ON COMPUTABILITY

Wilfried Sieg

1 INTRODUCTION

Computability is perhaps the most significant and distinctive notion modern logic has introduced; in the guise of decidability and effective calculability it has a venerable history within philosophy and mathematics. Now it is also the basic theoretical concept for computer science, artificial intelligence and cognitive science. This essay discusses, at its heart, methodological issues that are central to any mathematical theory that is to reflect parts of our physical or intellectual experience. The discussion is grounded in historical developments that are deeply intertwined with meta-mathematical work in the foundations of mathematics. How is that possible, the reader might ask, when the essay is concerned *solely* with computability? This introduction begins to give an answer by first describing the context of foundational investigations in logic and mathematics and then sketching the main lines of the systematic presentation.

1.1 Foundational contexts

In the second half of the 19^{th} century the issues of decidability and effective calculability rose to the fore in discussions concerning the nature of mathematics. The divisive character of these discussions is reflected in the tensions between Dedekind and Kronecker, each holding broad methodological views that affected deeply their scientific practice. Dedekind contributed perhaps most to the radical transformation that led to modern mathematics: he introduced abstract axiomatizations in parts of the subject (e.g., algebraic number theory) and in the foundations for arithmetic and analysis. Kronecker is well known for opposing that high level of structuralist abstraction and insisting, instead, on the decidability of notions and the effective construction of mathematical objects from the natural numbers. Kronecker's concerns were of a traditional sort and were recognized as perfectly legitimate by Hilbert and others, as long as they were positively directed towards the effective solution of mathematical problems and not negatively used to restrict the free creations of the mathematical mind.

At the turn of the 20^{th} century, these structuralist tendencies found an important expression in Hilbert's book *Grundlagen der Geometrie* and in his essay *Über den Zahlbegriff*. Hilbert was concerned, as Dedekind had been, with the consistency of the abstract notions and tried to address the issue also within a broad set

Handbook of the Philosophy of Science. Philosophy of Mathematics
Volume editor: Andrew Irvine. General editors: Dov M. Gabbay, Paul Thagard and John Woods.

theoretic/logicist framework. The framework could have already been sharpened at that point by adopting the contemporaneous development of Frege's *Begriffsschrift*, but that was not done until the late 1910s, when Russell and Whitehead's work had been absorbed in the Hilbert School. This rather circuitous development is apparent from Hilbert and Bernays' lectures [1917/18] and the many foundational lectures Hilbert gave between 1900 and the summer semester of 1917. Apart from using a version of *Principia Mathematica* as the frame for formalizing mathematics in a direct way, Hilbert and Bernays pursued a dramatically different approach with a sharp focus on meta-mathematical questions like the semantic completeness of logical calculi and the syntactic consistency of mathematical theories.

In his *Habilitationsschrift* of 1918, Bernays established the semantic completeness for the sentential logic of *Principia Mathematica* and presented a system of provably independent axioms. The completeness result turned the truth-table test for validity (or logical truth) into an effective criterion for provability in the logical calculus. This latter problem has a long and distinguished history in philosophy and logic, and its pre-history reaches back at least to Leibniz. I am alluding of course to the decision problem ("Entscheidungsproblem"). Its classical formulation for first-order logic is found in Hilbert and Ackermann's book *Grundzüge der theoretischen Logik*. This problem was viewed as *the* main problem of mathematical logic and begged for a rigorous definition of mechanical procedure or finite decision procedure.

How intricately the "Entscheidungsproblem" is connected with broad perspectives on the nature of mathematics is brought out by an amusingly illogical argument in von Neumann's essay *Zur Hilbertschen Beweistheorie* from 1927:

> ... it appears that there is no way of finding the general criterion for deciding whether or not a well-formed formula a is provable. (We cannot at the moment establish this. Indeed, we have no clue as to how such a proof of undecidability would go.) ... the undecidability is even a *conditio sine qua non* for the contemporary practice of mathematics, using as it does heuristic methods, to make any sense. The very day on which the undecidability does not obtain any more, mathematics as we now understand it would cease to exist; it would be replaced by an absolutely mechanical prescription (eine absolut mechanische Vorschrift) by means of which anyone could decide the provability or unprovability of any given sentence.
>
> Thus we have to take the position: it is generally undecidable, whether a given well-formed formula is provable or not.

If the underlying conceptual problem had been attacked directly, then something like Post's unpublished investigations from the 1920s would have been carried out in Göttingen. A different and indirect approach evolved instead, whose origins can be traced back to the use of calculable number theoretic functions in finitist consistency proofs for parts of arithmetic. Here we find the most concrete beginning of the history of modern computability with close ties to earlier mathematical and later logical developments.

There is a second sense in which "foundational context" can be taken, not as referring to work in the foundations of mathematics, but directly in modern logic and cognitive science. Without a deeper understanding of the nature of calculation and underlying processes, neither the scope of undecidability and incompleteness results nor the significance of computational models in cognitive science can be explored in their proper generality. The claim for logic is almost trivial and implies the claim for cognitive science. After all, the relevant logical notions have been used when striving to create artificial intelligence or to model mental processes in humans. These foundational problems come strikingly to the fore in arguments for Church's or Turing's Thesis, asserting that an informal notion of effective calculability is captured fully by a particular precise mathematical concept. Church's Thesis, for example, claims in its original form that the effectively calculable number theoretic functions are exactly those functions whose values are computable in Gödel's equational calculus, i.e., the general recursive functions.

There is general agreement that Turing gave the most convincing analysis of effective calculability in his 1936 paper *On computable numbers — with an application to the Entscheidungsproblem.* It is Turing's distinctive philosophical contribution that he brought the computing agent into the center of the analysis and that was for Turing a human being, proceeding mechanically.[1] Turing's student Gandy followed in his [1980] the outline of Turing's work in his analysis of machine computability. Their work is not only closely examined in this essay, but also thoroughly recast. In the end, the detailed conceptual analysis presented below yields rigorous characterizations that dispense with theses, reveal human and machine computability as axiomatically given mathematical concepts and allow their systematic reduction to Turing computability.

