologists, there was a group of dedicated behaviorists who conducted experiments using the condition-response method. The most important of these in the 1930's was Lashley, who had a significant influence on McCulloch and Pitts' work. Lashley's viewpoint is indicated by the following quotation:[47]

> To me the essence of behaviorism is the belief that the study of man will reveal nothing except what is adequately describable in the concepts of mechanics and chemistry, and this far outweighs the question of the method by which the study is conducted.

Warren McCulloch was trained within this psychological tradition of experimental epistemology. As an undergraduate at Haverford and Yale, he majored in philosophy and psychology. He then went to Columbia, where he received a master's degree in psychology for work in experimental aesthetics. After this, he entered the Columbia Medical School, where he studied the physiology of the nervous system.

> In 1928 I was in neurology at Bellevue Hospital and in 1930 at Rockland State Hospital for the Insane, but my purpose never changed |to manufacture a logic of transitive verbs|. It was then that I encountered Eilhard von Doramus, the great philosophic student of psychiatry, from whom I learned to understand the logical difficulties of true cases of schizophrenia and the development of psychopathia--not merely clinically, as he had learned them of Berger, Birnbaum, Bumke, Hoche, Westphal, Kahn, and others--but as he understood them from his friendship with Bertrand Russell, Heidegger, Whitehead, and Northrop--under the last of whom he wrote his great unpublished thesis, "The Logical Structure

[47] K. S. Lashley, "The Behavioristic Interpretation of Consciousness," Psychol. Rev., 30(1923), p. 244.

> of Mind: An Inquiry into the Foundations of Psychology and Psychiatry." It is to him and to our mutual friend, Charles Holden Prescott, that I am chiefly indebted for my understanding of paranoia vera and of the possibility of making the scientific method applicable to systems of many degrees of freedom.[48]

McCulloch left Rockland to return to Yale. There he studied experimental epistemology with the psychiatrist Dusser de Barenne. Upon de Barenne's death, he moved to the University of Illinois as a Professor of Psychiatry, where he continued his work, at this point collaborating with Pitts, on experimental epistemology. His career concluded at the Research Laboratory of Electronics at M.I.T., where he went in 1952 to work with Pitts, Wiener, and others on electronic circuit theory of the brain.

Walter Pitts came from a more mathematical background than McCulloch; nevertheless, it was tied to this work in experimental epistemology. Pitts was trained in mathematical logic by Rudolf Carnap at the University of Chicago. While there, he worked with Professor Nicholas Rashevsky and his school of biophysicists. Also while at Chicago, Pitts met the older McCulloch, with whom he began collaboration on a study of the mathematical structure of systems built out of nerve nets. Through a mutual friend, Dr. J. Lettvin of Boston City Hospital, Pitts was introduced to Wiener and Rosenblueth. Later the same year, 1943, Pitts accepted

[48] Warren McCulloch, "What is a Number that a Man May Know It?" as reprinted in Embodiments of Mind, pp. 2-3.

a permanent position at M.I.T. to work with Wiener and learn from him the cybernetic approach.

> At that time Mr. Pitts was already thoroughly acquainted with mathematical logic and neurophysiology, but had not had the chance to make very many engineering contacts. In particular, he was not acquainted with Dr. Shannon's work, and he had not had much experience of the possibilities of electronics. He was very very much interested when I showed him examples of modern vacuum tubes and explained to him that these were ideal means for realizing in the metal the equivalents of his neuronic circuits and systems. From that time, it became clear to us that the ultra-rapid computing machine, depending as it does on consecutive switching devices, must represent almost an ideal model of the problems arising in the nervous system.[49]

Pitts' work at the Research Laboratory of Electronics involved the relation between electronic computers and the human nervous system. During this time his collaboration with McCulloch continued, and eventually McCulloch joined him at M.I.T. in 1952.

Early in his career, between 1919 and 1923, McCulloch worked on a problem from philosophical logic, that of creating a formal logic to explain the usage of transitive verbs. While working on this project, he became interested in another problem in philosophical logic, the logic of relations. Recent mathematical developments had stirred this interest. As McCulloch recalled:[50]

> The forms of the syllogism and the logic of classes were taught, and we shall use some of their devices, but there

[49] Wiener, Cybernetics, p. 22.

[50] McCulloch, "What is a Number," in Embodiments, pp. 7-8.

> was a general recognition of their inadequacy to the problems in hand. . . . It was |Charles| Peirce who broke the ice with his logic of relatives, from which springs the pitiful beginnings of our logic of relations of two and more than two arguments. So completely had the traditional Aristotelean logic been lost that Peirce remarks that when he wrote the Century Dictionary he was so confused concerning abduction or apagoge, and induction that he wrote nonsense. . . . Frege, Peano, Whitehead , Russell, Wittgenstein, followed by a host of lesser lights, but sparked by many a strange character like Schroeder, Sheffer, Gödel, and company, gave us a working logic of propositions. By the time I had sunk my teeth into these questions, the Polish school was well on its way to glory. In 1923 I gave up the attempt to write a logic of transitive verbs and began to see what I could do with the logic of propositions.

It is clear that what McCulloch had in mind was a psychological, rather than a philosophical, theory for the logic of relations. Whereas a philosopher would have attempted to construct a formal system which mirrored typical usage of the logic of relations, McCulloch intended to develop a theory which explained the psychological underpinnings, not just the formal structure.

> My object, as a psychologist, was to invent a kind of least psychic event, or "psychon," that would have the following properties: First, it was to be so simple an event that it either happened or else it did not happen. Second, it was to happen only if its bound cause had happened--shades of Duns Scotus!--that is, it was to imply its temporal antecedent. Third, it was to propose this to subsequent psychons. Fourth, these were to be compounded to produce the equivalents of more complicated propositions concerning their antecedents.
>
> In 1929 it dawned on me that these events might be regarded as the all-or-none impulses of neurons, combined by convergence upon the next neuron to yield complexes of propositional events. During the nineteen-thirties, first under influences from F. H. Pike, C. H. Prescott, and Eilhard von Doramus, and later, Northrop, Dusser de Barenne, and a host of my friends in neurophysiology,

> I began to try to formulate a proper calculus for these events by subscripting symbols for propositions in some sort of calculus of propositions (connected by implications) with the time of occurrence of the impulse in each neuron.[51]

Technical problems stood in the way of his psychology of propositions.[52] It was then that McCulloch met Pitts, who was able to provide the requisite mathematical theory to resolve these problems. The result was their famous joint paper, "A Logical Calculus of the Ideas Immanent in Nervous Activity."[53] The paper was published in Rashevsky's journal, the Bulletin of Mathematical Biophysics, where it went unnoticed by the biology and psychology communities until von Neumann popularized it.

In this paper, using as axioms the rules McCulloch wished to be true for his psychons as well as Carnap's logical calculus and Russell and Whitehead's notation, McCulloch and Pitts provided a logical model of neuron nets which showed their functional similarity to Turing's computing machines.

> What Pitts and I had shown was that neurons that could be excited or inhibited, given a proper net, could extract any configuration of signals in its input. Because the form of the entire argument was strictly logical, and because Gödel had arithmetized logic, we

[51] Ibid., pp. 8-9.

[52] See Ibid., p. 9 for a listing of these problems.

[53] Bulletin of Mathematical Biophysics, 5 (1943), pp. 115-133.

> had proved, in substance, the equivalence of all general Turing machines--man-made or begotten.[54]

As von Neumann emphasized in his <u>General and Logical Theory of Automata</u>, what McCulloch and Pitts did, in essence, was to show how any functioning of the brain which could be described clearly and unambiguously in a finite number of words could be expressed as one of their formal neuron nets. The close relationship between Turing machines and neuron nets was intentional on the part of the authors, and it was soon apparent that neuron nets, when supplied with the analog of an infinite tape, were equivalent to Turing machines.[55] With the Turing machines providing an abstract characterization of all thinking in the machine world and McCulloch and Pitts' neuron nets providing an abstract characterization of thinking in the biological world, the equivalence result provided a unified theory of thinking for both the physical and the biological worlds.

Their paper not only showed the similarity in the abstract functioning between the human brain and computing devices; it also provided a way of conceptualizing the brain as a machine in a more precise way than had been available before. Thus there was a means

[54] <u>Ibid</u>., pp. 9-10.

[55] See the discussion in Chapter Seven. The class of Neuron nets is not as wide as the class of Turing machines. See <u>Embodiments of Mind</u>, p. xviii, for details.

for further study of the brain, starting from a precise mathematical formulation.

> But we had done more than this, thanks to Pitts' modulo mathematics. In looking into circuits composed of closed paths of neurons wherein signals could reverberate, we had set up a theory of memory--to which every other form of memory is but a surrogate requiring reactivation of a trace.[56]

In a series of papers,[57] McCulloch and Pitts carried out the mathematical details of this theory of the mind, providing, for example, a model of the way in which humans know universal ("for all") statements.

The precision of their mathematical theory enabled a great deal of additional speculation about the functioning of the mind. This was done at the expense of a detailed theory of the biological structure and functioning of the individual nerve cells. Similar to Sherrington's model of the simple reflex, which he termed but a "convenient abstraction," McCulloch and Pitts' neurons were idealized neurons. One knew what the input and output would be; but they were "black boxes," closed to inspection as far as their internal structure and functioning were concerned. Practicing

[56] McCulloch, Embodiments, p. 10.

[57] See "A Heterarchy of Values Determined by the Topology of Nerve Nets" (1945), "Finality and Form in Nervous Activity" (1946), and "How We Know Universals" (1947). All are included in Embodiments of Mind.

physiologists objected[58] that not only was this model of neurons incomplete, it was inconsistent with facts known about actual neurons. They argued that the simplicity of the idealized neuron was so misleading as to vitiate their results. Von Neumann, who was the popularizer of McCulloch and Pitts' theory to the biologists, argued that the simple, idealized nature of the model was necessary to understand the basic, logical nature of the functioning of these neurons and that once this nature was known, it would be easier for the biologists to account for the secondary effects due to the physiological details of the neurons. This disagreement is discussed in more detail in Chapter Seven.

McCulloch and Pitts worked on a number of specific projects using their mathematical theory to analyze aspects of the functioning of man's nervous system. An article of 1950, entitled "Machines that Think and Want,"[59] by McCulloch provided a retrospective of the possible applications of their theory, emphasizing especially the application of cybernetic techniques to understanding the functioning of the central nervous system. Typical of McCulloch and Pitts' application of information to

[58] See von Neumann, General and Logical Theory of Automata. This is discussed in more detail in Chapter Seven.

[59] Included in Embodiments of Mind, pp. 307-318.

physiology was a joint project in 1947 on prosthetic devices to enable the blind to read the printed page by ear.[60] The problem actually resolved into a problem of pattern recognition in order to translate letters of various sizes into particular sounds. Using cybernetic techniques, they produced a theory relating the anatomy and physiology of the visual cortex, which exhibited a similarity between human vision and televideo. Consequently, they were able to arrive at a reasonably successful principle for a translating device.

Alan Turing, Automata Theory, and Artificial Intelligence

As was discussed in the third chapter, Turing's computer work was influenced by his war experiences at Bletchley Park. Although he had already considered building a computer before the war began, his intentions were solidified by the experience. More important, the war provided him with technical engineering experience useful in designing a physical machine such as ACE. Although ACE can be viewed as an attempt to see whether Turing's theoretical machines could be effected as actual, physical machines--and there is reason to believe that Turing viewed the project as such--his main contributions to theoretical computer science are found

[60] See Wiener, Cybernetics, pp. 31-32, for a description of this project, or P. de Latil, Thinking by Machine, pp. 12-13.

in his pre-war work on Turing machines and in his later programming work at the University of Manchester.

The work on Turing machines provided the basis for the modern theory of automata. Modern computer scientists still use Turing machines as a way of comparing the powers of computing automata. In his early work Turing described the basic functions and components any computing automaton must have, whether the automaton be electro-mechanical or biological. His Turing machines were designed to provide a direct formal analogue of the way in which the human computer functions; and, in creating these machines, Turing gave a precise, mathematical model of the way the mind functions when carrying out computations. In this model he provided a clear characterization of the processing of information. This was not lost to McCulloch and Pitts, who used Turing's machine characterization as the basis for their characterization of human neuron nets as information processors.

Turing's work at the University of Manchester was among the earliest of investigations of electronic computers as artificial intelligence. He believed that electronic machines were not only capable of doing numerical computations, but also could be built to be general-purpose information processers capable of carrying out any mental activity of which the human mind was capable. He explicitly attempted to break down the distinctions between human and machine intelligence and to provide one standard of intelligence, in terms of mental behavior, upon which both machines and biological organisms could be judged. In providing this standard, he considered the automata only on the basis of the information which was input and output. Thus, Turing, as well as Shannon and Wiener, was moving toward a unified theory of information and information processing which applied to both the machine world and the biological world. The details of this theory are discussed in the next chapter.

It is important to note that Turing, although isolated in England from the bulk of the theoretical work being accomplished in America, did have some contact with his American colleagues. He corresponded with von Neumann. During the war he made a secret trip to the United States--presumably related to his computer work, although the details are not available. He knew of McCulloch and Pitts' work using his Turing machines. Wiener visited him in 1947 to discuss the new science of cybernetics. There were other scientists in England, in particular W. R. Ashby in the field of homeostasis, working on similar problems in theoretical information science, of which Turing was aware. Thus Turing, although inclined to investigate problems of his own choosing and outside of well-established research areas, was cognizant of other work in theoretical information science, if not a participant in cooperative research himself.

John von Neumann and the General Theory of Automata

Von Neumann is an appropriate culminating figure for the early period in the information sciences because of his attempts to unify the work of his colleagues. As was discussed in previous chapters, in his early career von Neumann made significant contributions to the development of mathematical logic. While at the Institute for Advanced Study in Princeton he first engaged in the discussion of computing machines through his interactions with Turing. Again, it was because of war-related work that von Neumann required the additional computing power possible only from an electronic computing machine. Thus von Neumann became involved in the computer project at the University of Pennsylvania being carried out for Army Ordnance. The upshot was his central role in the logical design (using ideas from mathematical logic) for EDVAC and his leadership in the IAS computer project.

Von Neumann's war-related computer activities spurred his interest in further theoretical work in the information sciences. His main interest was in developing a general, logical theory of automata. The hope was that this general theory would unify the work of Turing on theoretical machines, of McCulloch and Pitts on neural networks, and of Shannon on communication theory. Whereas Wiener attempted to unify cybernetics around the idea of feedback and control problems, von Neumann hoped to unify these various results, in both the biological and mechanical realms, around the concept of an information processer--which he called an automaton. Generally, automata are devices which carry out actions through the aid of a hidden mechanism. However, von Neumann was primarily concerned only with those automata whose action involved the processing of information.

The task of constructing a general and logical theory of automata was too large a project for von Neumann to carry out in detail in the final few years of his career. All he could do was provide a programmatic framework for the further working out of the general theory and limit himself to developing a few, specific aspects. Three interrelated topics were of primary concern. His foremost interest was with complicated automata, such as the human nervous system or modern electronic computers, and the importance complexity played in information processing. Second, he was interested in another class of highly complex automata--those capable of self-reproduction. He desired a model which would account for both the universal Turing machine and the genetic passage of information. Toward this problem, he designed a number of self-reproducing automata, both mechanical and organic. Third, he was interested in the probabilistic nature of automata. To this end, he wrote a paper[61] giving a probabilistic logic which tried to examine the problem of reliability for a complicated automaton containing unreliable components. The details of these projects are discussed in Chapter Seven.

Von Neumann did not merely attempt to incorporate prior work of others into his general theory of automata. Rather, he was in close contact with other information scientists. He discussed computers and artificial intelligence with Turing when they were both at Princeton. He had an active correspondence with Wiener and Weaver. Von Neumann and Wiener were the principal organizers of the interdisciplinary Princeton meetings in 1943 on cybernetics and computing. Von Neumann was in regular attendance at the Macy Foundation meetings. In fact, it was a paper by Pitts[62] at one of these meetings on the probabilistic nature of neuron nets that interested von Neumann in the probabilistic issues of automata. In sum, von Neumann recognized the relation of his work to the broader developments in theoretical information science.

[61] See "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," Collected Works, V, 329-378.

[62] See the introduction to McCulloch, Embodiments of Mind, for a discussion of this point.

Chapter Six: Turing's Contributions to the Development of a Theory of Information Processing

Chapter Two demonstrated that Turing's work on computable numbers answered questions about the foundations of mathematics of interest to mathematical logicians. Chapter Three demonstrated how the same work on computable numbers was utilized in the development of computing machinery. This chapter will show how the work on computable numbers was the basis for Turing's theoretical work on information processing and the reasons why it was of interest to the psychologists and theoretical computer scientists.

Turing must be credited with three significant contributions to the general theory of information processing:

(1) His contributions to the development of the study of artificial intelligence, including his tenacious argument that machines can be built which think, his criterion for deciding whether a machine can be said to be thinking, and his efforts to actually construct thinking machines;

(2) the application of results from mathematical logic to the information sciences, especially to automata theory; and

(3) his general characterization of brains and computers as

thinking automata, based on his idealized Turing machines.

Turing was not responsible for development of an organized, detailed theory of automata. Von Neumann was. Turing never developed a coherent theory of information processing which subsumed the study of both artificial and natural organisms. This was due to Szilard, Shannon, Wiener, Ashby, and von Neumann. Turing's contributions lay at the pre-discipline level. Turing was important for the many profound and novel thoughts and devices he contributed, not for the organization of ideas into a coherent and developed theory. Turing is most important for basic ideas: the invention of the Turing machine, the earliest attempts to discuss computable functions in terms of machines that can carry out the computations, the application of mathematical logic to the theory of automata, recognizing the striking similarities between the functioning of the brain and the functioning of the electronic computer, determining the criteria for deciding when a computer could be said "to think," the plan for education of unorganized machines, and the early attempts to program electronic computers to carry out thinking activities such as chess playing. All of these contributions relate to the one question foremost in Turing's mind: Can machines think?

Throughout his career Turing believed that the computer could adequately model the thought processes of the human brain. The first evidence of this is in the very first paper he wrote on automata, the famous 1936 paper[1] on computable numbers. In designing his (Turing) machines, Turing required that they have the features and limitations a person computing would have, since the question he was investigating was the possibility of characterizing mathematically those numbers which are humanly computable.[2] In fact, he began his paper with a description of the way in which his machine imitates the features of the human brain:

> We may compare a man in the process of computing a real number to a machine which is only capable of a finite number of conditions $q_1, q_2, \ldots, q_N$, which will be called "m-configurations." The machine is supplied with a tape (the analogue of paper) running through it, and divided into sections (called "squares") each capable of bearing a "symbol." At any moment there is just one square, say the r-th, bearing the symbol (r) which is "in the machine." We may call this square the "scanned square." The symbol on the scanned square may be called "the scanned symbol." The "scanned symbol" is the only one of which the machine is, so to speak, "directly aware." However, by altering its m-configuration the machine can effectively remember some of the symbols which it has "seen" (scanned) previously.[3]

Turing described the internal states (m-configurations) and operations of writing and erasing symbols on the paper, and then

[1] Alan Turing, "On Computable Numbers," Proceedings of the London Mathematical Society, Series 2, 42 (1936-37), 230-265.

[2] See Chapter Two for details.

[3] "On Computable Numbers," p. 231.

concluded:

> Some of the symbols written down will form the sequence of figures which is the decimal of the real number which is being computed. The others are just rough notes to "assist the memory." It will only be these rough notes which will be liable to erasure.
>
> It is my contention that these operations include all those which are used in the computation of a number.[4]

In a later section of the paper,[5] Turing returned to the comparison of the human computer and his machines to show, by direct appeal to intuition, that the machine computable numbers are just those humanly computable. For example, he argued that the tape should not contain an infinite number of symbols, should not scan an infinite number of squares at one time, or contain an infinite number of internal configurations because then the symbols, concatenations of symbols, or internal configurations could vary by only an "arbitrarily small extent"--which would contradict the condition of "immediate recognisability" necessary for the human computation of a number.[6] Turing concluded:

> We may now construct a machine to do the work of this [human] computer. To each state of mind of the [human] computer corresponds an "m-configuration" of the machine.

---

[4] *Ibid.*, p. 232. Again, my emphasis.

[5] Section 9.

[6] It is interesting that Wittgenstein, who frequented Cambridge, had a similar, though more radical, condition of "immediate recognisability" at about this same time. See his *Remarks on the Foundations of Mathematics*.

> The machine scans B squares corresponding to the B squares observed by the |human| computer. . . . The move which is done, and the succeeding configuration, are determined by the scanned symbol and the m-configuration. . . . A computing machine can be constructed to compute . . . the sequence computed by the |human| computer.[7]

These convictictions led Turing to argue repeatedly with von Neumann during the former's matriculation at Princeton and, according to Rosser,[8] left both men determined to build computers to test Turing's hypothesis. However, as was described in earlier chapters, the war intervened and these plans had to be set aside, although both men eventually became involved with calculating machinery during the war.

As soon as the war was over Turing was intent upon finding a position in which he could design and build a computer. As discussed in Chapter Three, he settled upon a position at the National Physical Laboratory. During his time at NPL his interest in constructing a machine capable of modeling the human thought processes bacame more intense. This interest is recorded in the paper, "Intelligent Machinery," which he wrote during a year at Cambridge while on a sabbatical from NPL.

[7] Turing, "On Computable Numbers," p. 232.

[8] In conversation with Rosser at the University of Wisconsin in April, 1979.

An abstract at the beginning of "Intelligent Machinery" summarizes Turing's intentions in the paper:

> The possible ways in which machinery might be made to show intelligent behaviour are discussed. The analogy with the human brain is used as a guiding principle. It is pointed out that the potentialities of the human intelligence can only be realized if suitable education is provided. The investigation mainly centres round an analogous teaching process applied to machines. The idea of an unorganized machine is defined, and it is suggested that the infant human cortex is of this nature. Simple examples of such machines are given, and their education by means of rewards and punishments is discussed. In one case the education process is carried through until the organization is similar to that on an ACE.[9]

The paper is a marvelous account of Turing's foresightedness and creative analysis. He began by attempting to dismiss common objections to machinery showing intelligent behavior,[10] thereby arguing the possibility that such machinery can exist. He then turned to categorizing the various types and attributes of such machinery. All such machinery, Turing argued, is either discrete or continuous, controlling (only dealing with information) or active (producing some definite physical effect). Thus every machine fits in to one of four categories. For example, bulldozers are continuous active, differential analyzers

[9] Turing, "Intelligent Machinery," p. 3.