1.2 Overview

The core of section 2 is devoted to decidability and calculability. Dedekind introduced in his essay *Was sind und was sollen die Zahlen?* the general concept of a "(primitive) recursive" function and proved that these functions can be made explicit in his logicist framework. Beginning in 1921, these obviously calculable functions were used prominently in Hilbert's work on the foundations of mathematics, i.e., in the particular way he conceived of finitist mathematics and its role in consistency proofs. Hilbert's student Ackermann discovered already before 1925 a non-primitive recursive function that was nevertheless calculable. In 1931, Herbrand, working on Hilbert's consistency problem, gave a very general and open-ended characterization of "finitistically calculable number-theoretic functions" that included also the Ackermann function. This section emphasizes the

[1] The Shorter Oxford English Dictionary makes perfectly clear that *mechanical*, when applied to a person or action, means "performing or performed without thought; lacking spontaneity or originality; machine-like; automatic, routine."

broader intellectual context and points to the rather informal and epistemologically motivated demand that, in the development of logic and mathematics, certain notions (for example, proof) *should* be decidable by humans and others *should not* (for example, theorem). The crucial point is that the core concepts were deeply intertwined with mathematical practice and logical tradition before they came together in Hilbert's consistency program or, more generally, in meta-mathematics.

In section 3, entitled *Recursiveness and Church's Thesis*, we see that Herbrand's broad characterization was used in Gödel's 1933 paper reducing classical to intuitionist arithmetic. It also inspired Gödel to give a definition of "general recursive functions" in his 1934 Princeton Lectures. Gödel was motivated by the need for a rigorous and adequate notion of "formal theory" so that a general formulation of his incompleteness theorems could be given. Church, Kleene and Rosser investigated Gödel's notion that served subsequently as the rigorous concept in Church's first published formulation of his thesis in [Church, 1935]. Various arguments in support of the thesis, given by Church, Gödel and others, are considered in detail and judged to be inadequate. They all run up against the same stumbling block of having to characterize elementary calculation steps rigorously and without circles. That difficulty is brought out in a conceptually and methodologically clarifying way by the analysis of "reckonable function" ("regelrecht auswertbare Funktion") given in Hilbert and Bernays' 1939 book.

Section 4 takes up matters where they were left off in the third section, but proceeds in a quite different direction: it returns to the original task of characterizing mechanical procedures and focuses on computations and combinatory processes. It starts out with a look at Post's brief 1936 paper, in which a human worker operates in a "symbol space" and carries out very simple operations. Post hypothesized that the operations of such a worker can effect all mechanical or, in his terminology, combinatory processes. This hypothesis is viewed as being in need of continual verification. It is remarkable that Turing's model of computation, developed independently in the same year, is "identical". However, the contrast in methodological approach is equally, if not more, remarkable. Turing took the calculations of human computers or "computors" as a starting-point of a detailed analysis and reduced them, appealing crucially to the agents' sensory limitations, to processes that can be carried out by Turing machines. The restrictive features can be formulated as *boundedness* and *locality* conditions. Following Turing's approach, Gandy investigated the computations of machines or, to indicate the scope of that notion more precisely, of "discrete mechanical devices" that can compute in parallel. In spite of the great generality of his notion, Gandy was able to show that any machine computable function is also Turing computable.

Both Turing and Gandy rely on a restricted *central* thesis, when connecting an informal concept of calculability with a rigorous mathematical one. I sharpen Gandy's work and characterize "Turing Computors" and "Gandy Machines" as discrete dynamical systems satisfying appropriate axiomatic conditions. Any Turing computor or Gandy machine turns out to be computationally reducible to a Turing machine. These considerations constitute the core of section 5 and lead to the conclusion that computability, when relativized to a particular kind of computing device, has a standard methodological status: no thesis is needed, but rather the recognition that the axiomatic conditions are correct for the intended device. The proofs that the characterized notions are equivalent to Turing computability establish then important mathematical facts.

In section 6, I give an "Outlook on Machines and Mind". The question, whether there are concepts of effectiveness broader than the ones characterized by the axioms for Gandy machines and Turing computors, has of course been asked for both physical and mental processes. I discuss the seemingly sharp conflict between Gödel and Turing expressed by Gödel, when asserting: i) Turing tried (and failed) in his [1936] to reduce all mental processes to mechanical ones, and ii) the human mind infinitely surpasses any finite machine. This conflict can be clarified and resolved by realizing that their deeper disagreement concerns the nature of machines. The section ends with some brief remarks about supra-mechanical devices: if there are such, then they cannot satisfy the physical restrictions expressed through the boundedness and locality conditions for Gandy machines. Such systems must violate either the upper bound on signal propagation or the lower bound on the size of distinguishable atomic components; such is the application of the axiomatic method.

1.3 Connections

Returning to the beginning, we see that Turing's notion of human computability is exactly right for both a convincing negative solution of the "Entscheidungsproblem" and a precise characterization of formal systems that is needed for the general formulation of the incompleteness theorems. One disclaimer and one claim should be made at this point. For many philosophers computability is of special importance because of its central role in "computational models of the human mind". This role is touched upon only indirectly through the reflections on the nature and content of Church's and Turing's theses. The disclaimer is complemented by the claim that the conceptual analysis naturally culminates in the formulation of axioms that characterize different computability notions. Thus, arguments in support of the various theses should be dismissed in favor of considerations for the adequacy of axiomatic characterizations of computations that do not correspond to deep mental procedures, but rather to strictly mechanical processes.