[10] As Turing elaborated on these arguments in "Computing Machinery and Intelligence," a discussion of these arguments will be held in abeyance until that paper is discussed below.

are continuous controlling, and computers, like ACE or ENIAC, are discrete controlling. Turing then stated, without argument, that although human brains (as machines) are probably continuous controlling,

> We shall mainly be concerned with discrete controlling machinery. As we have mentioned, brains very nearly fall into this class, and there seems every reason to believe[11] that they could have been made to fall genuinely into it without any change in their essential properties. However, the property of being 'discrete' is only an advantage for the theoretical investigator, and serves no evolutionary purpose, so we could not expect Nature to assist us by producing truly 'discrete' brains.[12]

This categorization allowed Turing to focus on discrete controlling machines as his model for the brain. He discussed the differences between logical computing machines (e.g., Turing machines) and practical computing machines (e.g., ACE), such as memory and computing bounds, in a knowledgable, matter-of-fact way which reflects his experience building computers at Bletchley Park and NPL. Typical of Turing's foresight, he even discussed the possibilities of random elements in computers--not to occur in practice for many years.

It was at this stage that Turing's thought takes an entirely novel direction. He pointed out that all of the machines discussed

[11] Which reasons he never mentions.

[12] Ibid., p. 6

so far have been organized for special purposes. He queried, "What happens when we make up a machine in a comparatively unsystematic way from some kind of standard components?"[13] He called these unorganized machines and created a new mathematical theory, similar to the modern theory of network flows, for analyzing them.[14] The aim of Turing's study was purely academic: what sorts of machines could be constructed which displayed some evidence of artificial intelligence? Turing had no intention of using these machines for any specific purposes.

Turing decided the best way to construct a machine displaying artificial intelligence was to build a machine which could learn, rather than construct a machine with mature reasoning ability. His plan was to create machines which he could organize to parallel the educational development of an infant into a thinking adult. He first noted that there was good reason to believe that thinking machinery could indeed be built because it was already possible (he claimed) to construct machinery to imitate any small part or function of man. However, Turing's aim was not to put together a series of machines (each of which carried out one

[13] Ibid., p. 9.

[14] This technique was taken in part from Goldstine and von Neumann's technique of flow diagramming. However, Turing's mathematical analysis goes beyond that of Goldstine and von Neumann in this regard.

limited human activity) which would, as a composite, carry out all of the human functions. Rather, he was most concerned to construct an analogue to the human nervous system. With the advanced state of electronics, Turing was confident of success.

> Here we are chiefly interested in the nervous system. We could produce fairly accurate electrical models to copy the behaviour of nerves, but there seems very little point in doing so. It would be rather like putting a lot of work into cars which walked on legs instead of continuing to use wheels. The electrical circuits which are used in electronic computing machinery seem to have the essential properties of nerves. They are able to transmit information from place to place, and also to store it. Certainly the nerve has many advantages. It is extremely compact, does not wear out (probably for hundreds of years if kept in a suitable medium!) and has a very low energy consumption. Against these advantages the electronic circuits have only one counter-attraction, that of speed. This advantage is, however, on such a scale that it may possibly outweigh the advantages of the nerve.[15]

The outcome of Turing's analysis was: (1) that electronics technology provides the possibilities for successfully constructing a thinking machine; (2) that the momentousness of the project and the size scale of the machine would make it impractical to construct a thinking machine by hooking together a series of machines, each able to imitate some small human feature, for instance by microphones or mechanical limbs; (3) that even if such

[15] Ibid., pp. 12-13.

a long and impractical project were completed, the machine would still lack certain human characteristics, such as interest in food, sex and sport; (4) that because of these difficulties and because the main concern was with constructing a thinking machine, the appropriate approach seemed to be to build a mind devoid of body, using electronics to model the nervous system; and (5) that useful projects to work on would include building a machine which would react the way the brain does to purely mental activities, including (Turing explicitly lists): (i) games like chess or bridge, (ii) learning of languages, (iii) translating of languages, (iv) cryptography, and (v) mathematics.

The problem remained, however, of the appropriate approach for constructing this electronic brain. One approach might be to analyze and categorize the various parts and functions of the brain and then to build machines to imitate each of these, in the end placing them all together in a composite thinking machine. However, this approach seemed liable to the same objections as the plan for using microphones for hearing, artificial limbs for moving, and such. The composite machine, difficult to build, would likely still be lacking certain characteristics of the human brain. Only good fortune would guarantee that, at the end of this long process of construction, a thinking machine would have been created with all of the required features, for there was no *a priori* way of determining an exclusive list of features of the human brain.

It was at this point that Turing's novel idea of unorganized machines came into play. Turing reasoned that a thinking machine should be given the essentially blank mind of an infant rather than all of the features common to the adult mind. The trick would then be to incorporate a mechanism by which the infant electronic brain could be trained in a way analogous to the education of children. The possibility of such an approach depended partly on Turing's analysis of the human cortex:

> We believe then that there are large parts of the brain, chiefly in the cortex, whose function is largely indeterminate. In the infant these parts do not have much effect: the effect they have is uncoordinated. In the adult they have great and purposive effect: the form of this effect depends on the training in childhood. A large remnant of the random behaviour of infancy remains in the adult.
>
> All of this suggests that the cortex of the infant is an unorganized machine, which can be organized by suitable interference training.[16]

Not only must one begin with an unorganized machine. One must also include some method for allowing the machine to be changed. Turing believed that humans learn from "interference" created by other humans. So he proposed that interference be the norm by which computers be educated. He realized, however, that only computers which had been allowed to continue operating

[16] Ibid., p. 16.

indefinitely without interference from outside had been considered. Thus Turing had to begin from scratch in constructing a theory of interference. He isolated two types of interference: "screwdriver interference," by which parts of the machine are removed or replaced, and "paper interference," consisting of "mere communication of information to the machine, which alters its behaviour."[17] He pointed out that screwdriver interference really produces a new machine; so he restricted himself to paper interference. Once this was decided he did additional network flow analysis to examine the possibilities of organizing machines through interference.

The question of mechanism for organization of the machine still remained. Turing again looked to the education of children for a clue.

> The organization of a machine into a universal machine would be most impressive if the arrangements or interference involve very few inputs. The training of the human child depends largely on a system of rewards and punishments, and this suggests that it ought to be possible to carry through the organizing with only two interfering inputs, one for 'pleasure' or 'reward' (R) and the other for 'pain' or 'punishment' (P). One can devise a large number of such 'pleasure-pain' systems. I will use this term to mean an unorganized machine of the following general character: The configurations of the machine are described by two expressions, which we may call the character-expression and the situation-expression. The character

[17] Ibid., p. 11.

> and situation at any moment, together with the imput signals, determine the character and situation at the next moment. The character may be subject to some random variation. Pleasure interference has a tendency to fix the character, i.e., towards preventing it changing, whereas pain stimuli tend to disrupt the character, causing features which had become fixed to change, or to become again subject to random variation.[18]

To make this vague plan more specific, Turing used his network flow techniques to specify a specific, mathematically characterized "P-type unorganized machine." This machine was a "logical computing machine" (as opposed to "practical") with an incomplete description. When an internal configuration was reached for which the action of the machine was undetermined, the random process generator was set in action and recorded tentatively. These tentative actions were followed eventually by either a pain stimulus, in which case they were all cancelled, or by a pleasure stimulus, in which case they were all made permanent. All of this was subjected to mathematical precision in this formal "P-type" machine.

Turing concluded the paper by calling for an attempt to supply not only discipline to the machine, as his scheme called for, but also for initiative--one of those intangibles that characterize living beings and is so hard to include in a machine. This was a call for help or wishful thinking, as he gave no indication how

[18] Ibid., p. 17.

this task might be fulfilled.

Turing's move to Manchester redounded in attempts to further his drive for a thinking machine, for at Manchester he soon had a working, powerful computer which he was able to program so as to attempt, albeit on a very limited scale, to test his ideas about training an unorganized machine to think. In fact, having a powerful new computer at his disposal to test his learning machine theory helps to explain why he would leave NPL before the computer he designed (ACE) was completed. The spirit and program of his work at this time were captured in his famous, popular, philosophical tract published in Mind in 1950, entitled "Computing Machinery and Intelligence."

This paper is best viewed as a continuation of the earlier paper, "Intelligent Machinery," reflecting, especially, on the experiments Turing had been able to perform on the Manchester computer. Although it addressed essentially all of the issues of the earlier "Intelligent Machinery," there were two significant new developments: a clear focus on behaviorist philosophy, which was implicit in all of Turing's previous work on thinking machines; and a systematic, detailed rebuttal of all of the objections to the possible existence of thinking machines which he had discussed piecemeal and in passing in his earlier works.

"Intelligent Machinery" is based upon the experiment Turing called the "imitation game." An interrogator is separated from a man and a machine and is allowed to ask them questions. From these questions the interrogator is to determine which answerer is the human and which the machine. By "machine" is meant an electronic digital computer. The interrogator is allowed to ask questions concerning intellectual, but not physical, properties. Turing argued that he could adequately answer the vague question "can machines think?" by answering the determinate question "are there imaginable digital computers which would do well in the imitation game?" In so doing, Turing exhibited his behaviorist philosophy. A machine was considered to think if its behavior was such that it could not be distinguished from the mental behavior of men. Thinking, for example, did not refer, in Turing's opinion, to such an inaccessible property as consciousness. Matters of thought were only to be decided on attributes which were observable--of observable behavior. Not only did this philosophical view provide a criterion for when a machine thinks, it also provided a vantage on how to build thinking machines--entirely consistent with his actual attempts at Manchester. For example, there was no need to make the internal workings of the machine actually parallel the internal workings of the human mind, just as long as the machine provided answers to questions of it in the way that a thinking man would if so queried.[19]

Turing then used the format of the imitation game to refute various arguments that machines were unable to think. The arguments, given together with his refutations, were as follows:

(1) Theological objection: Thinking is a function of man's immortal soul, and God has given this soul only to man. Therefore, machines do not have the capacity to think. Rather than assault this argument directly, Turing pointed out the inconsistency among religious beliefs as evidence that man cannot determine from theology a certain position on thinking faculties. Turing also argued that theological arguments are unsatisfactory in any account on historical and scientific grounds.

(2) "Heads in the sand" Objection: The consequences of machines which can think are so dreadful that we should not consider the possibility, but simply hope for the best. Turing pointed out that no refutation is needed to this argument--that it is emotional, not rational.

(3) Mathematical Objection: Gödel's incompleteness theorem and similar results demonstrate the limitations of any discrete

[19] It has already been discussed why we need not provide exact analogies of the human nerves in our machines. On the other hand, in his paper on computable numbers, the Turing machines are designed as exact mental, but not physical, analogues of the brain when computing numbers. Turing would probably argue that this similarity between man and Turing machine here simply reflects the logical constituents involved in computing a number.

state machines. Turing's response was:

> . . . although it is established that there are limitations to the powers of any particular machine, it has only been stated, without any sort of proof, that no such limitations apply to the human intellect. . . . Further, our superiority can only be felt on such an occasion in relation to the one machine over which we have scored our petty triumph. There would be no question of triumphing simultaneously over all machines. In short, then, there might be men cleverer than any given machine, but then again there might be other machines cleverer again, and so on.[20]

(4) Argument from Consciousness: Machines do not have the consciousness to write, say, a sonnet according to their emotions, except by a chance manipulation of symbols. This lack of consciousness demonstrated that machines can not think. Turing argued that this is a solipsist position which, although logically neat, was generally avoided in philosophy because it closed off any possible communication.[21]

(5) Arguments from Various Disabilities: "I grant you that you can make machines do all the things you have mentioned, but you will never be able to make one do X," for various X including: be kind, fall in love, enjoy strawberries and cream, learn from

[20] "Computing Machinery and Intelligence," p. 445.

[21] It is clear that the behavior of different men is sufficiently alike, and that their behavior is sufficiently unlike the behavior of machines to argue that there is a difference between men and machines. Thus, it might be reasonable to accept the consciousness of other men, but not of machines. This is the least reasoned of Turing's refutations.

experience, do something really new. Turing attacked this position in several ways. First, he pointed out that the only basis that people have to hold any one of these positions is by induction on their experiences with humans and with machines, and then referred to all of the philosophical discussion of the inadequacy of induction as an acceptable methodological principle. Second, he pointed out that he could design mechanisms which could do any one of the tasks mentioned above. Third, he argued that when, for example, if a critic were to complain that a machine he had designed did not "really enjoy" strawberries and cream the way in which a human does, the critic would be arguing unfairly--that the critic's objection would be _post hoc_ and that his machine would have satisfied the requirements set by the critic if judged according to Turing's behaviorist principles. Fourth, in passing, Turing discussed futuristic possibilities of machines programming themselves to show that machines could do something "really new."

(6) _Lady Lovelace's Objection_: The machine [Babbage's Analytical Engine] "has no pretensions to _originate_ anything. It can do _whatever we know how to order it_ to perform." Turing made several points against this position: First, although possibly what Lovelace said is true about Babbage's Analytical Engine, there is no reason to believe there could not be some other machine capable of original thought. Second, even the

Analytical Engine could mimic any machine in question if provided with the suitable programming. Third, the premise behind this objection (that once one knows something, one knows all of its consequences) is false. This was evident in mathematics, for example, where one has to work hard to attempt to find the consequences of an axiom system. Fourth, there is no way of knowing whether men, themselves, can originate anything. Perhaps every thought is implanted from previous teachings.

(7) Argument from continuity in the nervous system: The nervous system is not a discrete-state machine. Hence, it can not be mimicked by a discrete-state machine. Turing argued that this criticism has no bearing on the interrogation game and therefore, according to his behaviorist model, had no bearing on the possibility of thinking machines.

(8) Argument from the Informality of Behaviour: Men do not operate according to a fully determined set of rules, while machines do. Therefore, machines do not have the flexibility and spontaneity of human thought. First, Turing demonstrated that there is a logical flaw (undistributed middle) in the argument, which becomes apparent when it is formulated as a syllogism:

> If each man had a definite set of rules of conduct by which he regulated his life, he would be no better than a machine.
>
> There are no such rules.

Therefore, men cannot be machines.

Second, Turing observed that this argument did not make the important distinction between "rules of conduct" (such as "stop upon seeing a red light when driving") and "laws of behaviour" (such as "if you pinch him, he will squeak"). Moreover, scientific observation, he argued, is not sufficient to determine the absence of laws of behavior. For example, he claimed that he had fairly simple programs for MADAM for which scientific observation was incapable of determining the laws of behavior—although they did exist. Therefore, the second premise in the syllogism would also remain unfounded.

(9) The Argument from Extra-Sensory Perception: Humans have extra-sensory abilities such as clairvoyance, telepathy, pre-cognition, and psycho-kinesis which the machine cannot imitate. Turing stated that he did not understand the point very well, but that he did know that ESP preempted the ordinary laws of nature and that there is no way of foretelling what would happen in the imitation game or any other ordinary situation if ESP were involved.

Turing explicitly stated that he had no direct arguments to convince his critics of his views on thinking machines. His intention was to develop computer facilities capable of exhibiting the thinking properties he believed a machine could have. To that end, he used MADAM at Manchester as a rudimentary percursor of his ideal thinking machine. Turing's responsibilities at Manchester consisted mainly of the programming of MADAM, and included were programs teaching MADAM to play chess. However, Turing realized how far technology was from his goal of an adequate thinking machine. He estimated that even if visual retention was not required of his machine, a memory would be required on the order of 10^9 storage units--two orders of magnitude greater than current technology could provide. The almost insurmountable problem he foresaw, however, was the programming of such a machine. He estimated that it would take sixty programmers fifty years of errorless work to complete the programming of a thinking machine. So Turing returned to his dream of constructing an unorganized mind which could be educated through constructive interference.[22]

Turing provided more to the development of the theoretical information sciences than just his many arguments for the similar functioning of the nervous system of the brain and computing machinery. His invention of the Turing machine became the most celebrated achievement of early automata theory, for the Turing machine provided a theoretical model of the physical automata, with all of the same theoretical possibilities and

[22] In fact, Turing did some experiments concerning such a "child machine," but the unorthodoxy stood in the way of success.

limitations, but with none of the physical limitations peculiar to any particular physical machine. In fact, Turing's 1936 paper on computable numbers introduced many of the ideas fundamental in the development of automata theory.

Turing machines are not physical machines at all. They are paper machines whose fundamental components, operations, and programming have been described in a precise mathematical fashion, but which do not present any physical mechanism for carrying out these functions and operations. In fact, this is the most important aspect of the machines. In dealing with these machines, Turing characterized the basic functions and operations of a wide spectrum of mechanical computing devices which vary essentially only in their physical mechanisms, their capacity to store information, and their speed of operation.

What Turing machines do, in effect, is give a schematic picture of the logically necessary parts of a computing device and its operation: the necessary conponents of the device itself, the possible and impossible uses of such devices, the procedures necessary to a device carrying out such tasks, and the communication channels between the device and the outside world. Although Turing never explicitly stated in the 1936 paper what the logically necessary physical components of a computing machine were, one can easily abstract from the description of his machine (which is intended to incorporate only the essential properties of computing a number) the essential features of any such computing machine. They include the following:

(i) Some input and output equipment which transmits information to and from the computing device. In the case of the Turing machine it is the paper tape upon which information is read to the machine. It is the same paper tape upon which information is returned from the machine (albeit possibly in a different code).

(ii) A place for storing information either introduced from outside the machine or created by the machine itself in the process of computing. The cells of the paper tape not being scanned at the moment by the machine played this role for Turing.

(iii) A place for the machine to do its computation. Turing considered this to be the analogue of the human computer's scratch paper. In his machine the "scratch paper" consisted of the cells of the paper tape.

(iv) A device for recording information within the machine--the analogue of the human computer's pencil. The device in the Turing machine was the one which either wrote or erased a stroke in the scanned square.

(v) A reader which "reads" the information provided to the machine. This was performed in the Turing machine by the vaguely described equipment which scanned the cell of the paper tape in the machine at any instant ("the scanned square").

(vi) A mechanism (called by Babbage "the mill") which does the physical manipulation of the information. Perhaps the recording device (iv) could be considered as part of the "mill." In Turing's machine the mechanism which moved the tape left and right and possibly the recording device constituted the "mill."

(vii) A controlling device which directs the reader, recorder, mill, input, and output when and how to operate. In the case of the Turing machine the control consisted of a set of rules built into the machine which regulated the way in which it acted and a set of internal states which, together with the information in the reader, determined which rule was in effect.

By examining the 1936 paper one can determine not only the necessary components of a computing machine, but also the way in which these components operate to perform their computations. In Turing's machine, information to be coded into numerical systems comprehensible to the machine was of two kinds: numerical data upon which the machine was to compute, and instructions as to which operations the machine was to carry out on the numerical data. These instructions could either be pre-programmed into the machine, in which case the Turing machine was a special purpose machine capable only of carrying out that one pre-programmed task. Alternatively, the instructions could be entered on the paper tape (in the same way the numerical data was) to a universal machine,[23] which would make the universal machine carry out this one specific activity for this particular input.

Once the information was entered into the machine, the reader would examine part of the information (the rest of the information being stored on part of the paper tape not being examined by the reader at the time). The reader would report the information on the scanned square to the control. Depending on the information in the scanned square and the "internal configuration" of the machine, the control would choose one of the rules programmed into the machine. The control would then send the order to the mill to print or erase or possibly to move the tape one cell to the left or right. The reader would examine the new "scanned square," send a report to the control, whereupon the procedure would be carried out again. This routine would be repeated until the control ordered the machine to stop. Then, using the appropriate coding, the information could be read off of the paper tape as output.

While the mechanics in a physical computer's computation might differ from the mechanics of computation in a Turing machine,

[23] The universal machine is discussed in detail in Chapter Two.

essentially the same functional process would obtain in both cases in the transaction of a computation. The same logically necessary parts and operations must occur in the physical as well as in the theoretical automata. Moreover, the Turing machines exhibited the theoretical range of computation. If a certain computation could be carried out by a Turing machine, it could be carried out by a physical machine--although, for practical purposes, the computation might be prohibitively long or costly to be feasible. Similarly, if no Turing machine were capable of carrying out a particular computation, then no physical machine could be devised to carry out the computation. Thus, the theoretical bounds for physical computation were established. Now it was only necessary to consider the (crucial, but merely practical) bounds to physical computation.

It was this fact that made Turing's paper so important to the development of a theory of automata. Automata theory is the mathematical study of operation of the computation of functions. As one source[24] defines the subject:

> Much of mathematics studies functions, rules $f:X \to Y$ for assigning to each element s of the set X an element f(x) of the set Y. In computer science we seek to realize functions by computations, which are sequences of data manipulations under the control of a program. Programs are run by machines. In specifying a machine we have to specify the syntax, or legitimate forms, for the data structures that may occur as input

[24] Encyclopedia of Computer Science, 1975.

> to the machine, as descriptions of the internal state of the machine, and as output from the machine. A program must be so presented as input structure that it will be incorporated into the internal state structure in a fashion that will cause the machine to process data in such a way as to realize some given function. The semantics of the machine direct how it will change state, read input and provide output at each stage, and thus determine how the program will be interpreted.
>
> Automata theory is the mathematical study of questions of realization, decomposition, simulation, complexity, and computation abstracted from the above considerations.

Turing's 1936 paper is the beginning of automata theory because the Turing machine was the first mathematically precise characterization of a machine which would carry out the operations described in the above paragraph. In his account Turing explicitly discussed the syntax of the tape and m-configurations (internal state) and the semantics which directed the operation of the machine at each moment dependent on the m-configuration and scanned square.

The 1936 paper included other results of importance in automata theory besides the definition of the Turing machine. It included a discussion of the various types of functions which could and could not be computed. In carrying out this discussion, Turing applied many results from mathematical logic to the question of computability of functions.[25] Today, these questions of computability are often rephrased in terms of questions about formal grammars, due to the importance of Gödel's work on formal systems,

[25] Actually, to the computability of decimal numbers.

Chomsky's work on formal grammars, and the development of programming languages. Of particular concern are two questions:

(a) which languages a particular automaton can recognize, i.e., for a given string of symbols can the automaton determine whether or not they are grammatical members of a given formal language; in other words, is there a computable function which is 1 for the code number of grammatical members of the formal language, and 0 otherwise;

(b) which languages a particular automaton can generate, i.e., which languages can have their grammatical strings of symbols as the output of a particular automaton; in other words, is there a computable function which will give as output the code numbers of the grammatical strings of symbols.

Here is a survey of results Turing drew from mathematical logic on the question of computability.[26]

(1) G8del coding. G8del coding, which was originally developed for the incompleteness theorem, was used by Turing for the input and output of both instructions and data for his Turing machines. It provided a way of coding languages so that they could be the input or output of a machine which only accepted

[26] Although Turing did not phrase his results precisely this way in terms of formal grammars, it has been arranged this way here to compare his results with the modern theory of automata.

numerical data. The Gödel number of a particular string of symbols in a formal language could be put into a Turing machine as argument of the characteristic function for the grammatical strings of a particular language (what the Turing machine is programmed to do). Similarly, the various Gödel numbers of the grammatical strings of a formal language that could be printed as output as 1,2,3,... were entered as input into a Turing machine programmed to be the language generator for a particular formal language.

(2) Recursive unsolvability of the halting problem. The halting problem asks: Given a Turing machine in an arbitrary configuration with an arbitrary finite amount of information in the machine, will the Turing machine eventually halt (complete its computation)? Turing showed that this problem is recursively unsolvable, i.e., he showed there is no Turing machine R which, given the description of a particular Turing machine S, its internal configuration, and the state of the data in it, will always provide a "yes" or "no" answer as to whether S will halt. This was because, as Turing proved, for any candidate for R, there is a relevant input (the Gödel number of some machine S and its internal configuration and state of its data) for which R will not halt.

It is clear how this is relevant to automata theory. If the result were applied to those Turing machines which are F-generators or F-recognizers (for some formal language F), then there would be no guarantee for all such Turing machines (although there might be for any particular machine) that the machine will ever decide whether a particular string (appropriately Gödel coded) is a grammatical string in Language F or, alternatively, whether the Turing machine can halt in generating the nth grammatical string in the formal language.

(3) The Negative solution of the Entscheidungsproblem. Recursion theorists were quick to realize that the halting problem was useful in showing the recursive unsolvability of a variety of problems. Instead of showing directly that some problem P is recursively unsolvable (i.e., that there is no Turing machine which solves it), many showed that a problem P in question could be reduced to solving the halting problem and, since the halting problem was recursively unsolvable, then so was problem P . This led to the study of relative recursiveness, where A is recursive relative to B if the recursiveness of B implies the recursiveness of A .

Turing used this technique of relative recursiveness with the halting problem to demonstrate the recursive unsolvability of the Entscheidungsproblem. That is, Turing showed that there is no Turing machine which will recognize the provable formulas in a formal logical system. While this result was of momentous importance in the foundations of mathematics, it also had a different but important role in automata theory. It provided the first real example of a result in automata theory and thereby set out the type of problem to be attacked and the method to be used. That is, a logical system is a type of formal language, and in such a system the provable theorems correspond to the grammatical strings. Thus, in showing the recursive unsolvability of the Entscheidungsproblem, Turing had shown that there is no Turing machine which will recognize the grammatical strings in one particular formal language (predicate calculus with equality). This became the model for later results.

(4) Recursive enumerability.

In the 1936 paper Turing also discussed recursive enumerability. In particular, he discussed the possibility of enumerating the computable sequences. He showed that the problem of enumerating the computable sequences is equivalent to the problem of finding out whether a given number is the description of a circle-free machine[27] and that, by applying the diagonal process

[27] A circle-free machine is, in modern parlance, a machine with no infinite loops.

argument,[28] one can show that there cannot be any such general process. Turing briefly discussed as well the possible uses and misuses of the diagonal process in determining the enumerability of particular sets. This approach has been used to great effect in automata theory in determining which sets of grammatical strings could possibly be enumerated recursively and has subsequently led to the identification of Turing machines with languages generated by type 0 grammars,[29] an important concept in modern automata theory.

[28] If a series is doubly indexed by the positive integers, for example:

$$\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} & \cdots \\ a_{21} & a_{22} & a_{23} & a_{24} & \cdots \\ a_{31} & a_{32} & a_{33} & a_{34} & \cdots \\ a_{41} & a_{42} & a_{43} & a_{44} & \cdots \end{matrix}$$

then the diagonal process chooses the sequence a_{11}, a_{22}, a_{33}, ... The diagonal process was used for the first time by Cantor in showing that the real numbers were not denumerable. However, it later became an extensively used technique in recursion theory to show the existence of various types of functions which were not recursive.

[29] A type 0 grammar is the weakest, least restricted sort of grammar, A grammar is a four-tuple $G=(V_N,V_T,P,S)$, where V_N and V_T are disjoint finite sets, P is a finite subset of $(V_N \cup V_T)^* \times (V_N \cup V_T)^*$, and $S \in V_N$. V_N is called the non-terminal vocabulary (elements are called metavariables or syntactic classes), V_T is called the terminal vocabulary, P is called the set of productions, and S is the start symbol. For more details, see Hopcroft and Ullman, Formal Languages and Their Relation to Automata.

(5) Variations on the Turing machine.

In the 1936 paper Turing considered the possibility of alterations on his machine. In particular, he discussed the possibilities of multiple tapes or two-dimensional tapes (like ordinary scrap paper). His conclusion, although not proved, was that these additions to his basic machine provide no theoretical increase in computing power over the standard Turing machine. Turing also mentioned in passing the possibility of non-deterministic Turing machines, where the choice of operation at any step in the computation is not always completely determined and may be chosen at times arbitrarily from some (usually finite) set. This showed the foresightedness and intuition of Turing. The study of various types of Turing machines has held a major share of the interest in automata theory and it has been shown that, as Turing believed, most alterations of the basic Turing machine do not alter the theoretical computing power. Turing was also foresightful in recognizing the interest in non-deterministic Turing machines.[30]

[30] Turing's intuitions about the lack of change in making a Turing machine non-deterministic also were correct, for it has been shown that if a non-deterministic machine accepts some formal language L, then there is some deterministic Turing machine which also accepts the language—where "accepts" means that the machine halts with encoded members of the language as input.

(6) Universal Machines.

Perhaps the most important single novelty of Turing machines described in the 1936 paper were the universal machines. A universal machine is a machine which can be fed not only data to process, but also rules governing operations which can make it act like any particular Turing machine. At the beginning of §6 of his paper Turing describes his idea for a universal Turing machine:

> It is possible to invent a single machine which can be used to compute any computable sequence. If this machine U is supplied with a tape on the beginning of which is written the S.D. [standard description] of some computing machine M , then U will compute the same sequence as M . . . .
>
> The manner of operation of M' [a machine which will write down the configuration of M] could be make to depend on having the rules of operation (i.e., the S.D.) of M written somewhere within itself (i.e., within M'); each step could be carried out by referring to these rules. We have only to regard the rules as being capable of being taken out and exchanged for others and we have something very akin to the universal machine.

Turing pointed out how M' could be modified to print out figures and to do the "scratch work" necessary to do computations. All of this discussion was ancillary to a technical description of a universal Turing machine, which is given in §7 by breaking down the programming into subroutines and giving a technical description of each part. Apparently, Turing believed that such a machine would be impractically large--although today this belief has been shown to be incorrect.[31]

Turing's universal machine provided the first instance of an interpretive routine; this is "a computer program that performs the instructions of another program, where the other program is written in some machine-like language. By a machine-like language, we mean some way of representing instructions having, say, operation codes, addresses, etc."[32] These interpretive machines were used at first as ways of translating machine language into more tractable languages for programming. Programming languages have outmoded this technique; however, interpretive machines have established their place in the computer industry and are used even today.

The greatest importance of the universal Turing machine, however, was not as an interpreter, but as the prototype of the general purpose computer. In the tradition of Babbage's Analytical E- -ine, Turing's universal machine was designed to perform any computation possible on any Turing-like machine. No longer was it necessary to have a special machine for each special operation. This was among the most important ideas

[31] See Jeremy Bernstein, "Calculators: Self-Replications" in Experiencing Science.

[32] Knuth, op. cit., p. 197.

behind the development of a new generation of high-powered electronic digital computers in the 1940's and 1950's. It was what made programming possible. Certain basic machine operations could be concatenated and put into effect in various ways determined by the instructions put into the universal machine to reproduce in effect any particular Turing machine. This corresponds to the programming from basic operations into more complicated operations capable in the general, automatic, stored program computer. In fact, the universal Turing machine provided the theoretical characterization of stored programming, for it was able to process instructions as well as data in order to alter instructions in the middle of a computation.

The universal Turing machine was also important in recursion theory and automata theory. In both fields, the universal machine provides a convenient and powerful way of describing the power of all the Turing machines at once and showed that one mechanical device can do all of the different types of computations capable of human computers. However, the existence of a universal machine showed more than this:

> It shows that, for computing partial functions of one variable, there is a critical degree of "mechanical complexity (that of P_z [the universal machine]) beyond

> which all further complexity can be absorbed into increased size of program and increased use of memory stage.[33]

Von Neumann developed a theory of complexity around this fact--as will be discussed in the next chapter.

In retrospect, Turing made important contributions in three broad areas toward the development of a theory of information processing: technical contributions to the theory of automata, arguments for the possibility of artificial intelligence, and a general characterization of both brains and computers as types of computing automata.

Turing offered many different arguments that machines can think and made many attempts to build examples of artificial intelligence. Most important among these attempts were his plan for teaching unorganized machines to think by "constructive interference," his plan for testing his artificial intelligence thesis by programming existing computing machinery to carry out purely mental activities such as chess-playing or language learning, his behaviorist methodology for studying artificial intelligence in terms of the imitation game, and his coherent and systematic

[33] Hartley Rogers, Theory of Recursive Functions and Effective Computability, p. 23.

rebuttal of all the objections to the possibility of thinking machines.

This vociferously argued belief that machines can think was important toward convincing people of the strong similarity between the functioning of the computer and the human nervous system. The Turing machine provided an abstract model equally applicable to the functional process carried out by the brain or by the computer. This allowed Turing to provide a schematization of the basic functional processes of computing and the basic components of any computing automaton.

Turing also made more techincal contributions by suggesting or giving examples of how technical results from mathematical logic could be applied to the theory of automata. Turing machines provided the starting point for automata theory because they provided a theoretical characterization of physical automata without any of the physical limitations peculiar to any particular machine. Turing was the first to attempt to mathematically examine computable functions in terms of a mathematically precise, theoretical machine--a technique which was to later become common in automata theory. He was the first to illustrate the applicability of logical results and techniques--in particular, Gödel coding, recursive unsolvability of the halting program, the negative solution of the Entscheidungsproblem, recursive enumerability, and universal machines--to the theory of automata.

He was the first to contemplate the possible ways of varying the Turing machine (randomizers, multiple tapes, multi-dimensional tapes) to establish the relation of these variations to the power of computing.

Unfortunately, Turing never attempted to organize these various and fecund ideas into either a coherent theory of artificial intelligence or of automata. While his 1936 paper suggested a number of important ideas for both automata theory and artificial intelligence, he never pursued the consequences of these ideas. Although he carried out a number of experiments at Manchester to support his beliefs about artificial intelligence and though he had the imitation game as the basis for a theory of artificial intelligence, he never felt compelled to suggest (at least in writing or, in any clear way, to his colleagues) a program to be carried out to develop his ideas which would show how his work fit in and what specific projects should be carried out in the future. He was an amateur at his research and did not try to organize it in any systematic way so that others could work within a mutual framework on joint problems with him.

It will be shown in the next chapter that this is in direct contrast with von Neumann's approach. He attempted to use Turing's work on theoretical machines, Shannon's work on communication, and McCulloch and Pitts' work on the brain as the theoretical basis for a theory of automata. Logic, applied mathematics, and probability were used by von Neumann to turn a series of loosely related engineering results into a science of computing, including studies of the logical structure of computing machines, the theoretical principles of computing, and how to affect them with physical equipment.

Thus, even if Turing, himself, did not develop his automata ideas, von Neumann used them in his science of computing and, in particular, in part of it known as the theory of automata. Turing's technical applications of logic were also picked up in the 1950's and 1960's by theoretical computer scientists who developed them into the theory of automata. Turing's work on artificial intelligence was well known, and the imitation idea was much discussed (especially by philosophers), but the actual effect (other than inspirational) he had on the specific projects in artificial intelligence seems minimal. In particular, no one picked up directly on his "constructive interference" idea, nor did any experimental computer scientist make the imitation game the basic tool of his research. However, his arguments for the possibility of artificial intelligence were among the earliest stated, seemed to have been popular, and were perhaps persuasive in breaking down barriers against "artificial intelligence." (As the name implies, it is still something thought of as categorically different from human thinking.) Perhaps if Turing had lived longer automata theory and artificial intelligence would have profited more from his ideas.

Chapter Seven: Von Neumann's Contribution to Theoretical Computer Science: His Theory of Automata

Von Neumann's major contribution to the conceptual revolution in the field of information processing was the development of his theory of automata. This theory was developed near the end of his career, after he had had significant experience with mathematical logic, mathematical physics, and with the development of physical computing equipment. Von Neumann seems to have begun considering questions about the theory of automata before the second world war, through his interactions with Turing concerning the possibility of using Turing machines to simulate thought. However, von Neumann's sustained and serious attempt to develop a detailed theory of automata dates only from 1948 until his death in 1957.

The Program for a General Theory of Automata

Five extant documents contain von Neumann's thoughts on the theory of automata:

(1) "The General and Logical Theory of Automata," read at the Hixon Symposium on September 20, 1948, in Pasadena, California. The paper was published in 1951 and is contained in von Neumann's Collected Works (V.288-328).

(2) "Theory and Organization of Complicated Automata," a series of five lectures delivered at the University of Illinois in December, 1949. Edited by Arthur Burks, they comprise the first half of von Neumann's Theory of Self-Reproducing Automata, pp. 29-87.

(3) "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components," based on notes taken by Dr. R. S. Pierce of von Neumann's lectures in January, 1952 at the California Institute of Technology. These are contained in von Neumann's Collected Works (V.329-378).

(4) "The Theory of Automata: Construction, Reproduction, Homogeneity," a manuscript written by von Neumann in 1952 and 1953, edited by Arthur Burks, and included as the second half of Theory of Self-Reproducing Automata, pp. 89-380.

(5) The Computer and the Brain, a series of lectures which von Neumann intended to deliver at the Silliman Lectures at Yale University in 1956, but which the illness that terminated his life precluded. The completed fragments of the lectures were published in book form posthumously in 1958.

Automata theory is a field essentially created by von Neumann. There was no standard literature before von Neumann, and there were no predetermened topics or questions. Thus, von Neumann was left to develop the material as he wished. In the five publications noted above, von Neumann attempted to develop a comprehensive theory which would unify the study of computing machines and living organisms under one theory of information processing equipment. In doing this, von Neumann discussed the following issues: the logical structure of the functioning of the information processing equipment; the differences and similarities of components and overall systems in artificial and natural information processers; the purposes and uses of artificial processing equipment and the ways in which they could be conjoined with natural systems; the notion of complexity and its ramifications for information processing systems; the relation of reproducing, but especially self-reproducing, automata to self-replication in natural organisms; the problem of reliability in a system having unreliable components (and the difference in philosophy between natural and artificial systems for developing reliability); and the possible applications of this theory to the construction of computing machinery. These issues were discussed both in a general, informal way (especially in the Hixon Symposium) and within the context of a formal, mathematical theory (developed in his next three works on the subject).

A definite progression of ideas can be seen in von Neumann's five papers. The first paper, "The General and Logical Theory of Automata," was a short paper which was given in one lecture. In this paper von Neumann introduced all of the major issues which he examined in his theory of automata (as listed above), but did little more than initiate the discussion, marshall the most basic of facts, and indicate what needed to be done in the future.

The second publication, "Theory and Organization of Complicated Automata," was broken into five lectures. In the first two lectures von Neumann repeated the general comments that he had made in "The General and Logical Theory of Automata"—although he did seem to have a better grasp on his subject by this time, which comes across in better organization, more developed arguments, and more specific programs to be carried out in the future. The third lecture von Neumann devoted to statistical theories of information, for he argued that it is only through probability and statistics that one can develop an automata theory that will deal adequately with the realistic assumption that components of information systems are unreliable. The last two lectures were devoted to one of the most important concerns of von Neumann's theory of automata: complicated automata. Von Neumann was interested in a theory of automata which would adequately explain the human nervous system, the large and fast computers being designed during his day, and automata capable of self-reproduction

--like humans and, he hoped, some day artificial automata. All of these were extremely complicated. Von Neumann hoped to determine what was meant by "complexity" in such systems and what effect, if any, it had on such systems.

In the third lecture of "Theory and Organization of Complicated Automata," von Neumann argued that a statistical and probabilistic theory was needed for the study of the reliability of machines incorporating unreliable components. "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components" provided that theory. In this paper von Neumann first extended first order mathematical logic to a probabilistic system which could deal with such automata as Turing's universal machine and McCulloch and Pitts' neural networks. He then proved that a technique that he had suggested in "Theory and Organization of Complicated Automata" for making a system containing unreliable components reliable to any degree of accuracy actually worked as he suggested that it would.

In the fifth lecture of "Theory and Organization of Complicated Automata: von Neumann provided a sketch of an axiomatic model of a mechanical device which would self-reproduce and showed mathematically that there was nothing contradictory about such a machine. In "The Theory of Automata: Construction, Reproduction, Homogeneity" von Neumann provided four additional biological models of self-reproducing automata, together with discussion of their similarity to the reproductive rules carried in the human genetic system.

In The Computer and the Brain von Neumann collected all of the comparisons that he had made between natural and artificial systems in the other four works on automata theory and presented a coherent explanation of the similarities and differences in the functioning of the computer and the brain as information processors.

The Logical Theory of Automata

Von Neumann's ultimate aim in automata theory was to develop a precise mathematical theory which would compare computers and the human nervous system. His aim was not to study the particular mechanical devices or physiological devices which carry out the information processing, but only to study the structure and functioning of the entire system.

> The organisms can be viewed as made up of parts which to a certain extent are independent, elementary units. We may, therefore, to this extent, view as the first part of the problem the structure and functioning of such elementary units individually. The second part of the problem consists of understanding how these elements are organized into a whole, and how the functioning of the whole is expressed in terms of these elements.
>
> The first part of the problem is at present the dominant one in physiology. It is closely connected with the most difficult chapters of organic chemistry and of physical chemistry, and may in due course be greatly helped by quantum mechanics. I have little qualification to talk about it, and it is not this part with which I shall concern myself here.

> The second part, on the other hand, is the one which is likely to attract those of us who have the background and the tastes of a mathematician or a logician. With this attitude, we will be inclined to remove the first part of the problem by the process of axiomatization, and concentrate on the second one.[1]

Von Neumann treated the workings of the individual components of the systems, whether natural or artificial, as "black boxes," devices which work in a certain way (specified by axioms), but whose internal mechanism was unknown (and need not be known for these purposes).

> Axiomatizing the behavior of the elements means this: We assume that the elements have certain well-defined, outside, functional characteristics; that is, they are to be treated as "black boxes". They are viewed as automatisms, the inner structure of which need not be disclosed, but which are assumed to react to certain unambiguously defined stimuli, by certain unambiguously defined responses.[2]

He then proceeded to point out that this axiomatic approach, like any axiomatic approach in applied mathematics, had certain advantages and disadvantages. On the positive side, all situations were idealized and the various components were assumed to act universally in a precise, clear-cut manner. This allowed a study of the highly complicated behavior of organisms like computers

[1] For the remainder of this chapter, GALTA will signify "General and Logical Theory of Automata."

[2] GALTA, p. 289.

or human nervous systems which would be impossible unless such regularities and simplifications were assumed.

> This being understood, we may then investigate the larger organisms that can be built up from these elements, and the general theoretical regularities that may be detectable in the complex synthesis of the organism in question.[3]

On the negative side, and this was a criticism heard from many neurophysiologists, there was no way of testing the validity of the axioms and there was physiological evidence to indicate that the situation in the human nervous system was not as simple as von Neumann's analysis made it out to be. Moreover, even accepting von Neumann's analysis, one knew nothing about the physiological operation of the individual elements from his analysis. Nevertheless, von Neumann was convinced that the axiomatic approach, which had worked so successfully for him in clarifying a complicated situation in quantum mechanics, was the way in which to get a handle on the problems of information processing in complicated automata such as electronic computers or the human nervous system.

Von Neumann pointed to Turing's work on Turing machines and McCulloch and Pitt's axiomatic model of
the neural networks of the brain as the two most significant developments towards a formal theory of automata and indicated how

[3] GALTA, pp 289-290.

each of these developments was equivalent to a system in formal logic. Although von Neumann believed these were important steps towards a mathematical theory of automata, he was dissatisfied with what the approach of formal logics could contribute to a theory of automata that would be useful in the actual construction of computing machinery.