Wittgenstein's terse remark about Turing machines, "These machines are *humans* who calculate,"[2] captures the very feature of Turing's analysis of calculability that makes it epistemologically relevant. Focusing on the epistemology of mathematics, I will contrast this feature with two striking aspects of mathematical experience implicit in repeated remarks of Gödel's. The first "conceptional" aspect is connected to the notion of effective calculability through his assertion that

[2]From [1980, § 1096]. I first read this remark in [Shanker, 1987], where it is described as a "mystifying reference to Turing machines." In his later book [Shanker, 1998] that characterization is still maintained.

"with this concept one has for the first time succeeded in giving an absolute definition of an interesting epistemological notion". The second "quasi-constructive" aspect is related to axiomatic set theory through his claim that its axioms "can be supplemented without arbitrariness by new axioms which are only the natural continuation of the series of those set up so far". Gödel speculated how the second aspect might give rise to a humanly effective procedure that cannot be mechanically calculated. Gödel's remarks point to data that underlie the two aspects and challenge, in the words of Parsons[3], "any theory of meaning and evidence in mathematics". Not that I present a theory accounting for these data. Rather, I clarify the first datum by reflecting on the question that is at the root of Turing's analysis. In its sober mathematical form the question asks, "*What is an effectively calculable function?*"

2 DECIDABILITY AND CALCULABILITY

This section is mainly devoted to the *decidability* of relations between finite syntactic objects and the *calculability* of number theoretic functions. The former notion is seen by Gödel in 1930 to be derivative of the latter, since such relations are considered to be decidable just in case the characteristic functions of their arithmetic analogues are calculable. Calculable functions rose to prominence in the 1920s through Hilbert's work on the foundations of mathematics. Hilbert conceived of finitist mathematics as an extension of the Kroneckerian part of constructive mathematics and insisted programmatically on carrying out consistency proofs by finitist means only. Herbrand, who worked on Hilbert's consistency problem, gave a general and open-ended characterization of "finitistically calculable functions" in his last paper [Herbrand, 1931a]. This characterization was communicated to Gödel in a letter of 7 April 1931 and inspired the notion of general recursive function that was presented three years later in Gödel's Princeton Lectures and is the central concept to be discussed in Section 3.

Though this specific meta-mathematical background is very important, it is crucial to see that it is embedded in a broader intellectual context, which is philosophical as well as mathematical. There is, first, the normative requirement that some central features of the formalization of logic and mathematics should be decidable on a radically inter-subjective basis; this holds, in particular, for the proof relation. It is reflected, second, in the quest for showing the decidability of problems in pure mathematics and is connected, third, to the issue of predictability in physics and other sciences. Returning to the meta-mathematical background, Hilbert's Program builds on the formalization of mathematics and thus incorporates aspects of the normative requirement. Gödel expressed the idea for realizing this demand in his [1933a]:

> The first part of the problem [see fn. 4 for the formulation of "the problem"] has been solved in a perfectly satisfactory way, the solu-

[3]In [Parsons, 1995].

> tion consisting in the so-called "formalization" of mathematics, which means that a perfectly precise language has been invented, by which it is possible to express any mathematical proposition by a formula. Some of these formulas are taken as axioms, and then certain rules of inference are laid down which allow one to pass from the axioms to new formulas and thus to deduce more and more propositions, the outstanding feature of the rules of inference being that they are purely formal, i.e., refer only to the outward structure of the formulas, not to their meaning, so that they could be applied by someone who knew nothing about mathematics, or by a machine.[4]

Let's start with a bit of history and see how the broad issue of decidability led to the question, "What is the precise extension of the class of calculable number theoretic functions?"

2.1 Decidability

Any historically and methodologically informed account of calculability will at least point to Leibniz and the goals he sought to achieve with his project of a *characteristica universalis* and an associated *calculus ratiocinator*. Similar projects for the development of artificial languages were common in 17^{th} century intellectual circles. They were pursued for their expected benefits in promoting religious and political understanding, as well as commercial exchange. Leibniz's project stands out for its emphasis on mechanical reasoning: a universal character is to come with algorithms for making and checking inferences. The motivation for this requirement emerges from his complaint about Descartes's *Rules for the direction of the mind*. Leibniz views them as a collection of vague precepts, requiring intellectual effort as well as ingenuity from the agents following the rules. A reasoning method, such as the universal character should provide, comes by contrast with rules that completely determine the actions of the agents. Neither insight nor intellectual effort is needed, as a mechanical thread of reasoning guides everyone who can perceive and manipulate concrete configurations of symbols.

> Thus I assert that all truths can be demonstrated about things expressible in this language with the addition of new concepts not yet expressed in it — all such truths, I say, can be demonstrated *solo calculo*, or solely by the manipulation of characters according to a certain form, without any labor of the imagination or effort of the mind, just

[4]Cf. p. 45 of [Gödel 1933a]. To present the context of the remark, I quote the preceding paragraph of Gödel's essay: "The problem of giving a foundation of mathematics (and by mathematics I mean here the totality of the methods of proof actually used by mathematicians) can be considered as falling into two different parts. At first these methods of proof have to be reduced to a minimum number of axioms and primitive rules of inference, which have to be stated as precisely as possible, and then secondly a justification in some sense or other has to be sought for these axioms, i.e., a theoretical foundation of the fact that they lead to results agreeing with each other and with empirical facts."

> as occurs in arithmetic and algebra. (Quoted in [Mates, 1986, fn. 65, 185])

Leibniz's expectations for the growth of our capacity to resolve disputes were correspondingly high. He thought we might just sit down at a table, formulate the issues precisely, take our pens and say *Calculemus*! After finitely many calculation steps the answer would be at hand, or rather visibly on the table. The thought of having machines carry out the requisite mechanical operations had already occurred to Lullus. It was pursued further in the 19^{th} century by Jevons and was pushed along by Babbage in a theoretically and practically most ambitious way.