> There exists today a very elaborate system of formal logic, and specifically of logic applied to mathematics. This is a discipline with many good sides, but also with certain serious weaknesses. This is not the occasion to enlarge upon the good sides, which I have certainly no intention be belittle. About the inadequacies, however, this may be said: Everybody who has worked in formal logic will confirm that it is one of the technically most refractory parts of mathematics. The reason for this is that it deals with rigid, all-or-none concepts, and has very little contact with the continuous concept of the real or of the complex number, that is, with mathematical analysis. Yet analysis is the technically most successful and best-elaborated part of mathematics. Thus formal logic is, by the nature of its approach, cut off from the best cultivated portions of mathematics, and forced onto the most difficult part of the mathematical terrain, into combinatorics.
>
> The theory of automata, of the digital, all-or-none type, as discussed up to now, is certainly a chapter in formal logic. It would, therefore, seem that it will have to share this unattractive property of formal logic. It will have to be, from the mathematical point of view, combinatorial rather than analytical.[4]

Von Neumann pointed out, for example, that formal logic had never been concerned with how long a finite computation actually

[4] GALTA, p. 303.

is--as long as it was finite, it was treated in the same way. But this did not take into consideration the important fact for the theory of computing that certain finite computations are so long as to be practically prohibitive, and that other computations, though finite in length, are so long as to be theoretically impossible since they would take more time or room than there is in the physical universe. Von Neumann also observed that at each step in a computation there is a non-zero probability of error; thus, if computations were allowed to become arbitrarily long, the reliability of the computation would approach a probability of zero. Finally, he pointed out that, in actual practice, people using computers allot a certain fixed time to complete particular computations--a fact to which formal logics are not sensitive. Thus von Neumann suggested that the formal logical approach be modified in two ways to develop a "logic of automata": by considering the actual lengths of the "chains of reasoning" and by allowing for a small degree of error in logical operations. As von Neumann stated the second alteration:

> the operations of logic (syllogisms, conjunctions, disjunctions, negations, etc., that is, in the terminology that is customary for automata, various forms of gating, coincidence, anti-coincidence, blocking, etc., actions) will all have to be treated by procedures which allow exceptions (malfunctions) with low but non-zero probabilities.[5]

[5] GALTA, p. 304.

He indicated that such a logic would have a more analytical and less combinatorial approach than formal logic. In fact, it would resemble formal logic less than it would Boltzmann's theory of thermodynamics, which implicitly manipulated and measured a concept related to information.

Von Neumann contended that such a logic of automata was necessary if computing machinery were to be constructed of any significantly higher complexity than was being built at the time. *A fortiori*, he argued, it would be that much less likely that one would understand the workings of even more complicated systems such as the human central nervous system, without such a logic of automata. The higher complexity of the natural over the artificial system he illustrated by contrasting the techniques utilized in the two systems for handling malfunctions.

> The basic principle of dealing with malfunctions in nature is to make their effect as unimportant as possible and to apply correctives, if they are necessary at all, at leisure. In our dealings with artificial automata, on the other hand, we require an immediate diagnosis. Therefore, we are trying to arrange the automata in such a manner that errors will become as conspicuous as possible, and intervention and correction follow immediately.[6]

The consequences were enormous. A single error broke down the process of computation until the error could be painstakingly

[6] GALTA, pp. 305-6.

found in the artificial machine, whereas natural systems operated smoothly, with no loss of time, and with roughly the same accuracy as the artificial systems. A logic of automata, von Neumann believed, would provide the theory necessary to construct an artificial automaton which could utilize the natural technique of error handling.

Thus von Neumann argued that there were two components to his formal theory of automata: a logical theory of control and information and a probabilistic theory of reliability of machines with unreliable components. While von Neumann admitted that the probabilistic theory was the more important for the development of modern computing equipment, he pointed out that the logical theory was a necessary preliminary to the probabilistic theory.

As von Neumann recognized, the logical theory of automata amounted to the application of formal logics to the theory of automata. By 1948 these results had already been extensively developed by the mathematical logicians. Von Neumann was not responsible for the development of new logical results in the theory of automata, but he was responsible for the application of existing results. In fact, Turing, with his work on computable numbers, and McCulloch and Pitts, with their work on neural networks, had already made significant strides in applying results from mathematical logic to the theory of automata. What then was von Neumann's contribution to this aspect of the theory of automata?

Von Neumann contributed to the logical theory of automata by compiling the results of Turing, McCulloch and Pitts, and others into a coherent theory, drawing from it some remarkable and novel facts about the structure of complexity in information processing machinery, utilizing the results to develop a theory of and models of self-reproducing automata, and extending the theory to consider the probabilistic features of real automata.

Von Neumann's contributions strictly to the logical theory of automata can be understood by contrasting his work with the earlier work of Turing and of McCulloch and Pitts. As von Neumann pointed out,[7] McCulloch and Pitts intended their work only "as a simple mathematical, logical model to be used in discussions of the human brain." The fact that their system turned out equivalent to a formal logic was an interesting and important fact that they were careful to point out. However, they were merely intent on synthesizing an axiomatic model of the neural network and never deigned to develop the ramifications of their work for a general theory of automata. Turing, on the other hand, was interested in formal logics, in particular the Entscheidungs-

[7] TOCA, p. 43. For the remainder of this chapter TOCA will signify "Theory of Complicated Automata."

problem, and analyzed the process of computing in terms of a machine which would carry out these computations. While he wa heavily involved with the theoretical possibilities of compu he never attempted to draw a formal theory of automata from the results of his 1936 paper on computable numbers. Von Neumann's contribution to a logical theory of automata was in drawing together McCulloch and Pitts' synthetic account of the brain and Turing's analytic theory of computing into a comprehensive theory of automata which considered machine computation and brain processing as two aspects of the same theory and which could be examined by utilizing the results of both approaches.

In each of his first three papers on automata, von Neumann gave detailed examinations of the work of McCulloch and Pitts and of its significance to the theory of automata. What McCulloch and Pitts did was to provide an axiomatic account of the way in which an idealized neuron would work. Rather than axiomatize the functioning of a neuron as it actually existed, they decided to axiomatize a much simpler object which still had all the essential traits of an actual neuron without the incidental complications. Whereas the physiologists criticized this decision, von Neumann applauded it for the simplicity it introduced to the study. Here is von Neumann's lucid description of McCulloch and Pitts' definition of an idealized neuron:

> The definition of what we call a neuron is this. One should perhaps call it a formal neuron, because it certainly is not the real thing. A neuron will be symbolically designated by a circle, which symbolizes the body of the neuron, and a line branching out from the circle, which symbolizes the axon of the neuron. An arrow is used to indicate that the axon of one neuron is incident on the body of another. A neuron has two states: it's excited or not. As to what excitation is, one need not tell. Its main characteristic is its operational characteristic and that has a certain circularity about it: its main trait is that it can excite other neurons. Somewhere at the end of an involved network of neurons the excited neuron excites something which is not a neuron. For instance, it excites a muscle, which then produces physical motion; or it excites a gland which can produce a secretion, in which case you get a chemical change. So, the ultimate output of the excited state really produces phenomena which fall outside present treatment. These phenomena will, for the sake of the present discussion, be entirely disregarded.[8]

Von Neumann pointed out that the best way to regard these formal neurons was as "black boxes" with fixed finite numbers of inputs which receive stimuli and fixed finite numbers of outputs which emit stimuli. These plack boxes were then subject to certain rules:[9]

(1) Each input connection is of one of two types: excitatory or inhibitory;

[8] TOCA, p. 44.

[9] GALTA, p. 309.

(2) The boxes are of one of two types: threshold one or threshold two;

(3) To stimulate a box, there must be at least as many excitatory inputs simultaneously as the threshold level of the box, and no inhibitory inputs;

(4) After a box is stimulated, there is a finite delay time, which is assumed to always be the same, after which the body emits an output pulse; and

(5) Such an output may be carried by appropriate connections to any number of other neuron's inputs, where it will act as the same type of input stimulus as described above.

Formal neural networks were then built out of these formal neurons and the connections between them.

> The"functioning" of such a network may be defined by singling out some of the inputs of the entire system and some of its outputs, and then describing what original stimuli on the former are to cause what ultimate stimuli on the latter.[10]

Once von Neumann had described the neural system, he focused on the crucial result of McCulloch and Pitts' work, the result which made it important to the theory of automata. This important result was that any functioning of the brain which could

[10] GALTA, p. 309.

be described "logically, strictly, and unambiguously"[11] in a finite number of words could be represented by a formal neural network. The significance of this result to the theory of automata was described in detail by von Neumann:

> It is well to pause at this point and to consider what the implications are. It has often been claimed that the activities of the human nervous system are so complicated that no ordinary mechanism could possibly perform them. It has also been attempted to name specific functions which by their nature no ordinary mechanism could possibly perform them. It has been attempted to show that such specific functions, logically, completely described, are per se unable of mechanical neural realization. The McCulloch-Pitts result puts an end to this. In proves that anything that can be exhaustively and unambiguously described, anything that can be completely and unambiguously put into words, is ipso facto realizable in a suitable finite neural network. Since the converse statement is obvious, we can therefore say that there is no difference between the possibility of describing a real or imagined mode of behavior completely and unambiguously in words, and the possibility of realizing it by a finite formal neural network. The two concepts are co-extensive. A difficulty of principle embodying any mode of behavior in such a network can exist only if we are also unable to describe the behavior completely.[12]

Von Neumann reviewed five serious objections to the McCulloch and Pitts' account. First, although the above result showed that certain modes of behavior could be effected in finite formal neural networks, could it be shown as well that these finite networks

[11] GALTA, p. 309.

[12] GALTA, pp. 309-310.

were sufficiently small as to be practical? In particular, could a finite network be determined which was small enough to fit into the organism in question (if certain reasonable assumptions were made about the size of these formal neurons)? Second, could every mode of behavior be put completely and unambiguously into words? With regard to this point, von Neumann discussed the principle of visual analogy. He argued that it was too complex a principle to completely describe, say, all of the visual connections, connotations, and categories associated with the word "triangle." This led von Neumann into a discussion of complexity, which is discussed below. Von Neumann's next two objections concerned situations in which the formal system did not adequately explain the way the nervous system actually functioned: the way in which the nervous system transmitted continuous numbers as in the representation of blood pressure; and the way memory is stored. Finally, von Neumann objected to this account because it did not consider nerve fatigue resulting in a higher threshold after the neuron had fired.

The other formal logical system that von Neumann examined in several of his works on automata was Turing's work on machines and computable numbers. Von Neumann first described the machines and showed that they theoretically characterized all possible (information processing) automata. He then focused on the universal Turing machine, which he saw as Turing's major contribution to the theory of automata, since it embodied all of the Turing machines, and therefore all of the possible information processing automata, in a single automaton. Von Neumann described the principle behind Turing's dramatic result:

> An automaton is "universal" if any sequence that can be produced by any automaton at all can also be solved by this particular automaton. It will, of course, require in general a different instruction for this purpose.
>
> The Main Result of the Turing Theory. We might expect a priori that this is impossible. How can there be an automaton which is at least as effective as any conceivable automaton, including, for example, one of twice its size and complexity?
>
> Turing, nevertheless, proved that this is possible. While his construction is rather involved, the underlying principle is nevertheless quite simple. Turing observed that a completely general description of any conceivable automaton can be (in the sense of the foregoing definition) given in a finite number of words. This description will contain certain empty passages, those referring to the functions mentioned earlier . . ., which specify the actual functioning of the automaton. When these empty passages are filled in, we deal with a specific automaton. As long as they are left empty, this schema represents the general definition of the general automaton. Now it becomes possible to describe an automaton which has the ability to interpret such a definition. In other words, which, when fed the functions that in the sense described above define a specific automaton, will thereupon function like the object described. The ability to do this is no more mysterious than the ability to read a dictionary and a grammar and to follow their instructions about the uses and principles of combinations of words. This automaton, which is constructed to read a description and to imitate the object described, is then the universal automaton in the sense of Turing. To make it duplicate any operation that any other automaton can perform, it suffices to furnish it with a description of the

> automaton in question, and, in addition, with the instructions which the device would have required for the operation under consideration.[13]

Von Neumann then focused on a principle which Turing made critical use of, but did not discuss in detail: "universality is connected with a rigorous theory of how one describes objects and a rigorous routine of how to look up statements in a dictionary and obey them." He pointed out that Turing further utilized logic in the theory of automata by showing, by logical arguments, that it was impossible to construct certain sorts of automata.

> The formal logical investigations of Turing went a good deal further than this. Turing proved that there is something for which you cannot construct an automaton; namely, you cannot construct an automaton which can predict in how many steps another automaton which can solve a certain problem will actually solve it. So you can construct an automaton which can do anything an automaton can do, but you cannot construct an automaton which will predict the behavior of any arbitrary automaton. In other words, you can build an organ which can do anything that can be done, but you cannot build an organ which tells you whether it can be done.

13 GALTA, pp. 314-15.

> This is concerned with the structure of formal logics. . . . It is connected with the theory of types and with the results of Gödel.[14]

This was the extent of von Neumann's discussion of the logical theory of automata. He never systematically developed the logical theory, but only indicated its flavor by discussing some of the main issues. He discussed the contributions of Turing and of McCulloch and Pitts and showed what bearing they had on the theory of automata. He indicated briefly how results from

[14] TOCA, p. 51. There is a protracted discussion of Arthur Burks' investigation of just which results from logic meant by this statement. See Theory of Self-Reproducing Automata, pp. 51-56. (For the remainder of this chapter, SRA will signify "Theory of Self-Reproducing Automata.") Burks wrote to Gödel about this passage. Here is Gödel's reply, as printed in the above work, p. 55:

> I have some conjecture as to what von Neumann may have had in mind in the passages you quote, but since I never discussed these matters with him it is only a guess.
>
> I think the theorem of mine which von Neumann refers to is not that on the existence of undecidable propositions or that on the lengths of proofs, but rather the fact that a complete epistemological description of a language A cannot be given in the same language A, because the concept of truth of sentences of A cannot be defined in A. It is this theorem which is the true reason for the existence of undecidable propositions in the formal systems containing arithmetic. I did not, however, formulate it explicitly in my paper of 1931 but only in my Princeton lectures of 1934. The same theorem was proved by Tarski in his paper on the concept of truth published in 1933 in Act. Soc. Sci. Lit. Vars., translated on pp. 152-278 of Logic, Semantics, and Metamathematics.

mathematical logic could be applied directly to issues about automata. He showed some of the limitations of this logical theory of automata. He then turned to his probabilistic theory to solve some of the more egregious problems left untouched by the logical theory.

The Statistical Theory of Automata

Von Neumann's overriding concern in the development of a statistical (otherwise known as probabilistic) theory of information was the question of reliability of automata with unreliable components. His aim was a theory which would determine the likelihood of errors and malfunctions and a plan which would make what errors that did occur "non-lethal." The problem of reliability led him from a logical to a statistical account.[15]

> To permit failure as an independent logical entity means that one does not state the axioms in a rigorous manner. The axioms are not of the form: If A and B happen, C will follow. The axioms are always of this variety: if A and B happen, C will follow with a certain specified probability, D will follow with another specified probability, and so on. In other words, in every situation several alternatives are permitted with various probabilities.

Von Neumann pointed out that the logical and the statistical theories were not distinct. He argued a well-known philosophical position that probability can be considered as an extension of

[15] TOCA, p. 58.

logic; therefore, the statistical theory of automata was simply an extension of the logical theory of autonata.

There is a striking similarity between von Neumann's extension of automata from a logical theory to a statistical theory and his work along the same lines in the foundations of quantum mechanics. Von Neumann mentioned this similarity in passing:

> Though this new theory of information will be similar to formal logics in many respects, it will probably be closer to ordinary mathematics than formal logics is. The reason for this is that present day formal logics has a very un-analytical, un-mathematical characteristic: it deals with absolutely all-of-none processes, where everything that either does or does not happen is finitely feasible or not finitely feasible. These all-or-none processes are only weakly connected to analysis, which is the best developed and best known part of mathematics, while they are closely connected to combinatorics, that part of mathematics of which we know the least. There is reason to believe that the kind of formal logical machinery we will have to use here will be closer to ordinary mathematics than present day logic is. Specifically, it will be closer to analysis, because all axioms are likely to be of a probabilistic and not of a rigorous character. <u>Such a phenomenon has taken place in the foundations of quantum mechanics</u>.[16]

In fact, von Neumann turned to theoretical physics for an approach to use in his statistical theory of information. He explicitly stated that there were two statistical theories of information "which are quite relevant in this context although

[16] TOCA, p. 62. Underscoring not in the original.

they are not conceived from the strictly logical point of view."[17] He referred to the work of Boltzmann, Hartley, and Szilard on thermodynamics and the work of Shannon on the concept of noise and information on a communications channel. Von Neumann then proceeded to give an informal account of these theories,[18] arguing that this work on thermodynamics should be incorporated into the formal statistical account of automata.

> I have been trying to justify the suspicion that a theory of information is needed and that very little of what is needed exists yet. Such small traces of it which do exist, and such information as one has about adjacent fields indicate that, if found, it is likely to be similar to two of our existing theories: formal logics and thermodynamics. It is not surprising that this new theory of information should be like formal logics, but it is surprising that it is likely to have a lot in common with thermodynamics.[19]

This formal statistical theory of automata von Neumann attempted to develop in "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components."

Von Neumann began this statistical treatment by characterizing automata in the same way McCulloch and Pitts characterized their idealized neurons: as "black boxes" with given inputs and outputs and with rules relating input and output behavior. To

[17] TOCA, p. 59.

[18] For a discussion of this material, see Chapter Five.

[19] TOCA, p. 62.

be mathematically precise, von Neumann offered a precise definition of single output automata:[20]

> Definition 1: A single output automaton with time delay δ (δ is positive) is a finite set of inputs, exactly one output, and an enumeration of certain "preferred" subsets of the set of all inputs. The automaton stimulates its output at time $t + \delta$ if and only if at time t the stimulated inputs constitute a subset which appears in the list of "preferred" subsets, describing the automaton.

More generally, automata were formed by connecting single output automata into networks in any possible way, provided that certain basic rules were followed:[21]

> Single output automata with given time delays can be combined into a new automaton. The outputs of certain automata are connected by lines or wires or nerve fibers to some of the inputs of the same or other automata. The connecting lines are used only to indicate the desired connections; their function is to transmit the stimulation of an output instantaneously to all the inputs connected with that output. The network is subjected to one condition, however. Although the same output may be connected to several inputs, any one input is assumed to be connected to at most one output.

Rather than consider the problem of what could be constructed by means of networks of simple automata (as McCulloch and Pitts had done), von Neumann concentrated on the converse problem: Are

[20] "Probabilistic Logics . . .," in Collected Works, V, p. 330.

[21] Ibid., p. 331.

there a few simple automata which would serve as a basis for all other automata; that is, from which all other automata could be constructed as networks through the proper connection of these basic automata? In preparation for answering this question, he first considered the relation of these automata to the propositional calculus of mathematical logic. He showed that there was a one-one correspondence between single output automata (with given time delay) and propositions of the propositional calculus.[22] He then used the fact that the connectives "and", "or", and "not" were a basis for forming all propositions in the propositional calculus, in order to show that any single output automaton can be constructed as a network of "and", "or", and "not" automata.[23] In fact, von Neumann also showed that this was not the only possible basis for single output automata. For example, he showed that the automaton which represented the Sheffer stroke[24] could be a basis by itself for all other single output automata.

[22] See Theorem 1, Ibid., p. 333, for a proof of the fact.

[23] See Theorem 2, Ibid., p. 334.

[24] The Sheffer stroke is defined by A/B means not-A or not-B. It was well known in logic that this operation was sufficient as the only connective for the formation of propositional calculus. Von Neumann credits G. Gell-Mann and K. A. Brueckner with suggesting its usage in this context.

Other automata, which do not represent well-known logical connectives,[25] were also suggested as bases.

Until this point in the discussion, von Neumann had only considered "circle-free" networks, those in which an output of an organ does not contribute to a later input of the same organ. He demonstrated that, by dropping this restriction, one could construct a wider variety of complicated automata, whose performance depended on events in the indefinitely remote past. Automata, he claimed,[26] could be constructed which could count, do arithmetic, or perform inductive procedures. In fact, he constructed a network which could "remember" which of two inputs was last stimulated[27] and another network which could "learn" (in a Pavlovian way) that if an input a were stimulated, then input b would consequently be stimulated.[28]

This distinction between circle and circle-free machines parallelled an important distinction in mathematical logic--

[25] For example, there is the "majority organ" which, in propositional logic, relates to the proposition ab + ac + bc.

[26] Ibid., p. 340.

[27] See §6.1, Ibid., pp. 342-343.

[28] See §6.3, Ibid., pp. 343-345.

as von Neumann was well aware:[29]

> The use of cycles or feedback in automata extends the logic of constructable machines to a large portion of intuitionistic logic. Not all of intuitionistic logic is so obtained, however, since these machines are limited by their fixed size. . . . Yet if our automata are furnished with an unlimited memory—for example, an infinite tape, and scanners connected to afferent organs, along with suitable efferent organs to perform motor operations and/or print on the tape—the logic of constructable machines becomes precisely equivalent to intuitionistic logic.

The reason for this von Neumann described in detail:[30]

> These general automata |network of circle-free automata| are, in particular, not immediately equivalent to all of effectively constructive (intuitionistic) logics. That is to say, given a problem involving (a finite number of variables), which can be solved (identically in these variables) by effective construction, it is not always possible to construct a general automaton that will produce this solution identically (i.e. under all conditions). The reason for this is essentially, that the memory requirements of such a problem may depend on (actual values assumed by) the variables (i.e. they must be finite for any specific system of values of the variables, but they may be unbounded for the totality of all possible systems of values), while a general automaton in the above sense necessarily had a fixed memory capacity. That is to say, a fixed general automaton can only handle (identically, i.e. generally) a problem with fixed (bounded) memory requirements.

In other words, a fixed general automaton is only primitive, and not general, recursive!

[29] Ibid., pp. 340–41.

[30] Ibid., p. 335.

After concluding a description of automata and the way in which they could be built out of simpler components, von Neumann turned to the problem of error. As he recognized, it is unrealistic to expect faultless performance of the basic components, whether they be biological or electro-mechanical. The basic difficulty, as he saw it, was not that incorrect information might be obtained by an automaton with unreliable components, but that irrelevant results would be produced. As illustration, he considered his rudimentary memory automaton mentioned above. He was able to show that, assuming a probability of ε (with $0 < \varepsilon < \frac{1}{2}$) for the basic components of this automaton to misfire, the probability over time of the final output of the automaton misfiring tends to $\frac{1}{2}$. In other words, the content of the machine has been lost because the behavior of the output was no different from random behavior, due to the accumulation of errors in the basic conponents over time.[31]

To obtain a more precise, mathematical handle on the problem, von Neumann phrased it thus: given the function that an automaton is to perform, given the basic organs from which the automaton is

[31] An error $E>\frac{1}{2}$ just says that the automaton is behaving with the negative of its attributed function with a probability of $1 - E<\frac{1}{2}$ --a significant result!

to be built and the probability ε of malfunction of a basic component, can an automaton be constructed from the stipulated components with the stipulated function such that the probability of error in the final output of the automaton is less than δ , for a given $\delta > 0$? In particular, is there a lower bound on the size of δ ? Is there any technique which will assist in lowering the probability of error of the overall machine?

One apparent limitation to the overall reliability of the automaton seemed to be that δ must be at least as great as ε . However, von Neumann designed a technique, called "multiplexing," whereby δ could be made arbitrarily small for most (fixed) values of ε . The technique consisted of carrying all messages simultaneously on N lines instead of on a single line. Thus automata were conceived as black boxes with bundles of lines, instead of single lines, as input and output. A critical value Δ , with $0 < \Delta < \frac{1}{2}$, was chosen. Then the stimulation of at least $(1 - \Delta)N$ bundles was considered as a malfunction. The basic idea was that any function to be carried out was to be carried out in N identical machines. The output that was produced by the majority of these machines was then considered to be the true output. Von Neumann proved that, if N were made large enough, the malfunctioning of a small number of the basic components would only cause a malfunctioning of the entire automaton with arbitrarily small probability.

Von Neumann suggested multiplexing as a way to control error in the construction of machines. He found, upon completing detailed calculations for an electronic computing machine with 2500 vacuum tubes and an actuation of each tube on the average every five microseconds, that, if one desired an average of eight hours between errors, one needed to multiplex roughly 17,500 times. A similar calculation for the human nervous system with average error time of 10,000 years showed that the nervous system required multiplexing roughly 28,000 times. Von Neumann concluded the following:[32]

> It should be noticed that this construction [multiplexing] multiplies the number of lines by N and the number of basic organs by 3N . . . . Our above considerations show, that the size of N is 20,000 in all cases that interest us immediately. This implies that such techniques are impractical for present technologies of componentry (although this may perhaps not be true for certain conceivable technologies of the future), but they are not necessarily unreasonable (at least not on grounds of size alone) for the micro-componentry of the human nervous system.

Nonetheless, he advocated the use of multiplexing in future electronic machinery.

Von Neumann recognized that there were a number of difficulties with his probabilistic theory of automata.[33] First, although he had shown that, on grounds of size, the human nervous system

[32] *Ibid.*, p. 368.

[33] For details, see *Ibid.*, §11.

could actually be multiplexed, he recognized that the nervous system did not have the same logical structure as he assumed of automata in this study. For example, the nervous system transmits information by analog as well as by the digital means, whereas he assumed strictly digital transmission. Second, in his mathematical analysis, he had assumed that there was randomness of stimulation of inputs. Yet, as he pointed out, there was likely to be strong statistical correlation between pulses at different times in the same organ when one considered networks which allowed feedback. Third, in this work he had assumed that the probability of component malfunction was a constant, independent of time and of all previous inputs. Yet, he was able to show cases where this assumption was clearly unrealistic.

In fact, McCulloch also objected to a number of assumptions in von Neumann's probabilistic theory of automata as it applied to the functioning of the human nervous system:[34]

> He [von Neumann] had decided for obvious reasons to embody his [probabilistic] logic in a net of formal neurons that sometimes misbehaved, and to construct of them a device that was as reliable as was required in modern digital computers. Unfortunately, he made three assumptions, any one of which was sufficient to have precluded a reasonable solution. He was unhappy about it because it required neurons far more reliable than he could expect in human brains. The piquant

[34] McCulloch, "What is a Number . . .," in Embodiments, pp. 11-13.

> assumptions were: first, that failures were absolute--not depending upon the strength of signals nor on the thresholds of neurons; second, that his computing neurons had but two inputs apiece; third, that each computed the same single Sheffer stroke function. . . .
> There were two other problems that distressed him. He knew that caffeine and alcohol changed the threshold of all neurons in the same direction so much that every neuron computed some wrong function of its input. Yet one had essentially the same output for the same input. . . . Associated, unobtrusively, with the problem is this: That of the 16 possible logical functions of neurons with two inputs, two functions cannot be calculated by any one neuron. They are the exclusion "or," "A or else B," and "both or else neither"--the "if and only if" of logic. . . .
> The last of von Neumann's problems was proposed to the American Psychiatry Association in March 1955. It is this. The eye is only two logical functions deep. Granted that it has controlling signals from the brain to tell it what it has to compute, what sort of elements are neurons that it can compute so many different functions in a depth of 2 neurons (that is, in the bipolars and the ganglion cells)?

McCulloch then cited the relevant physiological research to show the limitations of von Neumann's idealized, mathematical theory. However, the wealth of related physiological research indicated the importance of such an idealized, mathematical approach to the theoretical understanding of the brain.

Complexity and Self-Reproducing Automata

Von Neumann's main concern in the theory of automata was to provide understanding of the theoretical functioning of the human nervous system and of the modern electronic computer. It is clear from the great number of neurons and electronic valves these automata contain and the variety of tasks they can perform that both types of automata are highly "complicated"--in some non-technical sense of the term. Von Neumann was interested in this concept of complication and in what ramifications it might have in the functioning of automata.

> There is a concept which will be quite useful here, of which we have a certain intuitive idea, but which is vague, unscientific, and imperfect. This concept clearly belongs to the subject of information, and quasi-thermodynamical considerations are relevant to it. I know no adequate name for it, but it is best described by calling it "complication." It is effectivity in complication, or the potentiality to do things. I am not thinking about how involved the object is, but how involved its purposive operations are. In this sense, an object is of the highest degree of complexity if it can do very difficult and involved things.[35]

He stated, in terms of a paradox, the problem concerning complication that most interested him. "There are two states of mind, in each of which one can put himself in a minute, and in each of which we feel that a certain statement is obvious. But each of these two statements is the opposite or negation of the other."[36] The first involved conclusions which could be made from the study of living organisms: "That [phylogenetic

[35] TOCA, p. 78. In TOCA, GALTA, and "Probabilistic Logic . . ." von Neumann discusses the application of Shannon's work to the theory of automata.

[36] TOCA, p. 78.

evolution] starts from simple entities, surrounded by an unliving amorphous milieu, and produces something more complicated. Evidently, these organisms have the ability to produce something more complicated than themselves."[37]

> The other line of argument, which leads to the opposite conclusion, arises from looking at artificial automata. Everyone knows that a machine tool is more complicated than the elements which can be made with it, and that, generally speaking, an automaton A , which can make an automaton B , must contain a complete description of B and also rules on how to behave while effecting the synthesis. So, one gets a very strong impression that complication, or productive potentiality in an organization, is degenerative, that an organization which synthesizes something is necessarily more complicated, of a higher order, than the organization it synthesizes. This conclusion, arrived at by considering artificial automata, is clearly opposite to our early conclusion, arrived at by considering living organisms.[38]

Von Neumann was particularly concerned with the paradox because it suggested that there was a rift between the laws governing living organisms and artificial automata--the two domains that he was trying to unify with his theory of automata. The quotation also indicated the relevance of the relf-reproduction issue to a discussion of information processing automata. Any automaton which is in the process of self-reproduction must pass on to its

[37] TOCA, p. 79. See further discussion of this point in GALIA, p. 310.

[38] TOCA, p. 79.

progeny information relating to its basic description and to its behavior. This is equally true in the case of genetic coding and of Turing machine description numbers.

Von Neumann replied that certain automata, including certain types of biological organisms, were sufficiently complicated themselves so that they could produce even more complicated automata, whereas other automata, including machine tools, have not the internal complexity to produce automata even as complicated as themselves.

> There is thus this completely decisive property of complexity, that there exists a critical size below which the process of synthesis is degenerative, but above which the phenomenon of synthesis, if properly arranged, can become explosive, in other words, where synthesis of automata can proceed in such a manner that each automaton will produce other automata which are more complex and of higher potentialities than itself.[39]

Von Neumann was not clear on how to measure the complexity, nor on what this critical size is. However, it was clear from his discussion that the critical size was large; that it might require the combination of millions of basic parts.

Von Neumann addressed a related question--the means by which an automaton can produce an even more complicated automaton --through an illustration of the working of the universal Turing

[39] TOCA, p. 80.

machine. The universal Turing machine was chosen for examination because of its precise mathematical formulation which enabled precise study of its structure. When supplied with information I and with a description I_A of Turing machine A, the universal Turing machine $\bar{A}$ is able to produce the same output as machine A would if supplied with input I. I_A consists of a set of rules, each of which describes how machine A would behave if it were in a particular state of mind and were confronted with a particular datum of information. Machine $\bar{A}$ is designed so that it can read this set of rules I_A, for any Turing machine A--no matter how complicated--and put them into action; thus simulating machine A. To read, interpret, and act on any such set of rules requires a certain, necessary level of complexity on the part of $\bar{A}$.

Nevertheless, this still does not explain the way in which the universal machine manages to impart greater complication to the machines it simulates. Von Neumann explained the way $\bar{A}$ compensated in the following quotation:[40]

> $\bar{A}$ is able to imitate any automaton, even a much more complicated one. Thus a lesser degree of complexity in an automaton can be compensated for by an appropriate increase of complexity of the instructions. The importance of Turing's research is just this: that if you construct an automaton right, then any additional requirements about the automaton can be handled by

[40] TOCA, p. 50.

> sufficiently elaborate instructions. This is only true if A is sufficiently complicated, if it has reached a certain minimum level of complexity. In other words, a simpler thing will never perform certain operations, no matter what instructions you give it; but there is a very definite finite point where an automaton of this complexity can, when given suitable instructions, do anything that can be done by automata at all.

Although von Neumann never explicitly stated that this was the means by which other automata than $\bar{A}$ compensated for greater complexity in their progeny, it is clear from the context that he considered this a possible mechanism for biological organisms.

Von Neumann applied a result due to Turing to demonstrate an ironic fact about automata. He showed that, although an automaton, such as the universal Turing machine, could be constructed which could carry out any computation capable by an automaton, one cannot construct an automaton which will predict the behavior of an arbitrary automaton! Turing, in proving the recursive unsolvability of the halting problem,[41] had shown that, due to logical considerations, there is no automaton which can determine the number of steps it will take another automaton to solve a problem—even if it is known that the latter automaton can solve the problem. Von Neumann suggested the relationship of these

[41] See "On Computable Numbers . . . ."

results from mathematical logic to the problem of complication:[42]

> This result of Turing is connected with the structure of formal logics . . . It is connected with the theory of types and with the results of Gödel. The feature is just this, that you can perform within the logical type that's involved everything that's feasible, but the question of whether something is feasible in a type belongs to a higher logical type. It's connected with the remark . . .: that it is characteristic of objects of low complexity that it is easier to talk about the object than produce it and easier to predict its properties than to build it. But in the complicated parts of formal logic it is always one order of magnitude harder to tell what an object can do than to produce the object. The domain of the validity of the question is of a higher logical type than the question itself.

Thus, another close relation was struck between mathematical logic, information theory (complexity), and the theory of automata.

Although the universal Turing machine provided a precise object to study because of its exact mathematical description, it was not a self-reproducing automaton in the sense that its output was not another machine like itself, but rather only a

[42] TOCA, p. 51. The result of Gödel's referred to is the fact that the truth of a language A cannot be defined in A . See K. Gödel, "On Undecidable Propositions of Formal Mathematical Systems" (1934) and A. Tarski, Logic, Semantics, and Metamathematics (1933), pp. 152-278. There is a fascinating discussion of this point in von Neumann's work on automata theory and its relation to Gödel's work in logic in TOCA, pp. 53-56. Cited there are letters between Arthur Burks and Gödel regarding Gödel's work.

paper tape which characterized the behavior of another--not necessarily similar--automaton. In fact, it was the original universal Turing machine, and not its output, that acted like the other automaton. Von Neumann was not satisfied with Turing's machine as a characterization of self-reproducing automata. Consequently, he planned to build four mathematical models of automata which actually would produce as output a machine like themselves. In fact, he completed only two of these models, known as the kinematic and and the cellular models.[43]

Before describing the individual models, von Neumann outlined a general description of a self-reproducing automaton in terms of a complex of machines A, B, C, and D.[44] Automaton A was designed to construct another automaton upon receipt of a description of the machine. The universal Turing machine would be such an automaton A. However, von Neumann preferred that the description be carried out on other than a punched tape (as in Turing's case) for structural reasons. This description was what von Neumann called the "instruction" and labelled I. Thus, various I could be inserted in A with differing results.

Automata B, C, and D were also constructed so that

[43] See Arthur Burks' comments in von Neumann's Theory of Self-Reproducing Automata, pp. 93-95.

[44] See GALTA, pp. 316-17 and TOCA, pp. 82-86.

instructions I might be inserted into them. Automaton B was designed to be a "reproducer." Its function was to read the inserted I and duplicate a copy of the instructions. Automaton C was a control mechanism. Automata A and B were combined and controlled by C. When I was inserted in A, C would cause A to construct the automaton described by I. C would then cause B to duplicate instructions I and insert them in the new automaton just constructed by A. Next C would separate the new construction from system A + B + C as an independent automaton. Call the complex A + B + C automaton D. Let I_D be the instructions which describe machine D. If I_D were inserted into automaton A of machine D, a new automaton E would be constructed, which would be self-reproductive. This provided the logical description of a self-reproducing automaton.

This logical schema for self-reproducing automata was not intended only for use in the construction of artificial self-reproducing automata. Von Neumann believed that it provided insight into the functioning of natural self-reproductive systems as well.

> . . . it is quite clear that the instruction I_D is roughly effecting the functions of a gene. It is also clear that the copying mechanism B performs the fundamental act of reproduction, the duplication of the genetic material, which is clearly the fundamental operation in the multiplication of living cells. It is also easy to see how arbitrary alterations of the system E, and in particular of I_D, can exhibit certain

> typical traits which appear in connection with mutation, lethally as a rule, but with a possibility of continuing reproduction with a modification of traits. It is, of course, equally clear at which point the analogy ceases to be valid. The natural gene does probably not contain a complete description of this object whose construction its presence stimulates. It probably contains only general pointers, general cues. In the generality in which the foregoing consideration is moving, this simplification is not attempted. It is, nevertheless, clear that this simplification, and others similar to it, are in themselves of great and qualitative importance.[45]

Von Neumann used this model for further study of the process of mutation.[46]

Von Neumann's kinematic model was his earliest[47] and most simplistic model of self-reproduction. His aim was to design an automaton, built from a few types of elementary parts, which could construct other automata like itself from a stockpile of the elementary parts. As von Neumann designed it, the kinematic model was to float in a reservoir filled with an unlimited supply of these elementary parts. The constructing automaton would contain a description of the automaton it was to build. It

[45] GALTA, pp. 317-18.

[46] See GALTA, p. 318, and TOCA, pp. 86-87.

[47] Von Neumann gave three lectures on automata at the Institute for Advanced Study in June, 1948, where he described the kinematic model. See TOCA, pp. 80-82 for details. More on the model can be found in GALTA, pp. 315-16 or in SRA, pp. 93-94.

would sort through the pieces in the reservoir until it found the ones it needed, which would then be assembled according to the instructions.

Units were needed to control the operation, to seize and identify elementary parts, and to shape them into the new automaton. Von Neumann identified eight basic units in his kinematic model. Four were for information processing and control. A stimulus organ received and transmitted stimuli. Coincidence and inhibitory organs managed logical operations. A stimuli producer produced stimuli. The fifth unit was a rigid member, from which a frame for the new automaton could be constructed. Connections between parts of the new automaton were to be made by a fusing organ. Connections were to be broken by a cutting organ. The final unit was a muscle, the task of which was to produce motion through contraction. Von Neumann claimed these were sufficient to carry out the operations of the kinematic model.

Von Neumann's second model of self-reproduction, his cellular model, was created with the aid of the logician, S. M. Ulam,[48] and was more amenable to mathematical examination than was the

[48] See S. M. Ulam, "Random Processes and Transformations," Proceedings of the International Congress of Mathematicians, 1950, Vol. II, pp. 264-75, Providence, RI, 1952.

kinematic model. The aim of this and the other two models was to avoid muscular, geometrical, and kinematic considerations and concentrate only on logical factors:

> The constituent organs that are needed for the automaton construction must thus be found and acquired in space, they must be moved and brought into contact and fastened together in space, and all automata must be planned as true geometric (and kinematical and mechanical) entities. . . . But even the simplest approach, which disregards the above-mentioned properly mechanical aspects entirely requires quite complicated geometrical-kinematical considerations. Yet, one cannot help feeling that these should be avoided in a first attempt like the present one: in this situation one ought to be able to concentrate all attention on the intrinsic, logical-combinatorial aspects of the study of automata. The use of the adjective formalistic . . . was intended to indicate such an approach—with, as far as feasible, an avoidance of the truly geometrical, kinematical, or mechanical complications. The propriety of this desideratum becomes even clearer if one continues the above list of avoidances, which progressed from geometry, to kinematics, to mechanics. Indeed, it can be continued (in the same spirit) to physics, to chemistry, and finally to the analysis of the specific physiological, physico-chemical structures. All these should come in later, successively, and about in the above order; but a first investigation might best avoid them all, even geometry and kinematics.[49]

The aim of the cellular model was to remove the kinematic considerations by constructing an automaton which consisted of stationary objects, normally in a quiescent state which, under certain circumstances, would assume an active state. In the

[49] SRA, p. 102.

cellular model, the automaton consisted of an infinite, two-dimensional array of square cells. Each cell contained the same basic automaton, which could assume any of twenty-nine internal states: one unexcited state, twenty excitable states, and eight excited states. Each cell was connected to the four contiguous cells, and rules were listed for the transmission of excitation from one cell to its neighbors. Thus the cellular model was intended as a two-dimensional, idealized model of a neural network. Self-reproduction occured when the initial logical structure of the automata, coded in terms of the cell states in a finite region of the cellular plane, was copied in a distinct finite region of the cellular plane which had originally been quiescent.

Von Neumann wanted a model of self-reproduction which was more like natural nervous systems than his cellular model. Thus, he proposed an excitation-threshold-fatigue model. It was to be based on the cellular model. However, each cell was to have a fatigue mechanism and a threshold. Although von Neumann never explicitly described the details of the excitation-threshold-fatigue model,[50] in his treatise on probabilistic logics he did

[50] This model was to be described in "Construction, Reproduction, Homogeneity." However, the work was never completed. Arthur Burks edited the fragmentary treatise, from which this information is taken.

describe an idealized neuron with threshold and fatigue mechanism.[51] According to this model, after each firing of a neuron, there is an absolute refractory period and later a relative refractory period before the neuron returns to normal threshold level. During the absolute refractory period, a neuron can not be excited under any circumstances. Other times, a neuron will be excited by any number of inputs exceeding the threshold level at the time. However, during the relative refractory period, the threshold level is increased, necessitating greater levels of stimulation to fire the neuron. This threshold and fatigue mechanism was to be incorporated into the rules of the automaton housed in each cell. Otherwise, this model was to function in the same way as the cellular model.

The fourth system, the continuous model, was to be a refinement of the excitation model. The plan was to consider the cells as "(infinitesimal) elements of a continuously extended medium" rather than as "discrete entities" as in the cellular and excitation models.[52] The plan was to use simultaneous differential equations to consider the chemical, physical, and biological

[51] See "Probabilistic Logic . . .," pp. 372-78.

[52] SRA, p. 103.

as well as logical factors in the firing of neurons.[53] The details of this model were never worked out.

The Computer and the Brain

Von Neumann's overriding concern throughout the development of his theory of automata was for a unified study of modern computing machines and the human nervous system. Even in his early works on automata, GALTA and TOCA, he pointed out similarities and dissimilarities of the two systems when viewed as digital processors of information. He intended to present a detailed comparison of the two types of automata in the Silliman Lectures at Yale University in 1956. However, the illness that eventually ended in his death precluded his completing this project. The fragments of the lectures which were completed were published under the title The Computer and the Brain. When considered in light of his earlier comments, it is possible to reconstruct his comparison of the computer and the brain as digital information processers.

Von Neumann began by comparing the basic components, i.e., the switching organs, in the natural and artificial systems. In

[53] Von Neumann's training as a chemical engineer and his work with fluid dynamics perhaps prompted this approach.

the case of the brain, the basic switching organ is the neuron. In the case of the computing machine of von Neumann's day, it was the vacuum tube. (Von Neumann envisioned making similar comparisons with other types of artificial switching organs as technology advanced.) Included were detailed study of the speed, energy comsumption, size (volume), efficiency, and number of basic switching organs required in each system. He concluded that the artificial switching units required greater volume, consumed more energy, and were 10,000 times less efficient (in ergs/binary action) than their natural counterparts. However, the artificial organs did have one advantage over neurons which possibly could conpensate in future machines for all other disadvantages; they were considerably (roughly 5,000 times) faster than neurons and did not require nearly so long for recovery between firings.

Von Neumann also considered the brain and the computer as total information processing systems. He contrasted the number of multiplications necessary to carry out certain basic computations, the precision, and the reliability of the two types of systems, and compared their means of memory storage, input and output, control, and balance of components. Several conclusions were drawn from these comparisons important to the practical theory of computing. However, von Neumann recognized that so little was known about the human brain and that computer construction was in such a nascent stage that any conclusions were highly liable to change.

One conclusion concerned the means of memory storage in the two systems. Although the actual mechanism utilized by the brain to store information was not understood, it was clear from all that the mind remembers in one lifetime that the brain must store (according to von Neumann's estimate) at least 10^{10}, and possibly 10^{15}, bits of information in memory. Von Neumann hoped for significant improvement in artificial memones; however, it was clear that the best artificial memory of his time would fall at least several orders of magnitude short in storage capacity of even the conservative 10^{10} figure. In fact, von Neumann pointed to the lack of accessible memory storage as the most severe limitation on the computers of his day. Two factors entered the storage problem: capacity and access. Whereas the vacuum tube provided quick access to memory, its bulk and other technical factors precluded its use as the sole type of organ in a large memory. On the other hand, von Neumann foresaw the use of acoustic delay lines and cathode ray tubes for information storage. While they provided extensive storage, they were slow to access. Thus, he suggested a technique for the construction of computer memories involving a hierarchy of memories, each level of memory having more storage, but slower access time, than the preceding level. This plan was implemented by von Neumann and became part of standard computer design.