The idea of an epistemologically unproblematic method, turning the task of testing the conclusiveness of inference chains (or even of creating them) into a purely mechanical operation, provides a direct link to Frege's *Begriffsschrift* and to the later reflections of Peano, Russell, Hilbert, Gödel and others. Frege, in particular, saw himself in this Leibnizian tradition as he emphasized in the introduction to his 1879 booklet. That idea is used in the 20^{th} century as a normative requirement on the fully explicit presentation of mathematical proofs in order to insure inter-subjectivity. In investigations concerning the foundations of mathematics that demand led from axiomatic, yet informal presentations to fully formal developments. As an example, consider the development of elementary arithmetic in [Dedekind 1888] and [Hilbert 1923]. It can't be overemphasized that the step from *axiomatic systems* to *formal theory* is a radical one, and I will come back to it in the next subsection.[5]

There is a second Leibnizian tradition in the development of mathematical logic that leads from Boole and de Morgan through Peirce to Schröder, Löwenheim and others. This tradition of the algebra of logic had a deep impact on the classical formulation of modern mathematical logic in Hilbert and Ackermann's book. Particularly important was the work on the decision problem, which had a longstanding tradition in algebraic logic and had been brought to a highpoint in Löwenheim's paper from 1915, *Über Möglichkeiten im Relativkalkül*. Löwenheim established, in modern terminology, the decidability of monadic first-order logic and the reducibility of the decision problem for first-order logic to its binary fragment. The importance of that mathematical insight was clear to Löwenheim, who wrote about his reduction theorem:

> We can gauge the significance of our theorem by reflecting upon the fact that every theorem of mathematics, or of any calculus that can be invented, can be written as a relative equation; the mathematical

[5]The nature of this step is clearly discussed in the Introduction to Frege's *Grundgesetze der Arithmetik*, where he criticizes Dedekind for not having made explicit all the methods of inference: "In a much smaller compass it [i.e., Dedekind's *Was sind und was sollen die Zahlen?*] follows the laws of arithmetic much farther than I do here. This brevity is only arrived at, to be sure, because much of it is not really proved at all. ... nowhere is there a statement of the logical laws or other laws on which he builds, and, even if there were, we could not possibly find out whether really no others were used — for to make that possible the proof must be not merely indicated but completely carried out." [Geach and Black, 119]

> theorem then stands or falls according as the equation is satisfied or not. This transformation of arbitrary mathematical theorems into relative equations can be carried out, I believe, by anyone who knows the work of Whitehead and Russell. Since, now, according to our theorem the whole relative calculus can be reduced to the binary relative calculus, it follows that we can decide whether an arbitrary mathematical proposition is true provided we can decide whether a binary relative equation is identically satisfied or not. (p. 246)

Many of Hilbert's students and collaborators worked on the decision problem, among them Ackermann, Behmann, Bernays, Schönfinkel, but also Herbrand and Gödel. Hilbert and Ackermann made the connection of mathematical logic to the algebra of logic explicit. They think that the former provides more than a precise language for the following reason: "Once the logical formalism is fixed, it can be expected that a systematic, so-to-speak calculatory treatment of logical formulas is possible; that treatment would roughly correspond to the theory of equations in algebra." (p. 72) Subsequently, they call sentential logic "a developed algebra of logic". The decision problem, solved of course for the case of sentential logic, is viewed as one of the most important logical problems; when it is extended to full first-order logic it must be considered "as the main problem of mathematical logic". (p. 77) Why the decision problem should be considered as *the* main problem of mathematical logic is stated clearly in a remark that may remind the reader of Löwenheim's and von Neumann's earlier observations:

> The solution of this general decision problem would allow us to decide, at least in principle, the provability or unprovability of an arbitrary mathematical statement. (p. 86)

Taking for granted the finite axiomatizability of set theory or some other fundamental theory in first-order logic, the general decision problem is solved when that for first-order logic has been solved. And what is required for its solution?

> The decision problem is solved, in case a procedure is known that permits — for a given logical expression — to decide the validity, respectively satisfiability, by finitely many operations. (p. 73)

Herbrand, for reasons similar to those of Hilbert and Ackermann, considered the general decision problem in a brief note from 1929 "as the most important of those, which exist at present in mathematics" (p. 42). The note was entitled *On the fundamental problem of mathematics*.

In his paper *On the fundamental problem of mathematical logic* Herbrand presented a little later refined versions of Löwenheim's reduction theorem and gave positive solutions of the decision problem for particular parts of first-order logic. The fact that the theorems are refinements is of interest, but not the crucial reason for Herbrand to establish them. Rather, Herbrand emphasizes again and again that Löwenheim's considerations are "insufficient" (p. 39) and that his proof "is totally inadequate for our purposes" (p. 166). The fullest reason for these judgments is given in section 7.2 of his thesis, *Investigations in proof theory*, when discussing two central theorems, namely, if the formula P is provable (in first-order logic), then its negation is not true in any infinite domain (Theorem 1) and if P is not provable, then we can construct an infinite domain in which its negation is true (Theorem 2).

> Similar results have already been stated by Löwenheim, but his proofs, it seems to us, are totally insufficient for our purposes. First, he gives an intuitive meaning to the notion 'true in an infinite domain', hence his proof of Theorem 2 does not attain the rigor that we deem desirable .... Then — and this is the gravest reproach — because of the intuitive meaning that he gives to this notion, he seems to regard Theorem 1 as obvious. This is absolutely impermissible; such an attitude would lead us, for example, to regard the consistency of arithmetic as obvious. On the contrary, it is precisely the proof of this theorem ... that presented us with the greatest difficulty.
>
> We could say that Löwenheim's proof was sufficient in mathematics. But, in the present work, we had to make it 'metamathematical' (see Introduction) so that it would be of some use to us. (pp. 175–176)

The above theorems provide Herbrand with a method for investigating the decision problem, whose solution would answer also the consistency problem for finitely axiomatized theories. As consistency has to be established by using restricted meta-mathematical methods, Herbrand emphasizes that the decision problem has to be attacked exclusively with such methods. These meta-mathematical methods are what Hilbert called finitist. So we reflect briefly on the origins of finitist mathematics and, in particular, on the views of its special defender and practitioner, Leopold Kronecker.