A second point to come out of this comparison of the brain and the computer involved the materials used in the construction of automata. According to von Neumann, the computer engineer of the future would be well advised to imitate in artificial switching organs the means of construction and materials used in neurons, because of the neuron's superiority in smaller scale, greater precision, lower energy requirements, and ability for self-repair.

> The materials which we are using are by their very nature not well suited for the small dimensions nature uses. Our combinations of metals, insulators, and vacuums are much more unstable than the materials used by nature . . . If a membrane is damaged it will reconstruct itself, but if a vacuum tube develops a short between its grid and cathode it will not reconstruct itself. Thus the natural materials have some sort of mechanical stability and are well balanced with respect to mechanical properties, electrical properties, and reliability requirements. Our artificial systems are patchworks in which we achieve desirable electrical traits at the price of mechanically unsound things. . . . the differences in size between artificial and natural automata are probably connected essentially with quite radical differences in materials.[54]

In fact, von Neumann paid attention to the physiological studies of the make-up of the neuron with intent to apply it to the development of new artificial switching organs.

[54] TOCA, p. 70.

A third observation made by von Neumann from his comparison of artificial and natural automata involved the differing philosophies concerning the treatment of error in the two systems. Using Shannon's theory of communication and some probability theory, von Neumann calculated the probability for error per individual action of a switching organ provided that the computer and the brain were to function errorlessly for an empirically estimated reasonable period of time. Assuming a mean free path of approximately seven hours between fatal errors on a contemporary machine like ENIAC, von Neumann computed that the probability of error per actuation of a switching organ should be of the order of 10^{-13}. Assuming that fatal errors really are "fatal" in humans and thus requiring a mean free path of sixty years between fatal errors, he computed that the probability of error per neuron actuation should be of the order of 10^{-20}.

Yet, he knew that the vacuum tube and suspected that the neuron were not nearly that accurate. Thus, both systems must have some means for dealing with errors which did not result in total collapse of the system. He pointed out that artificial automata are designed so that, every time there is an error, the machine will stop, locate the error, and correct it. This is the idea behand the technique of multiplexing; a multiplexed machine will do computation a number of times and, if not enough of them agree, the machine will not operate. The philosophy for handling errors is radically different in natural automata:[55]

> It's very likely that on the basis of the philosophy that every error has to be caught, explained, and corrected, a system of the complexity of the living organism would not run for a millisecond. Such a system is so well integrated that it can operate across errors. An error in it does not in general indicate a degenerative tendency. The system is sufficiently flexible and well organized that as soon as an error shows up in any part of it, the system automatically senses whether this error matters or not. If it doesn't matter, the system continues to operate without paying any attention to it. If the error seems to the system to be important, the system blocks that region out, by-passes it, and proceeds along other channels. The system then analyzes the region separately at leisure and corrects what goes on there, and if correction is impossible the system just blocks the region off and by-passes it forever. The duracion of operability of the automaton is determined by the time it takes until so many incurable errors have occurred, so many alterations and permanent by-passes have been made, that finally the operability is really impaired.

Although von Neumann never cited any evidence to show that this is an accurate sketch of what occurs in the human nervous system, it is known that he did pay close attention to work on the physiology of the brain. Presumably, this physiological research supported his theory. His clear intention was to apply the techniques utilized in the natural systems to the new artificial systems to be constructed. However, there were

[55] TOCA, p. 71.

several difficulties with this project:[56]

> To apply the philosophy underlying natural automata we must understand complicated mechanisms better than we do, we must have more elaborate statistics about what goes wrong, and we must have much more perfect statistical information about the milieu in which a mechanism lives than we now have. An automaton can not be separated from the milieu to which it responds. By that I mean that it's meaningless to say that an automaton is good or bad, fast or slow, reliable or unreliable, without telling in what milieu it operates. . . . in discussing a computing machine it is meaningless to ask how fast or slow it is, unless you specify what types of problems will be given to it.

Machine as Thinker or Computer?

Von Neumann and Turing held radically different views about the types of problems to which computers should be set. From the very beginning, Turing's intention was to design machines which could carry out any computations capable of a human computer. Later, he claimed that machines could carry out any sort of intelligent behavior. In other words, his aim was to design machines which could perform any task possible through digital information processing.

[56] TOCA, pp. 71-72.

Von Neumann's view of the role of computers was much narrower than that of Turing:

> It makes an enormous difference whether a computing machine is designed, say, for more or less typical problems of mathematical analysis, or for number theory, or combinatorics, or for translating a text. We have an approximate idea of how to design a machine to handle the typical general problems of mathematical analysis. I doubt that we will produce a machine which is very good for number theory except on the basis of our present knowledge of the statistical properties of number theory. I think we have very little idea as to how to design good machines for combinatorics and translation.[57]

According to von Neumann, the difficulty in designing machines to carry out such activities was due to the differences between the computer and the brain. His comparison of the two had shown that the brain outperformed the computer in many essential ways, and this precluded the computer from accomplishing many tasks other than pure numerical computation. For some unknown reason, von Neumann insisted on working with the mature and complex adult mind rather than the learning mind which Turing hoped to model. Von Neumann hoped, instead, to use the one advantage of the computer over the adult mind, speed of computation, in the role he assigned computers.

The role of computers in numerical computations in engineering

[57] TOCA, p. 72.

appeared straight-forward and obvious to von Neumann. Yet, he also saw the computer as an important tool in scientific research in two important and more subtle ways. One involved using the computer to do computations in situations where the proper equations were known, but where the number of calculations to be made was prohibitively large for the human computer. Such was the case in physical chemistry and in quantum mechanics.

> A considerable segment of chemistry could be moved from the laboratory field into the purely theoretical and mathematical field if one could integrate the applicable equations of quantum theory. Quantum mechanics and chemistry offer a continuous spectrum of problems of increasing difficulty and increasing complexity, treating, for example, atoms with increasing numbers of electrons and molecules with increasing numbers of valence electrons. Almost any improvement in our standards of computing would open important new areas of application and would make new areas of chemistry accessible to strictly theoretical methods.[58]

The other application of computers in pure scientific research involved use of the computers to generate mathematical models. Von Neumann pointed out that, in certain areas of applied mathematics, such as the theories of turbulence and of compressible, non-viscous flow, it is known that solutions involve non-linear partial differential equations. However, the only breakthroughs in determining the particular mathematical

[58] TOCA. p. 33.

solutions have come through insight gained by experimentation. Von Neumann believed that computer modelling could replace this experimentation:[59]

> I wanted to point out that there are large areas in pure mathematics |what we today call "applied mathematics"| where we are blocked by a peculiar inter-relation of rigor and intuitive insight, each of which is needed for the other, and where the unmathematical process of experimentation with physical problems has produced almost the only progress which has been made. Computing, which is not too mathematical either in the traditional sense but is still closer to the central area of mathematics than this sort of experimentation is, might be a more flexible and more adequate tool in these areas than experimentation.

In fact, von Neumann put the computer to work in the way his theoretical work suggested that he employ it. He used the Institute for Advanced Study computer for mathematically modelling of three-dimensional weather flows in an attempt to improve weather forecasting. He used ENIAC to model a two-dimensional hydrodynamical problem.[60] He also investigated the use of the computer for solving other flow problems involving hyperbolic and

[59] TOCA, p. 35.

[60] See his "A Study of a Numerical Solution to a Two-Dimensional Hydrodynamic Problem" (1959), reprinted in Collected Works, V, 611-52.

parabolic partial differential equations for both industry and the army.[61]

Thus, von Neumann's theoretical observations about the brain and the computer bore practical advantage in the design and use of the computer for scientific research. His hope was that this theoretical work would become ever more important in the practical design of computers.

[61] See the first and second reports "On the Numerical Calculation of Flow Problems" which were presented in 1948 and are reprinted in Collected Works, V, 664-750. For further references, see the editor's note, Ibid., p. 664.

Bibliography

Ackermann, Wilhelm. "Begründung des 'tertium non datur' mittels der Hilbertschen Theorie der Widerspruchsfreiheit." Mathematische Annalen, 93 (1924-5), 1-136.

----------. "Beiträge zum Entscheidungsproblem der mathematischen Logik." Mathematischen Annalen, 112 (1936), 419-432.

----------. "Zum Hilbertschen Aufbau der reelen Zahlen." Mathematische Annalen, 99 (1930), 118-133.

Ambrose, Alice. "A Controversy in the Logic of Mathematics." The Philosophical Review. 42 (1933), 594-611.

----------. "Finitism in Mathematics." Mind, n. s., 8 (1933), 152-153.

Anderson, Alan Ross. Minds and Machines. Englewood Cliffs, N. J., 1964.

Anisimov, S. F. Man and Machine. Report No. JPRS 1990-N. Washington, D. C.: Joint Publications Research Service, 1959.

Apostle, Hippocrates George. Aristotle's Philosophy of Mathematics. London, 1936.

Arbib, M. A. Brains, Machines, and Mathematics. New York, 1964.

----------. "Turing Machines, Finite Automata and Neural Nets." Journal of the Association of Computing Machinery, 8 (1961), 467-475.
Machinery

Armer, Paul. Attitudes Towards Intelligent Machines. Santa Monica, CA.: The RAND Corporation, 1962.

Arzela, Cesare. "Funzioni di linee." Atti della Reale Academia dei Lincei Rendiconti, 5 (1889), 342-348.

Ascoli, G. "Le Curve Limite di una Varieta data di Curve." Atti della Reale Academia dei Lincei Rendiconti, 18 (1883), 521-586.

Ashby, W. R. "Can a Mechanical Chess Player Outplay Its Designer?" British Journal for the Philosophy of Science, 3 (1952), 44-47.

----------. "The Cerebral Mechanism of Intelligent Action." In Perspectives in Neuropsychiatry. Ed. D. E. Richter. London, 1950.

----------. Design for a Brain. London, 1952.

----------. "Design for an Intelligence-Amplifier." In Automata Studies. Ed. C. E. Shannon and J. McCarthy. Princeton, N. J., 1956.

----------. Introduction to Cybernetics. London, 1956.

----------. "The Nervous System as Physical Machine: With Special Reference to the Origin of Adaptive Behavior." Mind, 56 (1947), 113-127.

Baire, Rene. "Sur les fonctiones de variables reeles." Diss. Milan 1899. Paris, no. 977.

Baire, R., E. Borel, and J. Hadamard. "Cinq lettres sur la theorie des ensembles." Bulletin Societe de France, 33 (1905), 261-273.

Bar Hillel, Yehoshua. "An Examination of Information Theory." Philosophy of Science, 22 (1955), 86-105.

Beach, F. A., et. al. Ed. The Neurophysiology of Lashley: Selected Papers of K. S. Lashley. New York, 1960.

Beck, Lewis White. "Neo-Kantianism." Encyclopedia of Philosophy. 1967.

Beckner, Morton O. "Mechanism in Biology." Encyclopedia of Philosophy. 1967.

Belinfante, H. J. "Der Levysche Umordungssatz und seine intuitionische Übertragung." Composito Math., 6 (1938), 124-135.

----------. "Das Riemannische Umordnungsprinzip in der intuitionischen Theorie der unendlichen Reihen." Composito Math., 6 (1938), 118-123.

----------. "Über eine besondere Klasse von non-oszillierenden Reihen." Proceedings of the Amsterdam Society, 33 (1930), 1170-1179.

----------. "Zur intuitionischen Theorie der unendlichen Reihen." Sitz. Berlin, 1929, XXIX.

Bell, E.T. Development of Mathematics. 2nd ed. New York, 1945.

Benacerraf, Paul and Hilary Putnam. Readings in the Philosophy of Mathematics. Englewood Cliffs, N. J., 1964.

Berkeley, E. C. The Computer Revolution. Garden City, N. Y., 1962.

----------. Giant Brains. New York, 1949.

Bernkopf, M. "Ivar Fredholm." Dictionary of Scientific Biography . 1969.

Bernstein, Jeremy. Experiencing Science. New York, 1978.

Bershteyn, N. A. "New Lines of Development in Physiology and their Relation to Cybernetics." Problems of Philosophy, 8 (1962), 78-87.

Beth. E. W. "L' evidence intuitive dans les mathematiques modernes." In Travaux du IXe Congres International de Philosophie, VI, Logique et mathematiques. Paris, 1937.

----------. The Foundations of Mathematics. Amsterdam, 1968.

----------. "De Significa van de pasigrafische systemen." Euclides, 13 (1936-7), 145-158.

---------- and Jean Piaget. Mathematical Epistemology and Psychology. Dordrecht, Holland, 1966.

Beurle, R. L. "Properties of a Mass of Cells Capable of Regenerating Pulses." Philosophical Transactions of the Royal Society of London, Series B, 240 (1956), 55.

Bianchi, L. The Mechanism of the Brain and the Function of the Frontal Lobes. Edinburgh, 1922.

Billing, H. Lernende Automaten. Munich, 1961.

Bishop, Errett. "The Crisis in Contemporary Mathematics." Historia Mathematica, 2 (1975), 507-517.

----------. Foundations of Constructive Analysis. New York, 1967.

Black, Max. Problems of Analysis. Ithaca, New York, 1954.

Blake, D. V. and Uttley, A. M. Ed. Proceedings of a Symposium on Mechanization of Thought Processes. 2 vol. London, 1959.

Boltzmann, Ludwig. Vorlesungen über Gas Theorie. 2 vol. Leipzig, 1896, 1898.

Booth, Andrew D. "The Future of Automatic Digital Computers." Communications of the Association for Computing Machinery, 3 (June, 1960), 339-341, 360.

Borel, Emile. "I. Aggregates of Zero Measure." Rice Institute Pamphlet, 4th ser., 1 (1917), 1-52.

----------. L'espace et le temps. Paris, 1922.

----------. Lecons sur la theorie des fonctiones. Paris, 1898. (4th ed., 1950).

----------. Les nombres inaccessibles. Paris, 1952.

----------. "Allocution." Notices et discours de l'Academie des Sciences, 2 (1949), 350-359.

----------. "Quelques remarques sur les ensembles de droites ou de plans." Bulletin de la Societe Mathematique de France, 31 (1903), 272-275.

Borel, Emile. "Quelques remarques sur les principes de la theorie des ensembles." Mathematische Annalen, 60 (1905), 119-126.

----------. "Remarque relative a la communication de M. Hadamard." Verhandlungen des Ersten Internationalen Mathematiker-Kongresses, 204-205. Leipzig, 1898.

----------. "Sur quelque points de la theorie des fonctions." Annales de l'Ecole Normale, 3rd. ser., 12 (1895), 9-55.

----------. Selecta. Jubile scientifique de M. Emile Borel. Paris, 1940.

Boyer, Carl B. The History of the Calculus and Its Conceptual Development. New York, 1949.

----------. A History of Mathematics. New York, 1968.

Brainerd, J. G. and T. K. Sharpless. "The ENIAC." Electrical Engineering, 67 (1948), 163-172.

Brillouin, Leon. Science and Information Theory. New York, 1956.

Broad, C. D. The Mind and Its Place in Nature. London, 1925.

Brouwer, L. E. J. "Consciousness, Philosophy, and Mathematics." Proceedings of the Tenth International Congress of Philosophy. Amsterdam, 1940.

----------. "Historical Background, Principles and Methods of Intuitionism." South African Journal of Science, (October-November, 1952), 346-357.

----------. "Intuitionism and Formalism." In Paul Benacerraf and Hilary Putnam, Readings in the Philosophy of Mathematics. Englewood Cliffs, N. J., 1964.

----------. "Over de Grondslagen der Wiskunde." Doctoral dissertation. University of Amsterdam, 1907.

Büchner, Ludwig. Force and Matter. Trans. by J. Frederick Collingwood. London, 1884.

Buhl, A. and G. Bouligand. "En memoire de Rene Baire." L'enseignement mathematique, 31 (1932), nos. 1-3.

Bunge, M. "Do Computers Think?" British Journal for the Philosophy of Science, 7 (1956), 1- 13.

Burali-Forti, Cesare. Rendiconti del Circolo Matematico di Palermo, 11 (1897), 154-164, 260.

Burks, Arthur. "Computation, Behavior, and Structure in Fixed and Growing Automata." Behavioral Science, 6 (1961), 5-22.

----------. "Electronic Computing Circuits of the ENIAC." Proceedings of the Institute of Radio Engineers, 35 (August, 1947), 756-767.

----------. "Programming and the Theory of Automata." In P. Braffort and D. Hirschberg, ed., Programming and Formal Systems. New York, 1963.

----------. "Super Electronic Computing Machines." Electronic Industries, 5 (1946), 62-67, 96.

----------, and Hao Wang. "The Logic of Automata." Journal of the Association of Computing Machinery, 4 (1957), 193-218, 279-297.

----------, and Jesse Wright. "Theory of Logical Nets." Proceedings of the Institute of Radio Engineers, 41 (1953), 1357-1365.

Burnham, W. H. "Memory Historically and Experimentally Considered." American Journal of Psychology, 2 (1888-1889).

Cajori, Florian. "The History of Zeno's Arguments on Motion." American Mathematical Monthly, 22 (1915), 1-6, 39-47, 77-82, 109-115, 143-149, 179-186, 253-258, 292-297.

Calder, Allan. "Constructive Mathematics." Scientific American, Oct. 1979, 146-171.

Campbell, D. "Why is Britain's Wartime Code-Breaking Still Secret?" New Scientist, 73 (1977), 40?.

Campbell, Keith. "Materialism." Encyclopedia of Philosophy, 5, 179-188.

Campbell-Kelly, Martin. "Programming the Mark I: Early Programming Activity at the University of Manchester." Annals of the History of Computing, 2 (1980), 130-168.

Cantor, Georg. "Beiträge zur Begrundung der transfiniten Mengenlehre." Part I. Mathematische Annalen, 46 (1895), 236-291.

----------. Contributions to the Founding of the Transfinite Numbers. Trans. by P. E. B. Jourdain. Chicago, 1915.

----------. Gesammelte Abhandlungen mathematischen und philosophischen Inhalts. Berlin, 1932.

----------. "Über eine Eigenschaft des Inbegriffes aller reellen algebraischen Zahlen." Journal für die Reine und Angewandte Mathematik, 77 (1895), 258-262.

----------. "Über eine elemantare Frage der Mannigfaltigkeitslehre." Jahresbericht der Deutschen Mathematiker-Vereinigung, I, 75-78.

----------. "Über unendliche, lineare Punktmannichfaltigkeiten." Mathematische Annalen, 21 (1883).

Carnap, Rudolf. "Psychology in Physical Language." In A. J. Ayer, ed., Logical Positivism. Glencoe, Illinois, 1959.

Carpenter, B. E. and R. W. Doran. "The Other Turing Machine." Computer Journal, 20 (1977).

Carpenter, W. B. Principles of Mental Physiology. London, 1874.

Carroll, Charles Michael. The Great Chess Automaton. New York, 1965.

Cassirer, Ernst. Philosophy of Symbolic Forms. 3 vol. New York, 1955-1957.

Cave Brown, Anthony. Bodyguard of Lies. New York, 1975.

Cayley, A. Journal für die Reine und Angewandte Mathematik, 50 (1855), 282-285.

Chapuis, Alfred E. and Edmond Droz. Automata: Historical and Technological Study. Neuchatel, 1958.

Chapuis, Alfred E. and Edouard Gelis. Le Monde des Automates. Paris, 1928.

Cherry, E. C. "A History of the Theory of Information." Proc. Inst. Electrical Engineers, 98 (1951), 383.

----------. On Human Communication. Cambridge, Mass., 1957.

----------, ed. Third London Symposium on Information Theory. London, 1955.

Church, Alonzo. "Applications of Recursive Arithmetic to the Problem of Circuit Synthesis." Summaries of Talks Presented at the Summer Institute for Symbolic Logic. Cornell University, 1957. Princeton, 1960.

----------. "Logic." Encyclopedia Britannica. 14th ed.

----------. "A Note on the Entscheidungsproblem." Journal of Symbolic Logic, 1 (1936), 40-41.

----------. "Correction to a Note on the Entscheidungsproblem." Journal of Symbolic Logic, 1 (1936), 101-102.

----------. "A Set of Postulates for the Foundation of Logic." Annals of Mathematics, 2 s., 33 (1932), 346-366 and 34 (1933), 839-864.

----------. "An Unsolvable Problem of Elementary Number Theory." American Journal of Mathematics, 58 (1936), 345-363.

----------, and S. Kleene. "Formal Definitions in the Theory of Ordinal Numbers." Fundamenta Mathematicae, 28 (1937), 11-21.

----------, and J. B. Rosser. "Some Properties of Conversion." Transactions of the American Mathematical Society, 39 (1936), 472-482.

Cohen, Jonathan. "Can There Be Artificial Minds?" Analysis, 16 (1955), 36-41.

Collingwood, E. F. "Borel." Journal of the London Mathematical Society, 34 (1959), 488-512, and 35 (1960), 384.

Copi, I. M., C. C. Elgot, and J. B. Wright. "Realization of Events by Logical Nets." Journal of the Association for Computing Machinery, 5 (1958), 181-196.

Costabel, Pierre. "Rene Louis Baire." Dictionary of Scientific Biography, I, 406-408.

Crowe, Michael. A History of Vector Analysis. Notre Dame, Indiana, 1967.

Curry, H. B. "Grundlagen der kombinatorischen Logik." American Journal of Mathematics, 52 (1930), 509-536, 789-834.

Dauben, Joseph. Georg Cantor. Cambridge, Mass., 1979.

Davis, Martin. Computability and Unsolvability. New York, 1958.

Dechert, Charles R., ed. The Social Impact of Cybernetics. Notre Dame, Indiana, 1966.

DeMorgan, Augustus. Trigonometry and Double Algebra. London, 1849.

Diebold, John. The World of the Computer. New York, 1973.

Dieudonne, Jean. "Jules Henri Poincare." Dictionary of Scientific Biography, 11, 51-61. New York, 1975.

Dijkman, J. G. "Convergentie en divergentie in de intuitionistische wiskunde." Thesis. University of Amsterdam, 1952.

----------. "Einige Sätze über mehrfach negativ-konvergente Reihen in der intuitionischen Mathematik." Proc. Amsterdam Soc., 49 (1946), 829-833.

----------. "A Note on Intuitionistic Divergence Theory." N. Archief v. Wiskunde, (3) 10 (1962), 17-19.

Eccles, J. C., The Neurophysiological Basis of Mind. Oxford, 1953.

----------. "The Physiology of the Imagination." Scientific American, 199 (1958), 135-146.

Eilenberg, Samuel. Automata, Languages, and Machines. A. New York, 1974.

Eliot, Hugh. Modern Science and Materialism. London, 1919.

Esenin-Volpin, A. S. "Le programme ultra-intuitioniste des fondements des mathematiques." Infinitistic Methods, Warsaw, 1961.

----------. "The Ultra-intuitionistic Criticism and the Antitraditional Program for Foundations of Mathematics." In Kino-Myhill-Vesley, Intuitionism and Proof Theory. Buffalo, New York, 1968.

Estrin, Gerald. "The Electronic Computer at the Institute for Advanced Study." Mathematical Tables and Other Aids to Computation, 7 (1953), 108-114.

Euclid. The Elements. Trans. T. L. Heath. 3 vol. London, 1930.

Farley, B. G., "Self-Organizing Models for Learned Perception." In Self-Organizing Systems. New York, 1960.

Fefferman, S. "Autonomous Transfinite Progressions and the Extent of Predicative Mathematics." In B. van Rootselaar and J. F. Staal, eds., Logic, Methodology and Philosophy of Science, III. Amsterdam, 1968.

----------. "On the Strength of Ordinal Logics." Journal of Symbolic Logic, 23 (1958), 105-106.

----------. "Ordinal Logics--Reexamined." Journal of Symbolic Logic, 23 (1958).

----------. "Transfinite Recursive Progressions of Axiomatic Theories." Journal of Symbolic Logic, 27 (1962), 259-316.

Feigl, Herbert. "The 'Mental' and the 'Physical.'" In Minnesota Studies in the Philosophy of Science. II. Minneapolis, 1958.

Feindel, William. Memory, Learning, and Language. Toronto, 1959.

Feinstein, Amiel. Foundation of Information Theory. New York, 1958.

Feyerabend, Paul K. "Materialism and the Mind-Body Problem." Review of Metaphysics, 17 (1963), 49-66.

Finsler, P. "A propos de la discussion sur les fondements des mathematiques." In Entretiens I: Les Entretiens de Zurich sur les fondements et la methode des sciences mathematiques. Zurich, 1941.

Fitch, Frederick B. "Systems of 'Complete Logic' not within the Scope of Godel's Theorem." Journal of Symbolic Logic, 2 (1937), 63.

Fraenkel, A. "Georg Cantor." Jahresbericht der Deutschen Mathematiker-Vereinigung, 39 , 189-266.

----------, and Y. Bar-Hillel. Foundations of Set Theory. New York, 1963.

----------, and P. Bernays. Axiomatic Set Theory. New York, 1968.

Frechet, Maurice. L'arithmetique de l'infini. Paris, 1934.

----------. "Sur quelques points du calcul fonctionnel." Rendiconti del Cicolo Matematico di Palermo, 22 (1906), 1-74.

----------. "La vie et l'oeuvre d'Emile Borel." Enseignement mathematique, 2nd ser., 11 (1965), 1-95.

Fredholm, Ivar. "Sur une classe d'equations fonctionelles." Acta Mathematica, 27 (1903), 365-390.

----------. Oeuvres completes de Ivar Fredholm. Malmö, 1955.

Freudenthal, Hans. "David Hilbert." Dictionary of Scientific Biography, VI, 388-395.

Gandy, Robin. "Alan M. Turing." Nature, 174 (1954), 535-536.

----------. "The Simple Theory of Types." In Robin O. Gandy and J. M. E. Hyland, eds., Logic Colloquium 76. Amsterdam, 1977.

George, F. H. The Brain as a Computer. London, 1962.

----------. "Logic and Behavior." Science News, 45 (1957), 46-60.

Gibbs, J. Willard. "Elements of Vector Analysis." Privately published pamphlet. New Haven, Ct., 1881.

----------, and E. B. Wilson. Vector Analysis. New Haven, Ct., 1901.

Gleiser, M. "The Curious Life of Alan Turing." Computer Decisions, August, 1976, 30, 31, 34, 36.

Gödel, Kurt. "On Undecidable Propositions of Formal Mathematical Systems." Institute for Advanced Study pamphlet. Princeton, N. J., 1934.

----------. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I." Monatshefte für Mathematik und Physik, 38 (1931), 173-198.

----------. "Über die Länge der Beweis." Ergebnisse eines mathematischen Kolloquiums, 7, 23-24.

Goldstine, Herman H. The Computer from Pascal to von Neumann. Princeton, 1972.

----------, and Adele Goldstine. "The Electronic Numerical Integrator and Computer (ENIAC." Mathematical Tables and Other Aids to Computation, 2 (1946), 97-110.

Gonseth, F. "Qu'est-ce que la logique?" Actualites scientifiques et industrielles 524. Paris, 1937.

Good, I. J. "Early Work on Computers at Bletchley." Annals of the History of Computing, 1 (1979).

----------. "Logic of Man and Machine." New Scientist, 26 (1965).

Grattan-Guiness, Ivar. "The Correspondence between Georg Cantor and Philip Jourdain." Jahresbericht der Deutschen Mathematiker-Vereinigung, 73, 111-130.

----------. "Towards a Biography of Georg Cantor." Annals of Science, 1973, 345-391.

Graves, Rev. Robert Perceval. The Life of Sir William Rowan Hamilton. 3 vol., Dublin, 1882-1891.

Grelling, Kurt. "Der Einfluss der Antinomien auf die Entwicklung der Logik im 20.Jahrhundert." Travaux du IXe Congres Internationale de Philosophe, VI Logique et mathematiques, Actualites scientifiques et industrielles 535. Paris, 1937.

----------. "Gibt es Gödelsche Antinomie?" Theoria, 3 (1937), 297-306.

Greniewsky, H. Cybernetics Without Mathematics. New York, 1960.

Gross, Maurice, and Andre Lentin. Notions sur les grammaires formelles. Paris, 1970.

Grunbaum, Adolf. Philosophical Problems of Space and Time. New York, 1963.

__________ Modern Science and Zeno's Paradoxes. Middletown, Ct., 1967.

Guilbaud, G. T. What is Cybernetics? London, 1959.

Hadamard, Jacques. "Sur certaines applications possible de la Theorie des Ensembles." Verhandlungen des Ersten Internationalen Mathematiker-Kongresses, 201-202. Leipzig, 1898.

Haldane, J. S. Mechanism, Life and Personality. 2nd ed., New York, 1923.

Halsbury, Earl of. "Ten Years of Computer Development." Computer Journal, 1 (1959), 153-159.

Hamilton, Sir William Rowan. Lectures on Quaternions. Dublin, 1853.

----------. "Theory of Conjugate Functions, or Algebraic Couples; with a Preliminary and Elementary Essay on Algebra as the Science of Pure Time." Proceedings of the Royal Irish Academy, 17 (1837), 293-422.

Harrow, Keith. "Theoretical and Applied Computer Science: Antagonism or Symbiosis?" American Mathematical Monthly, 86 (1979), 253-260.

Hartley, R. V. L. "Transmission of Information." Bell System Technical Journal, 7 (1928), 535.

Hartree, D. R. Calculating Instruments and Machines. Urbana, Illinois, 1946.

Hatfield, H. Stafford. Automaton: Or the Future of the Mechanical Man. London, 1928.

Hausdorff, Felix. "Gründzuge einer Theorie der geordneten Mengen." Mathematische Annalen, 65 (1908), 435-505.

Hawkins, Thomas. "Henri Leon Lebesgue." Dictionary of Scientific Biography, 8 (1973), 110-112.

----------. Lebesgue's Theory of Integration. Madison, Wi., 1970.

Heath, Thomas L. A History of Greek Mathematics. Oxford, 1921.

----------. A Manuel of Greek Mathematics. Oxford, 1931.

----------. Mathematics in Aristotle. Oxford, 1949.

Heaviside, Oliver. Electromagnetic Theory. 3 vol. London, 1893, 1899, 1912.

Hellinger, Ernst and Otto Toeplitz. "Integralgleichungen und Gleichungen mit unendlichvielen Unbekannten." Encyklopädie der Mathematische Wissenschaften. Leipzig, 1923-1927.

Helmer, Olaf. "Perelman versus Gödel." Mind, 46 (1937), 58-60.

----------. "The Significance of Undecidable Sentences." The Journal of Philosophy, 34 (1937), 490-494.

Hempel, Carl G. "The Logical Analysis of Psychology." Revue de Synthese, 10 (1935).

Herbrand, Jacques. Logical Writings. Ed. by D. Goldfarb. Cambridge, Mass., 1971.

----------. "Sur la non-contradiction de l'arithmetique." Journal für die reine und angewandte Mathematik, 166 (1931-32), 1

Hering, E. On Memory and the Specific Energies of the Nervous System. Chicago, 1896.

Hermes, Hans. Enumerability, Decidability, Computability. New York, 1969.

----------, and H. Scholz. "Mathematische Logik." Encyk. d. Math. Wissen., I, 82.

Hilbert, David. "Axiomatische Denken." Mathematische Annalen, 78 (1918), 405-415.

----------. Gesammelte Abhandlungen. Berlin.

----------. Grundlagen der Geometrie. Berlin, 1899.

----------. "Die Grundlagen der Mathematik." Abhandlungen aus dem mathematische Seminar der Hamburgischen Universitat, 6 (1928), 65-85.

----------. Grundzüge einer allgemeinen Theorie der linearen Integralgleichungen. Berlin, 1912.

----------. "Neubegrundung der Mathematik, Erste Mitteilung." Abhandlungen Mathematische Seminar der Hamburger Universitat, 1 (1922), 157-177.

----------. "Uber das unendliche." Mathematische Annalen, 95 (1926), 161-190.

----------, and W. Ackermann. Grundzüge der theoretischen Logik. Berlin, 1928.

Hölder, Otto. Die Aritmetik in Strenger Begrundung. Leipzig, 1914.

----------. Die Mathematische Methode. Berlin, 1924.

Holt, E. B. The Freudian Wish and Its Place in Ethics. New York: 1915.

Hook, Sidney, ed. Dimensions of Mind: A Symposium. New York, 1960.

Hooke, Robert. "An Hypothetical Explanation of Memory: How the Organs Made Use of by the Mind in its Operation May Be Mechanically Understood." In The Posthumous Works of Dr. Robert Hooke. London, 1705.

Holmes, Oliver Wendell. Mechanism in Thought and Morals. 2nd ed. London, 1871.

Hopcroft, John, and Jeffrey Ullman. Formal Languages and Their Relation to Automata. Reading, Mass., 1969.

Hopf, E. "Ergodentheorie." Ergebnisse der Mathematik und Ihrer Grenzgebiete, 5.

Hu, Sze-Tsen. Mathematical Theory of Switching Circuits and Automata. Berkeley, Ca., 1968.

Hunter, I. M. L., Memory. London, 1957.

Jacker, Corinne. Man, Memory, and Machines. New York, 1964.

Jackson, W., ed., Communication Theory. London, 1952.

Jaki, Stanley L. Brain, Mind and Computers. New York, 1969.

James, H. M., N. B. Nichols, and R. S. Phillips. Theory of Servo-Mechanisms. Cambridge, Mass., 1947.

James, William. The Principles of Psychology. New York, 1890.

Jefferson, G. "The Mind of Mechanical Man." British Medical Journal, 1 (1949), 1105-1121.

Jeffress, L. A., ed. Cerebral Mechanisms in Behavior: The Hixon Symposium. New York, 1951.

Johnson, Brian. The Secret War. London, 1978.

Jones, R. V. "The Secret War." New Scientist, 73 (1977), 480.

Jorgensen, J. The Development of Logical Empiricism. Chicago, 1951.

Kahn, David. The Codebreakers. New York, 1967.

Kalmar, Laszlo. "Zur Reduktion des Entscheidungsproblem." Norsk matematisk tidsskrift, 19 (1937), 121-130.

Keister, H., et. al. The Design of Switching Circuits. New York, 1951.

Kemeny, John. "Man Viewed as a Machine." Scientific American, 192 (1955), 58-67.

King, H. R. "Aristotle and the Paradoxes of Zeno." Journal of Philosophy, 46 (1949), 657-670.

Kleene, S. C. "General Recursive Functions of Natural Numbers." Mathematische Annalen, 112 (1935-6), 727-742.

----------. Introduction to Metamathematics. Amsterdam, 1952.

----------. "Lambda-definability and Recursiveness." Duke Mathematical Journal, 2 (1936), 340-353.

----------. Mathematical Logic. New York, 1967.

----------. "A Note on Recursive Functions." Bulletin of the American Mathematical Society, 42 (1936), 544-546.

----------. "On Notation for Ordinal Numbers." The Journal of Symbolic Logic, 3 (1938), 150-155.

----------. "Recursive Predicates and Quantifiers." Transactions of the American Mathematical Society, 53 (1943), 41-73.

----------. "Representation of Events in Nerve Nets and Finite Automata." In Claude Shannon and J. McCarthy, eds., Automata Studies, 3-42. Princeton, N. J., 1956.

----------. "A Theory of Positive Integers in Formal Logic." American Journal of Mathematics, 57 (1934), 231.

Kline, Morris. Mathematical Thought from Ancient to Modern Times. New York, 1972.

Kneale, William, and Martha Kneale. The Development of Logic. Oxford, 1962.

Knuth, Donald. The Art of Computer Programming. 2nd ed. New York, 1975.

König, Jules. "Zum Kontinuum-Problem." Verhandlungen des Dritten Internationalen Mathematiker-Kongresses, 1905, 144-147. Leipzig, 1905.

Kreisel, G. "Ordinal Logics and the Characterization of Informal Concepts of Proof." In Proceedings of the International Congress of Mathematicians, 1958, 289-299. Cambridge, 1960.

Kronecker, Leopold. "Über die Zahlbegriff." Journal für die reine und angewandte Mathematik, 101 (1887), 337-355.

Lacey, A. R. "Men and Robots." Philosophical Quarterly, 10 (1960), 61-72.

Ladriere, Jean. Les Limitations Internes des Formalism. Paris, 1957.

Lange, F. A. The History of Materialism. 3rd ed. London, 1925.

Larguier, Everett H. "The Schools of Thought in Modern Mathematics." Thought, 12 (1937), 225-240.

Lashley, K. S. "The Behavioristic Interpretation of Consciousness." Psychological Review, 30 (1923), 237-272, 329-353.

----------. Brain Mechanisms and Intelligence. Chicago, 1929.

----------. "Coalescence of Neurology and Psychology." Proceedings of the American Philosophical Society, 84 (1941), 461-470.

----------. "Persistent Problems in the Evolution of Mind." Quarterly Review of Biology, 24 (1949), 28.

Laslett, P., ed. The Physical Basis of Mind. Oxford, 1950.

Latil, Pierre de. Thinking by Machine. Trans. by Y. M. Golla. London, 1956.

Lavington, S. A History of Manchester Computers. Manchester, 1975.

Lebesgue, Henri. "Integrale, longeur, aire." In Annali di matematica pura ed applicata, 3rd. ser., 7 (1902), 231-359.

----------. Lecons sur l'integration et la recherche des fonctions primitives. Paris, 1904.

----------. Lecons sur les series trigonometrique. Paris, 1906.

----------. Measure and the Integral. San Francisco, 1966.

----------. Notice sur les travaux scientifiques de M. Henri Lebesgue. Toulouse, 1922.

----------. "Sur l'integration des fonctions discontinues." Annales scientifiques de l'Ecole normale superieure, 3rd ser., 27 (1910), 361-450.

----------. "Sur le mesure des grandeurs." Enseignement mathematique, 31-34 (1933-1936).

Lee, H. D. P. Zeno of Elea. Cambridge, 1936.

Levy, David. Chess and Computers. Potomac, Md., 1976.

Lewin, Ronald. Ultra Goes to War. London, 1978.

Lindemann, Ferdinand. Mathematische Annalen, 20 (1882), 213-225.

Lindsay, R. B. "A Scientific Analogy: The Thermodynamic Imperative." In The Role of Science in Civilization, 290-298. New York, 1963.

Lindsley, D. B. "Electroencephalography." In J. M. Hunt, Personality and the Behavior Disorders, II, 1033-1103. New York, 1944.

Locher, L. "Die Finsler'sche Arbeiten zur Grundlegung der Mathematik." Comentarii mathematici Helvetici, 10 (1937-8), 206-207.

Loeb, Jacques. Der Heliotropismus der Tiere. Paris, 1890.

Lucas, J. R. "Minds, Machines, and Gödel." Philosophy, 36 (1961), 112-127.

Lytel, Allan. Digital Computers in Automation. Indianapolis, Indiana, 1966.

MacKay, D. M. "Mentality in Machines." Proceedings of the Aristotelean Society, Supp. Vol. 26 (1952), 61-86.

----------. "Mindlike Behavior in Artefacts." British Journal for the Philosophy of Science, 2 (1951), 31-56.

Malik, R. And Tomorrow the World. London, 1975.

----------. "In the Beginning--Early Days with ACE." Data Systems, March, 1969, 56-59, 82.

----------. "Only Begetters of the Computer." New Scientist, 47, 710 (16 July 1970), 138-139.

Maloney, Russell. "Inflexible Logic." New Yorker, 3 Feb. 1940.

Manheim, Jerome. The Genesis of Point Set Topology. New York, 1964.

Maxwell, James Clerk. "On Governors." Proceedings of the Royal Society, XVI, 270-283. London, 1868.

----------. A Treatise on Electricity and Magnetism. London, 1873.

May, Kenneth O. Bibliography and Research Manual of the History of Mathematics. Toronto, 1973.

----------. "Emile Borel." Dictionary of Scientific Biography. New York, 1970.

Mays, W. "The Hypothesis of Cybernetics." British Journal for the Philosophy of Science, 2 (1951), 38-51.

Maziarz, Edward A. and Thomas Greenwood. Greek Mathematical Philosophy. London, 1968.

McCallum, M., and J. B. Smith. "Mechanized Reasoning." *Electronic Engineering*, April, 1951.

McColl, L. *Servomechanisms*. New York, 1945.

McCorduck, Pamela. *Machines Who Think*. San Francisco, 1979.

McCulloch, Warren S. "The Design of Machines to Simulate the Behavior of the Human Brain." *IRE National Convention*, 1955.

----------. *Embodiments of Mind*. New York, 1967.

----------. "A Logical Calculus of the Ideas Immanent in Nervous Activity." *Bull. Math. Biophysics*, 5 (1943), 115-133.

----------, and J. E. Pfeiffer. "Of Digital Computers Called Brains." *Scientific Monthly*, 49 (1949), 368-376.

McDougall, W. *Physiological Psychology*. London, 1913.

McKeon, Richard. *The Basic Works of Aristotle*. New York, 1941.

Meschkowski, Herbert. *Probleme des Unendlichen: Werk und Leben Georg Cantors*. Braunschweig, 1967.

Michie, D. "The Disaster of Alan Turing's Buried Treasure." *Computer Weekly*, 3 March 1977, 10.

Minsky, Marvin L. "Heuristic Aspects of the Artificial Intelligence Problem." Group Report 34-55; Massachusetts Institute of Technology, Lincoln Laboratory, Lexington, MA., December, 1956.

----------. "A Selected Descriptor-Indexed Bibliography to the Literature on Artificial Intelligence." *IRE Transactions on Human Factors in Electronics*, 49 (1961), 39-55.

----------. "Steps Toward Artificial Intelligence." *Proceedings of the IRE*, 49 (1961), 8-30.

Moleschott, Jakob. *Der Kreislauf des Lebens*. Mainz, 1852.
ings of the IRE
Monne, L. "Structure and Function of Neurons in Relation to Mental Activity." *Biological Reviews and Biological Proceedings*, Cambridge Philosophical Society, 24 (1949), 18-37.

Moore, Gregory. "The Origins of Zermelo's Axiomatization of Set Theory." Journal of Philosophical Logic, 7 (1978).

Morgan, Clifford T. Physiological Psychology. New York, 1943.

Murphy, Gardner. Historical Introduction to Modern Psychology. New York, 1949.

Nagel, Ernest. "Are Naturalists Materialists?" In Logic Without Metaphysics. Glencoe, Illinois, 1957.

Newell, A. and H. A. Simon. The Simulation of Human Thought. Report No. P-1734, The RAND Corporation. Santa Monica, CA., 1959.

Newman, M. H. A. "Alan Mathison Turing." Biographical Memoirs of Fellows of the Royal Society, I (1955), 253-263.

----------, and Alan Turing. "A Formal Theorem in Church's Theory of Types." Journal of Symbolic Logic, 7 (1942), 28.

Nyquist, H. "Certain Factors Affecting Telegraph Speed." Bell System Technical Journal, 3 (1924), 324-354.

----------. "Certain Topics in Telegraph Transmission." A. I. E. E. Transactions, 47 (1928), 617.

Oettinger, A. G. "Programming a Digital Computer to Learn." Philosophical Magazine, 43 (1952), 1243-1263.

Owen, G. E. L. "Zeno and the Mathematicians." Proceedings of the Aristotelean Society, 58 (1957-8), 199-222.

Parsons, Charles. "Mathematics, Foundations of." Encyclopedia of Philosophy, 1967.

Pasch, M. and M. Dehn. Vorlesungen über neuere Geometrie. 2nd ed. Berlin, 1926.

Passmore, John. A Hundred Years of Philosophy. London, 1957.

Pavlov, I. P. Conditioned Reflexes: An Investigation of the Physiological Activity of the Cerebral Cortex. Trans. by G. V. Anrep. London, 1927.