2.2 Finitist mathematics

In a talk to the Hamburg Philosophical Society given in December 1930, Hilbert reminisced about his finitist standpoint and its relation to Kronecker; he pointed out:

> At about the same time [around 1888], thus already more than a generation ago, Kronecker expressed clearly a view and illustrated it by several examples, which today coincides essentially with our finitist standpoint. [Hilbert, 1931, 487]

He added that Kronecker made only the mistake "of declaring transfinite inferences as inadmissible". Indeed, Kronecker disallowed the classical logical inference from the negation of a universal to an existential statement, because a proof of an existential statement should provide a witness. Kronecker insisted also on the decidability of mathematical notions, which implied among other things the rejection of the general concept of irrational number. In his 1891 lectures *Über den Zahlbegriff in der Mathematik* he formulated matters clearly and forcefully:

> The standpoint that separates me from many other mathematicians culminates in the principle, that the definitions of the experiential sciences (Erfahrungswissenschaften), — i.e., of mathematics and the natural sciences, ... — must not only be consistent in themselves, but must be taken from experience. It is even more important that they must contain a criterion by means of which one can decide for any special case, whether or not the given concept is subsumed under the definition. A definition, which does not provide that, may be praised by philosophers or logicians, but for us mathematicians it is a mere verbal definition and without any value. (p. 240)

Dedekind had a quite different view. In the first section of *Was sind und was sollen die Zahlen?* he asserts that "things", any objects of our thought, can frequently "be considered from a common point of view" and thus "be associated in the mind" to form a system. Such systems S are also objects of our thought and are "completely determined when it is determined for every thing whether it is an element of S or not". Attached to this remark is a footnote differentiating his position from Kronecker's:

> How this determination is brought about, and whether we know a way of deciding upon it, is a matter of indifference for all that follows; the general laws to be developed in no way depend upon it; they hold under all circumstances. I mention this expressly because Kronecker not long ago (*Crelle's Journal*, Vol. 99, pp. 334–336) has endeavored to impose certain limitations upon the free formation of concepts in mathematics, which I do not believe to be justified; but there seems to be no call to enter upon this matter with more detail until the distinguished mathematician shall have published his reasons for the necessity or merely the expediency of these limitations. (p. 797)

In Kronecker's essay *Über den Zahlbegriff* and his lectures *Über den Zahlbegriff in der Mathematik* one finds general reflections on the foundations of mathematics that at least partially address Dedekind's request for clarification.

Kronecker views arithmetic in his [1887] as a very broad subject, encompassing all mathematical disciplines with the exception of geometry and mechanics. He thinks that one will succeed in "grounding them [all the mathematical disciplines] solely on the number-concept in its narrowest sense, and thus in casting off the modifications and extensions of this concept which were mostly occasioned by the applications to geometry and mechanics". In a footnote Kronecker makes clear that he has in mind the addition of "irrational as well as continuous quantities". The principled philosophical distinction between geometry and mechanics on the one hand and arithmetic (in the broad sense) on the other hand is based on Gauss' remarks about the theory of space and the pure theory of quantity: only the latter has "the complete conviction of necessity (and also of absolute truth)," whereas the former has also outside of our mind a reality "to which we cannot *a priori* completely prescribe its laws".

These programmatic remarks are refined in the 1891 lectures. The lecture of 3 June 1891 summarizes Kronecker's perspective on mathematics in four theses. The first asserts that mathematics does not tolerate "Systematik," as mathematical research is a matter of inspiration and creative imagination. The second thesis asserts that mathematics is to be treated as a natural science "for its objects are as real as those of its sister sciences (Schwesterwissenschaften)". Kronecker explains:

> That this is so is sensed by anyone who speaks of mathematical 'discoveries'. Since we can discover only something that already really exists; but what the human mind generates out of itself that is called 'invention'. The mathematician 'discovers', consequently, by methods, which he 'invented' for this very purpose. (pp. 232–3)

The next two theses are more restricted in scope, but have important methodological content. When investigating the fundamental concepts of mathematics and when developing a particular area, the third thesis insists, one has to keep separate the individual mathematical disciplines. This is particularly important, because the fourth thesis demands that, for any given discipline, i) its characteristic methods are to be used for determining and elucidating its fundamental concepts and ii) its rich content is to be consulted for the explication of its fundamental concepts.[6] In the end, the only real mathematical objects are the natural numbers: "True mathematics needs from arithmetic only the [positive] integers." (p. 272)

In his Paris Lecture of 1900, Hilbert formulated as an axiom that any mathematical problem can be solved, either by answering the question posed by the problem or by showing the impossibility of an answer. Hilbert asked, "What is a legitimate condition that solutions of mathematical problems have to satisfy?" Here is the formulation of the central condition:

> I have in mind in particular [the requirement] that we succeed in establishing the correctness of the answer by means of a finite number of inferences based on a finite number of assumptions, which are inherent in the problem and which have to be formulated precisely in each case. This requirement of logical deduction by means of a finite

[6]Kronecker explains the need for ii) in a most fascinating way as follows: "Clearly, when a reasonable master builder has to put down a foundation, he is first going to learn carefully about the building for which the foundation is to serve as the basis. Furthermore, it is foolish to deny that the richer development of a science may lead to the necessity of changing its basic notions and principles. In this regard, there is no difference between mathematics and the natural sciences: new phenomena overthrow the old hypotheses and replace them by others." (p. 233)

> number of inferences is nothing but the requirement of rigor in argumentation. Indeed, the requirement of rigor ... corresponds [on the one hand] to a general philosophical need of our understanding and, on the other hand, it is solely by satisfying this requirement that the thought content and the fruitfulness of the problem in the end gain their full significance. (p. 48)

Then he tries to refute the view that only arithmetic notions can be treated rigorously. He considers that opinion as thoroughly mistaken, though it has been "occasionally advocated by eminent men". That is directed against Kronecker as the next remark makes clear.