Peano, Guiseppe. "Additione." Revistade mathematica, 8 (1906), 136-143.

----------. Arithmetices Principia. Turin, 1889.

----------. Formulaire de mathematiques. Turin, 1895-1901.

----------. Opera scelte. 3 vol. 1957-1959.

----------. Sui fondamenti della geometria. Turin, 1894.

Pepis, Jozef. "Ein Verfahren der mathematischen Logik." Journal of Symbolic Logic, 3 (1938), 61-76.

----------. "Beiträge zur Reduktiostheorie des logischen Entscheidungsproblem." Acta literarium ac scientiarum Regiae Universitatis Hungaricae Francisco-Iosephinae, Sectio scientiarum mathematicarum. 8 (1936-7), 7-41.

----------. "Untersuchungen über das Entscheidungsproblem der mathematischen Logik." Fundamenta mathematicae, 3C (1938), 257-348.

Perelman, C. "L'antinomie de M. Gödel." Academie royale de Belgique, Bulletin de la classe des sciences, 5 ser., 22 (1936), 730-736.

----------. "Une solution des paradoxes de la logique et ses ses consequences pour le conception de l'infini." Travaux du IXe Congres International de Philosophe, VI Logique et mathematiques, 206-210. Paris, 1937.

Peter, Rozsa. "Über die mehrfach Rekursion." Mathematische Annalen, 113 (1936), 489-527.

Place, U. T. "Is Consciousness a Brain Process?" British Journal of Psychology, 47 (1956), 44-50.

Poincare, Henri. "L'Avenir des mathematiques." Atti del IV Congresso Internazionale dei Matematici. Rome, 1908.

----------. "On the Foundations of Geometry." Monist, 9 (1898).

Post, Emil. "Finite Combinatory Processes--Formulation 1." Journal of Symbolic Logic, 1 (1936), 103-105.

Post, Emil. "Recusively Enumerable Sets of Positive Integers and Their Decision Problems." Bulletin of the American Mathematical Society, 50 (1944), 284-316.

Poynter, F. N. L. The Brain and its Functions. Oxford, 1958.

Rabin, M. O. and D. Scott. "Finite Automata and their Decision Problems." IBM J. Res. Dev., 3 (1959), 114-125.

Randall, Brian. The Colossus. Report No. 90, Computing Laboratory, University of Newcastle upon Tyne. June, 1976.

----------. "Colossus: Godfather of the Computer." New Scientist, 73, 1038 (10 Feb. 1977), 346-348.

----------. "On Alan Turing and the Origins of Digital Computing." In B. Meltzer and D. Michie, eds., Machine Intelligence, 7 (1972), 3-20.

----------, ed. The Origins of Digital Computers. New York, 1975.

Ranson, S. W. and S. L. Clark. The Anatomy of the Nervous System. Philadelphia, 1959.

Rashevsky, Nicholas. "Is the Concept of an Organism as a Machine a Useful One?" Scientific Monthly, 80 (1955), 32-35.

----------. "Outline of a Physico-Mathematical Theory of the Brain." Journal of General Psychology, 3 (1945), 82-112.

Reid, Constance. Hilbert. New York, 1970.

Reid-Green, Keith S. "A Short History of Computing." Byte, 3 (July, 1978), 84-94.

Richtmyer, R. D. "The Post-War Computer Development." American Mathematical Monthly, 72 (1965), 8-14.

Riesz, Frigyes. "Stetigkeit und Abstrakte Mengenlehre." Atti del IV Congresso Internazionale dei Matematici, 2, 18-24. Rome, 1909.

Rogers, Hartley. Theory of Recursive Functions and Effective Computability. New York, 1967.

Rohwer, Jurgen. "Notes on the Security of the German Decoding System." Trans. A. J. Barker. In The Critical Convoy Battles of March 1943. Annapolis, Md., 1977.

Rosen, Saul. "A Quarter Century View." Association for Computing Machinery pamphlet. 1971.

Rosenberg, Jerry M. The Computer Prophets. London, 1969.

Rosenblatt, F. "The Design of an Intelligent Automaton." In Research Reviews. Washington, Office of Naval Research publication. October, 1958.

Rosenblueth, Arturo, and Norbert Wiener. "Purposeful and Non-Purposeful Behavior." Philosophy of Science, 17 (1950), 318-326.

----------, Norbert Wiener, and Julian Bigelow. "Behavior, Purpose and Teleology." Philosophy of Science, 10 (1943), 18-24.

Ross, T. "The Synthesis of Intelligence—Its Implications." Psychological Review, 45 (1938), 185-189.

Rosser, J. Barkley. "Constructability as a Criterion for Existence." Journal of Symbolic Logic, 1 (1936), 68.

----------. "Extensions of Some Theorems of Gödel and Church." Journal of Symbolic Logic, 1 (1936), 87-91.

----------. "Gödel Theorems for Non-constructable Logics." Journal of Symbolic Logic, 2 (1937), 129-137.

----------. "A Mathematical Logic without Variables II." Duke Mathematical Journal, 1 (1935), 328-355.

Rothstein, Jerome. Communication, Organization, and Science. Indian Hills, CO., 1958.

Russell, Bertrand. The Autobiography of Bertrand Russell. 3 vol. London, 1967-1969.

----------. Foundations of Geometry. London, 1902.

----------. A History of Western Philosophy. New York, 1945.

Russell, Bertrand. "Mathematical Logic as Based on the Theory of Types." American Journal of Mathematics, 30 (1908), 222-262.

----------. "Mathematics and the Metaphysicians." In James R. Newmann, ed., The World of Mathematics, III, 1574-1596. New York, 1956.

----------. Our Knowledge of the External World. London, 1914.

----------. Principles of Mathematics. London, 1903.

----------, and A. N. Whitehead. Principia Mathematica. 3 vol. Cambridge, 1915-1917.

Ryle, Gilbert. The Concept of Mind. London, 1949.

----------. Dilemmas. Cambridge, 1954.

Schmitt, O. H. "The Design of Machines to Simulate the Behavior of the Human Brain." IRE National Convention, 1955.

Schoenflies, A. "Die Krisis in Cantor's mathematischen Schaffen." Acta Mathematica, 50, 1-23.

----------. Entwickelung der Mengenlehre und Ihrer Anwendungen. Leipzig, 1913.

Schuh, J. F. Principles of Automation. London, 1965.

Semon, R. The Mneme. Trans. by Louis Simon. London, 1921.

Shaffer, Jerome. "Mind-Body Problem." Encyclopedia of Philosophy, 1975.

Shannon, Claude. Automata Studies. Princeton, N.J.

----------. "Communication in the Presence of Noise." Proceedings of the Institute of Radio Engineers, 37 (1949), 10-21.

----------. "A Mathematical Theory of Communication." Bell System Technical Journal, 27 (1948), 379-423.

----------. "Von Neumann's Contributions to Automata Theory." Bulletin of the American Mathematical Society, 64 (1958), 123-129.

Shannon, Claude, and Warren Weaver. The Mathematical Theory of Communication. Urbana, Illinois, 1949.

Sherrington, C. The Integrative Action of the Nervous System. New Haven, Ct., 1906.

----------. Man on His Nature. Cambridge, 1953.

Simmons, P. L., and R. F. Simmons. The Simulation of Cognitive Behavior, II: An Annotated Bibliography. Report SP-590/002/00. System Development Corporation. Santa Monica, CA., 1961.

----------. "The Simulation of Cognitive Processes: An Annotated Bibliography." IRE Transactions on Electrical Computers, EC-10, 3 (1961), 462-483.

Singh, Jagjit. Great Ideas in Information Theory, Language, and Cybernetics. New York, 1966.

Sinnige, Theo. Matter and Infinity in the Pre-Socratics and Plato. Amsterdam, 1968.

Skinner, B. F. Science and Human Behavior. New York, 1953.

Skolem, Thoralf. "Einige Reduktion des Entschiedungsproblem." Avhandlinger utgitt av Det Norske Videnskaps-Akademie i Oslo, I. 6 (1936).

----------. "Über die Zurückfuhr barkeit einiger durch Rekursionen definierter Relationen auf 'arithmetische.'" Journal of Symbolic Logic, 1937, 85.

----------. Selected Works in Logic. Ed. by Jens Erik Fenstad. Oslo, 1970.

Smart, Harold. "Cassirer versus Russell." Philosophy of Science, 10 (1943), 167- 175.

Smart, J. J. C. "Gödel's Theorem, Church's Theorem, and Mechanism." Synthese, 13 (1961), 105-110.

----------. Philosophy and Scientific Realism. London, 1963.

----------. "Sensations and Brain Processes." Philosophical Review, 68 (1959), 141-156.

Smith, Thomas M. "Origins of the Computer." In M. Kranzberg and K. Purcell, Technology in Western Civilization. Vol. 2, Ch. 20.

Somenzi, Vittorio. "Information Theory." In Third London Symposium. Ed. by Colin Cherry. London, 1955.

Steen, S. W. P. Mathematical Logic. Cambridge, 1972.

Stewart, D. J., ed. Automaton Theory and Learning Systems. London, 1967.

Surma, Stanislaw. Studies in the History of Mathematical Logic. Warsaw, 1973.

Szilard, Leo. "The Decrease of Entropy in a Thermodynamic System by the Intervention of Intelligent Beings." Trans. in Behavioral Science, IX.

Taube, Mortimer. Computers and Common Sense: The Myth of Thinking Machines. New York, 1961.

Thatcher, James. "Universality in the von Neumann Cellular Model." Technical Report 03105-30-T, ORA, University of Michigan, 1965.

Thompson, D'Arcy. Growth and Form. London, 1945.

Thomson, James. "Infinity in mathematica and logic." Encyclopedia of Philosophy, 1975.

Tuller, W. G. "Theoretical Limitations on the Rate of Transmission of Information." Proceedings of the IRE, 37 (1949), 468-478.

Turing, Alan M. "The Chemical Basis of Morphogenesis." Philosophical Transactions of the Royal Society, B, 237 (1952), 37.

----------. "Computability and Lambda-definability." Journal of Symbolic Logic, 2 (1937), 153-163.

----------. "Computing Machinery and Intelligence." Mind, n. s., 59 (1950), 433-460.

Turing, Alan M. "Digital Computers Applied to Games: Chess." In B. V. Bowden, ed., Faster Than Thought, pp. 288-295. London, 1953.

----------. "Equivalence of Left and Right Almost Periodicity." Journal of the London Math. Society, 10 (1935), 284.

----------. "The Extensions of a Group." Comp. Math., 5 (1938), 357.

----------. "Finite Approximations to Lie Groups." Annals of Mathematics, 39 (1938), 105.

----------. "Intelligent Machinery." In B. Meltzer and D. Michie, ed., Machine Intelligence, V, 3-26. Edinburgh, 1969.

----------. "Intelligent Machinery: A Heretical Theory." In Sara Turing, Alan M. Turing, 128-134. Cambridge, 1955.

----------. "A Method for the Calculation of the Zeta Function." Proceedings of the London Mathematical Society, 2nd ser., 48 (1943), 180.

----------. "Proposals for Development in the Mathematics Division of an Automatic Computing Engine (ACE)." Report to the Executive Committee of the National Physical Laboratory, Teddington, England. 19 March, 1946.

----------. "On Computable Numbers, with an Application to the Entscheidungsproblem." Proceedings of the London Math. Society, ser. 2, 42 (1936-7), 230-265.

----------. On the Gaussian Error Function. King's College, Cambridge, Fellowship dissertation, 1935.

----------. "The p-function in Lambda-K-Conversion." Journal of Symbolic Logic, 2 (1937), 164.

----------. "Practical Forms of Type Theory." Journal of Symbolic Logic, 13 (1948), 80.

----------. "Programmer's Handbook for the Manchester Electronic Computer." Manchester, 1950.

----------. "Rounding-off Errors in Matrix Processes." Quarterly J. Mech. App. Math., 1 (1948), 287.

Turing, Alan M. "Solvable and Unsolvable Problems." Science News, 31 (1954), 7.

----------. "Some Calculations of the Riemann Zeta-Function." Proceedings of the London Math. Soc., ser. 3, 3 (1953), 99.

----------. "Systems of Logic Based on Ordinals." Proceedings of the London Math. Soc., ser. 2, 45 (1939), 161.

----------. "The Use of Dots as Brackets in Church's System." Journal of Symbolic Logic, 7 (1942), 146.

----------. "The Word Problem in Semi-groups with Cancellation." Annals of Mathematics, 52 (1950), 491.

Turing, Sara. Alan M. Turing. Cambridge, 1959.

Ulam, S. M. A Collection of Mathematical Problems. New York, 1960.

----------. "Electronic Computers and Scientific Research." In C. F. J. Overhage, ed., The Age of Electronics, 95-108. 1962.

----------. "John von Neumann." Bulletin of the American Mathematical Society, 64 (1958), 1-49.

Uttley, A. M. "The Classification of Signals in the Nervous System." E. E. G. Clin. Neurophysiol., 6 (1954), 479.

----------. Conditional Probability Computing in a Nervous System. N. P. L. Symposium. London, 1958.

----------. "Conditional Probability Machines and Conditioned Reflexes." In C. E. Shannon and J. McCarthy, ed., Automata Studies. Princeton, 1956.

----------. "The Probability of Neural Connections." Proc. of the Royal Society of London, ser. B, 144 (1955), 916.

Van der Waerden, B. L. Science Awakening. Amsterdam, 1954.

Van Heijenoort, Jean. From Frege to Gödel. Cambridge, Mass., 1967.

Van Rootselaar, B. "Luitzen Egbertus Jan Brouwer." Dictionary of Scientific Biography, 1970.

----------. "Alan Mathison Tur:ng." Dictionary of Scientific Biography, 1976.

----------. "Un probleme de M. Dijkman." Proc. Amsterdam, ser. A, 55 (1955), 405-407.

Veblen, O. "A System of Axioms for Geometry." Transactions of the American Mathematical Society, 5 (1904), 343-384.

Vogt, Karl. Köhlerglaube und Wissenschaft. Giessen, 1854.

----------. Physiologische Briefe. Stuttgart, 1845-6.

----------. Vorlesungen über die Menschen. Giessen, 1863.

Volterra, Vito. "Sopra le Funzioni che Dipendono da altre Funzioni." Atti della Reale Accademia dei Lincei Rendiconti, III, 97-104, 141-146. Rome, 1887.

----------. "Sopra le Funzioni da linee." Atti della Reale Accademia dei Lincei Rendiconti, III, 225-230, 274-281. Rome, 1887.

Von Bertalanffy, Ldwig. General System Theory. New York, 1968.

Von Foerster, H., ed. Cybernetics: Transactions of Conferences 1950-54. Josiah Macy, Jr. Foundation. New York, 1955.

Von Neumann, John. Collected Works. Ed. by A. H. Taub. New York, 1961-63.

----------. The Computer and the Brain. New Haven, Ct., 1958.

----------. First Draft of a Report on the EDVAC. University of Pennsylvania, June 30, 1945. Mimeographed.

----------. "The General and Logical Theory of Automata." In Cerebral Mechanisms in Behaviour--The Hixon Symposium. New York, 1951.

----------. Mathematische Grundlagen der Quantenmechanik. Trans. by Robert Beyer. Princeton, 1955.

Von Neumann, John. "Probabilistic Logics and the Synthesis of Reliable Organisms from Unreliable Components." In C. E. Shannon and J. McCarthy, ed., Automata Studies, 43-98. Princeton, 1956.

----------. Remarks on Ralph Gerard's "Some of the Problems Concerning Digital Notions in the Central Nervous System." In H. von Foerster, ed., Cybernetics, 19-31. New York, 1951.

----------. Review of Norbert Wiener's Cybernetics. Physics Today, 2 (1949), 33-34.

----------. Theory of Self-Reproducing Automata. Ed. by Arthur Burks. Urbana, Illinois, 1966.

----------, and Garrett Birkhoff. "The Logic of Quantum Mechanics." Annals of Mathematics, 37 (1936), 823-843.

----------, Arthur Burks, and H. H. Goldstine. Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. 1946. In Collected Works, V, 34-79.

----------, and H. H. Goldstine. "On the Principles of Large Scale Computing Machines." 1946. In Collected Works, V, 1-33.

----------, and H. H. Goldstine. Planning and Coding of Problems for an Electronic Computing Instrument. 3 vol., 1947-1948. In Collected Works, V, 80-235.

----------, and Oskar Morgenstern. Theory of Games and Economic Behavior. Princeton, 1944.

----------. "Zur Hilbertschen Beweistheorie." Mathematische Zeitschrift, 26 (1927), 1-46.

Walter, W. Grey. The Living Brain. London, 1953.

Wang, Hao. Logic, Computers, and Sets. New York, 1970.

Watson, J. B. Behaviorism. New York, 1925.

----------. "Is Thinking Merely the Action of Language Mechanisms?" British Journal of Psychology, 11 (1920), 87-104.

Watson, J. B. "Psychology as the Behaviorist Views It." Psych Review, 20 (1913), 158-177.

----------. Psychology from the Standpoint of a Behaviorist. New York, 1919.

Weiss, A. P. A Theoretical Basis of Human Behavior. New York, 1925.

Weizenbaum, Joseph. Computer Power and Human Reason. San Francisco, 1976.

Werkmeister, William Henry. "Seven Theses of Logical Positivism Critically Examined." The Philosophical Review, 46 (1937), 276-297, 357-376.

Weyl, Hermann. "David Hilbert and His Work." Bulletin of the American Mathematical Society, 50 (1944), 612-654.

Wiener, Norbert. Cybernetics: or Control and Communication in the Animal and the Machine. New York, 1948.

----------. "The Ergodic Theorem." Duke Mathematical Journal, 5 (1939), 235-267.

----------. The Extrapolation, Interpolation, and Smoothing of Stationary Time Series. New York, 1949.

----------. God and Golem, Inc. Cambridge, Mass., 1964.

----------. The Human Use of Human Beings. Boston, 1950.

----------. Selected Papers. Cambridge, Mass., 1964.

----------, and Arturo Rosenblueth. "Conduction of Impulses in Cardine Muscle." Arch. Inst. Cardiol. Mex., 16 (1946), 206-265.

----------, and J. P. Schade, ed. Symposium on Cybernetics of the Nervous System. Amsterdam, 1962.

Wigington, R. L. "A New Concept in Computing." Proceedings of the IRE, 47 (1949), 516-523.

Wilkes, M. V. "Automatic Computing Machines." Journal of the Roy. Soc. Arts, 100 (1953), 56-90.

Wilkes, M. V. Automatic Digital Computers. London, 1956.

----------. "Can Machines Think?" Spectator, 1951, 177-178.

----------. "Can Machines Think?" Proceedings of the IRE, 41 (1953), 1230-1234.

----------. "Progress in High-Speed Calculating Machine Design." Nature, 164 (1949), 341-343.

Wilkie, J. S. The Science of Mind and Brain. London, 1953.

Wilkinson, J. H. "The Pilot ACE at the National Physical Laboratory." Radio and Electronic Engineer, 45 (1975), 336-340.

----------. "Progress Report on the Automatic Computing Engine." Divisional Report, National Physical Laboratory. April, 1948.

Williams, F. C. "A Cathode-Ray Tube Digit Store." Proceedings of the Royal Society of London, 195 (1948), 279-284.

Winterbotham, F. W. The Ultra Secret. New York, 1974.

Widom, J. O. "The Hypothesis of Cybernetics." British Journal for the Philosophy of Science, 2 (1951), 1-24.

----------. "A New Model for the Mind-Body Relationship." British Journal for the Philosophy of Science, 2 (1951), 295-301.

----------, R. J. S. Spilsburg, and D. M. MacKay. "Symposium: Mentality in Machines." Proceedings of the Aristotelean Society, Supp. Vol. 26 (1952), 1-61.

Wittgenstein, Ludwig. Remarks on the Foundations of Mathematics. Oxford, 1956.

Womersley, J. R. "ACE" Machine Project. Memorandum to Executive Committee, National Physical Laboratory. Teddington, England. 19 March 1946.

Woodger, M. A Little Light on the Prehistory of ACE and EDVAC: Extracts from Correspondence on File at N.P.L.. Unpublished memorandum. Feb. 1, 1977. Teddington, England.

Woodger, M. "The History and Present Use of Digital Computing at the National Physical Laboratory." Process Control and Automation, 5 (1958), 438-443.

Woodward, P. M. Probability and Information Theory, with Application to Radar. London, 1953.

Wooldridge, D. E. The Machinery of the Brain. New York, 1963.

Wright, M. A. "Can Machines Be Intelligent?" Process Control and Automation, 6 (1959), 2-6.

Young, J. Z. A Model of the Brain. Oxford, 1964.

Zeilon, Nils. "Ivar Fredholm." Acta Mathematica, 54 (1930), 1-16.

Zermelo, Ernst. "Beweis, dass jede Menge wohlgeordnet werden Kann." Mathematische Annalen, 59 (1904), 514-516.

----------. "Neuer Beweis für die Möglichkeit einer Wohlordnung." Mathematische Annalen, 65 (1908), 107-128.

----------. "Untersuchungen über die Grundlagen der Mengenlehre, I." Mathematische Annalen, 65 (1908), 261-281.

TITLE OF THESIS FROM MATHEMATICAL CONSTRUCTIVITY TO COMPUTER SCIENCE: ALAN TURING, JOHN VON NEUMANN, AND THE ORIGINS OF COMPUTER SCIENCE IN MATHEMATICAL LOGIC

MAJOR PROFESSOR Victor L. Hilts

MAJOR DEPARTMENT History of Science

MINOR(S) Mathematics

NAME William F. Aspray, Jr.

PLACE AND DATE OF BIRTH Trenton, N. J. May 12, 1952

COLLEGES AND UNIVERSITIES: YEARS ATTENDED AND DEGREES

Wesleyan University 1970-73 B. A., M. A.

Princeton University 1975-76

University of Toronto 1976-77

University of Wisconsin 1973-75, 77-80 M. A., Ph. D.

MEMBERSHIPS IN LEARNED OR HONORARY SOCIETIES Sigma Xi, History of Science Society

PUBLICATIONS none

DATE 18 July 1980

F-5266