> Such a one-sided interpretation of the requirement of rigor soon leads to ignoring all concepts that arise in geometry, mechanics, and physics, to cutting off the flow of new material from the outer world, and finally, as a last consequence, to the rejection of the concepts of the continuum and the irrational number. (p. 49)

Positively and in contrast, Hilbert thinks that mathematical concepts, whether emerging in epistemology, geometry or the natural sciences, are to be investigated in mathematics. The principles for them have to be given by "a simple and complete system of axioms" in such a way that "the rigor of the new concepts, and their applicability in deductions, is in no way inferior to the old arithmetic notions". This is a central part of Hilbert's much-acclaimed axiomatic method, and Hilbert uses it to shift the Kroneckerian effectiveness requirements from the mathematical to the "systematic" meta-mathematical level.[7] That leads, naturally, to a distinction between "solvability in principle" by the axiomatic method and "solvability by algorithmic means". Hilbert's famous 10^{th} Problem concerning the solvability of Diophantine equations is a case in which an algorithmic solution is sought; the

[7]That perspective, indicated here in a very rudimentary form, is of course central for the meta-mathematical work in the 1920s and is formulated in the sharpest possible way in many of Hilbert's later publications. Its epistemological import is emphasized, for example in the first chapter of *Grundlagen der Mathematik I*, p. 2: "Also formal axiomatics definitely requires for its deductions as well as for consistency proofs certain evidences, but with one essential difference: this kind of evidence is not based on a special cognitive relation to the particular subject, but is one and the same for all axiomatic [formal] systems, namely, that primitive form of cognition, which is the prerequisite for any exact theoretical research whatsoever." In his Hamburg talk of 1928 Hilbert stated the remarkable philosophical significance he sees in the proper formulation of the rules for the meta-mathematical "formula game": "For this formula game is carried out according to certain definite rules, in which the *technique of our thinking* is expressed. These rules form a closed system that can be discovered and definitively stated. The fundamental idea of my proof theory 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." He adds, against Kronecker and Brouwer's intuitionism, "If any totality of observations and phenomena deserves to be made the object of a serious and thorough investigation, it is this one. Since, after all, it is part of the task of science to liberate us from arbitrariness, sentiment, and habit and to protect us from the subjectivism that already made itself felt in Kronecker's views and, it seems to me, finds its culmination in intuitionism." [van Heijenoort, 1967, 475]

impossibility of such a solution was found only in the 1970s after extensive work by Robinson, Davis and Matijasevic, work that is closely related to the developments of computability theory described here; cf. [Davis, 1973].

At this point in 1900 there is no firm ground for Hilbert to claim that Kroneckerian rigor for axiomatic developments has been achieved. After all, it is only the radical step from axiomatic to formal theories that guarantees the rigor of solutions to mathematical problems in the above sense, and that step was taken by Hilbert only much later. Frege had articulated appropriate mechanical features and had realized them for the arguments given in his concept notation. His booklet *Begriffsschrift* offered a rich language with relations and quantifiers, whereas its logical calculus required that all assumptions be listed and that each step in a proof be taken in accord with one of the antecedently specified rules. Frege considered this last requirement as a sharpening of the axiomatic method he traced back to Euclid's *Elements*. With this sharpening he sought to recognize the "epistemological nature" of theorems. In the introduction to *Grundgesetze der Arithmetik* he wrote:

> Since there are no gaps in the chains of inferences, each axiom, assumption, hypothesis, or whatever you like to call it, upon which a proof is founded, is brought to light; and so we gain a basis for deciding the epistemological nature of the law that is proved. (p. 118)

But a true basis for such a judgment can be obtained only, Frege realized, if inferences do not require contentual knowledge: their application has to be recognizable as correct on account of the form of the sentences occurring in them. Frege claimed that in his logical system "inference is conducted like a calculation" and observed:

> I do not mean this in a narrow sense, as if it were subject to an algorithm the same as ... ordinary addition and multiplication, but only in the sense that there is an algorithm at all, i.e., a totality of rules which governs the transition from one sentence or from two sentences to a new one in such a way that nothing happens except in conformity with these rules.[8] [Frege, 1984, 237]

Hilbert took the radical step to fully formal axiomatics, prepared through the work of Frege, Peano, Whitehead and Russell, only in the lectures he gave in the winter-term of 1917/18 with the assistance of Bernays. The effective presentation of formal theories allowed Hilbert to formulate in 1922 the finitist consistency program, i.e., describe formal theories in Kronecker-inspired finitist mathematics and formulate consistency in a finitistically meaningful way. In line with the Paris

[8]Frege was careful to emphasize (in other writings) that all of thinking "can never be carried out by a machine or be replaced by a purely mechanical activity" [Frege 1969, 39]. He went on to claim: "It is clear that the syllogism can be brought into the form of a calculation, which however cannot be carried out without thinking; it [the calculation] just provides a great deal of assurance on account of the few rigorous and intuitive forms in which it proceeds."

remarks, he viewed this in [1921/22] as a dramatic expansion of Kronecker's purely arithmetic finitist mathematics:

> We have to extend the domain of objects to be considered, i.e., we have to apply our intuitive considerations also to figures that are not number signs. Thus we have good reason to distance ourselves from the earlier dominant principle according to which each theorem of mathematics is in the end a statement concerning integers. This principle was viewed as expressing a fundamental methodological insight, but it has to be given up as a prejudice. (p. 4a)

As to the extended domain of objects, it is clear that formulas and proofs of formal theories are to be included and that, by contrast, geometric figures are definitely excluded. Here are the reasons for holding that such figures are "not suitable objects" for finitist considerations:

> . . . the figures we take as objects must be completely surveyable and only discrete determinations are to be considered for them. It is only under these conditions that our claims and considerations have the same reliability and evidence as in intuitive number theory. (p. 5a)

If we take this expansion of the domain of objects seriously (as we should, I think), we are dealing not just with numbers and associated principles, but more generally with elements of inductively generated classes and associated principles of proof by induction and definition by recursion. That is beautifully described in the Introduction to Herbrand's thesis and was strongly emphasized by von Neumann in his Königsberg talk of 1930. For our systematic work concerning computability we have to face then two main questions, i) "How do we move from decidability issues concerning finite syntactic configurations to calculability of number theoretic functions?" and ii) "Which number theoretic functions can be viewed as being calculable?"

2.3 (Primitive) Recursion

Herbrand articulated in the Appendix to his [1931] (the paper itself had been written already in 1929) informed doubts concerning the positive solvability of the decision problem: "Note finally that, although at present it seems unlikely that the decision problem can be solved, it has not yet been proved that it is impossible to do so." (p. 259) These doubts are based on the second incompleteness theorem, which is formulated by Herbrand as asserting, "it is impossible to prove the consistency of a theory through arguments formalizable in the theory."

> . . . if we could solve the decision problem in the restricted sense [i.e., for first-order logic], it would follow that every theory which has only a finite number of hypotheses and in which this solution is formalizable

> would be inconsistent (since the question of the consistency of a theory having only a finite number of hypotheses can be reduced to this problem). (p. 258)

A historical fact has to be mentioned here: Herbrand spent the academic year 1930/31 in Germany and worked during the fall of 1930 with von Neumann in Berlin. Already in November of 1930 he learned through von Neumann about Gödel's first incompleteness theorem and by early spring of 1931 he had received through Bernays the galleys of [Gödel 1931].

Von Neumann, in turn, had learned from Gödel himself about a version of the first incompleteness theorem at the *Second Conference for Epistemology of the Exact Sciences* held from 5 to 7 September 1930 in Königsberg. On the last day of that conference a roundtable discussion on the foundations of mathematics took place to which Gödel had been invited. Hans Hahn chaired the discussion and its participants included Carnap, Heyting and von Neumann. Toward the end of the discussion Gödel made brief remarks about the first incompleteness theorem; the transcript of his remarks was published in *Erkenntnis* and as [1931a] in the first volume of his *Collected Works*. This is the background for the personal encounter with von Neumann in Königsberg; Wang reports Gödel's recollections in his [1981]:

> Von Neumann was very enthusiastic about the result and had a private discussion with Gödel. In this discussion, von Neumann asked whether number-theoretical undecidable propositions could also be constructed in view of the fact that the combinatorial objects can be mapped onto the integers and expressed the belief that it could be done. In reply, Gödel said, "Of course undecidable propositions about integers could be so constructed, but they would contain concepts quite different from those occurring in number theory like addition and multiplication." Shortly afterward Gödel, to his own astonishment, succeeded in turning the undecidable proposition into a polynomial form preceded by quantifiers (over natural numbers). At the same time but independently of this result, Gödel also discovered his second theorem to the effect that no consistency proof of a reasonably rich system can be formalized in the system itself. (pp. 654–5)

This passage makes clear that Gödel had not yet established the second incompleteness theorem at the time of the Königsberg meeting. On 23 October 1930 Hahn presented to the Vienna Academy of Sciences an abstract containing the theorem's classical formulation. The full text of Gödel's 1931-paper was submitted to the editors of *Monatshefte* on 17 November 1930.[9] The above passage makes

[9] As to the interaction between von Neumann and Gödel after Königsberg and von Neumann's independent discovery of the second incompleteness theorem, cf. their correspondence published in volume V of Gödel's *Collected Works*. — In the preliminary reflections of his [1931] Gödel simply remarks on p. 146 about the "arithmetization": "For meta-mathematical considerations it is of course irrelevant, which objects are taken as basic signs, and we decide to use natural numbers as such [basic signs]."

also clear something surprising, namely, that the arithmetization of syntax used so prominently in the 1931 paper was seemingly developed only after the Königsberg meeting. (There is also no hint of this technique in [Gödel, 1931a].) Given an effective coding of syntax the part of finitist mathematics needed for the description of formal theories is consequently contained in finitist number theory, and finitist decision procedures can then presumably be captured by finitistically calculable number theoretic functions. This answers the first question formulated at the end of section 2.2. Let us take now a step towards answering the second question, "Which number theoretic functions can be viewed as being calculable?"

It was Kronecker who insisted on decidability of mathematical notions and calculability of functions, but it was Dedekind who formulated in *Was sind und was sollen die Zahlen?* the general concept of a "(primitive) recursive" function. These functions are obviously calculable and Dedekind proved, what is not so important from our computational perspective, namely, that they can be made explicit in his logicist framework.[10] Dedekind considers a *simply infinite system* $(N, \varphi, 1)$ that is characterized by axiomatic conditions, now familiar as the Dedekind-Peano axioms:

$$
\begin{gathered}
1 \in N, \\
(\forall n \in N)\ \varphi(n) \in N, \\
(\forall n, m \in N)(\varphi(n) = \varphi(m) \rightarrow n = m), \\
(\forall n \in N)\ \varphi(n) \neq 1 \text{ and} \\
(1 \in \Sigma\ \&\ (\forall n \in N)(n \in \Sigma \rightarrow \varphi(n) \in \Sigma)) \rightarrow (\forall n \in N)\ n \in \Sigma.
\end{gathered}
$$

(Σ is any subset of N.) For this and other simply infinite systems Dedekind isolates a crucial feature in theorem 126, *Satz der Definition durch Induktion*: let $(N, \varphi, 1)$ be a simply infinite system, let θ be an arbitrary mapping from a system Ω to itself, and let ω be an element of Ω; then there is exactly one mapping ψ from N to Ω satisfying the recursion equations:

$$
\begin{gathered}
\psi(1) = \omega, \\
\psi(\varphi(n)) = \theta(\psi(n)).
\end{gathered}
$$

The proof requires subtle meta-mathematical considerations; i.e., an inductive argument for the existence of approximations to the intended mapping on initial segments of N. The basic idea was later used in axiomatic set theory and extended to functions defined by transfinite recursion. It is worth emphasizing that Dedekind's is a very abstract idea: show the existence of a unique solution for a functional equation! Viewing functions as given by calculation procedures, Dedekind's general point recurs in [Hilbert, 1921/22], [Skolem, 1923], [Herbrand, 1931a], and [Gödel, 1934], when the existence of a solution is guaranteed by the existence of a calculation procedure.

In the context of his overall investigation concerning the nature and meaning of number, Dedekind draws two important conclusions with the help of theorem

[10] However, in his [193?] Gödel points out on p. 21, that it is Dedekind's method that is used to show that recursive definitions can be defined explicitly in terms of addition and multiplication.

126: on the one hand, all simply infinite systems are similar (theorem 132), and on the other hand, any system that is similar to a simply infinite one is itself simply infinite (theorem 133). The first conclusion asserts, in modern terminology, that the Dedekind-Peano axioms are categorical. Dedekind infers in his remark 134 from this fact, again in modern terminology, that all simply infinite systems are elementarily equivalent — claiming to justify in this way his abstractive conception of natural numbers.

Dedekind's considerations served a special foundational purpose. However, the recursively defined number theoretic functions have an important place in mathematical practice and can be viewed as part of constructive (Kroneckerian) mathematics quite independent of their logicist foundation. As always, Dedekind himself is very much concerned with the impact of conceptual innovations on the development of actual mathematics. So he uses the recursion schema to define the arithmetic operations of addition, multiplication and exponentiation. For addition, to consider just one example, take Ω to be N, let ω be m and define

$$m + 1 = \varphi(m)$$
$$m + \varphi(n) = \varphi(m + n).$$

Then Dedekind establishes systematically the fundamental properties of these operations (e.g., for addition and multiplication, commutativity, associativity, and distributivity, but also their compatibility with the ordering of N). It is an absolutely elementary and rigorously detailed development that uses nothing but the schema of primitive recursion to define functions and the principle of proof by induction (only for equations) to establish general statements. In a sense it is a more principled and focused presentation of this elementary part of finitist mathematics than that given by either Kronecker, Hilbert and Bernays in their 1921/22 lectures, or Skolem in his 1923 paper, where the foundations of elementary arithmetic are established on the basis "of the recursive mode of thought, without the use of apparent variables ranging over infinite domains".

In their Lecture Notes [1921/22], Hilbert and Bernays treat elementary arithmetic from their new finitist standpoint; here, in elementary arithmetic, they say, we have "that complete certainty of our considerations. We get along without axioms, and the inferences have the character of the concretely-certain." They continue:

> It is first of all important to see clearly that this part of mathematics can indeed be developed in a definitive way and in a way that is completely satisfactory for knowledge. The standpoint we are gaining in this pursuit is of fundamental importance also for our later considerations. (p. 51)

Their standpoint allows them to develop elementary arithmetic as "an intuitive theory of certain simple figures ... , which we are going to call number signs (Zahlzeichen)". The latter are generated as $1, 1+1$, etc. The arithmetic operations are introduced as concrete operations on number signs. For example, $a + b$ refers to the number sign "which is obtained by first placing + after the number sign a and then the number sign b". (p. 54) Basic arithmetic theorems like associativity of addition, are obtained by intuitive considerations including also the "ordinary counting of signs". They define less-than as a relation between number signs: a is less than b, just in case a coincides with a proper part of b. Then they use the *method of descent* to prove general statements, for example, the commutativity of addition. Having defined also divisibility and primality in this concrete manner, they establish Euclid's theorem concerning the infinity of primes. They assert:

> Now we can proceed in this manner further and further; we can introduce the concepts of the greatest common divisor and the least common multiple, furthermore the number congruences. (p. 62)

That remark is followed immediately by the broader methodological claim that the definition of number theoretic functions by means of recursion formulas is admissible from the standpoint of their intuitive considerations. However, "For every single such definition by recursion it has to be determined that the application of the recursion formula indeed yields a number sign as function value — for each set of arguments."[11] They consider then as an example the standard definition of exponentiation. The mathematical development is concluded with the claim:

> Fermat's little theorem, furthermore the theorems concerning quadratic residues can be established by the usual methods as intuitive theorems concerning the number signs. In fact all of elementary number theory can be developed as a theory of number signs by means of concrete intuitive considerations. (p. 63)

This development is obviously carried farther than Dedekind's and proceeds in a quite different, constructive foundational framework. For our considerations concerning computability it is important that we find here in a rough form Herbrand's way of characterizing finistically calculable functions; that will be discussed in the next subsection.

Skolem's work was carried out in 1919, but published only in 1923; there is an acknowledged Kroneckerian influence, but the work is actually carried out in a fragment of *Principia Mathematica*. Skolem takes as basic the notions "natural number", "the number $n + 1$ following the number n", as well as the "recursive mode of thought". By the latter, I suppose, Skolem understands the systematic use of "recursive definitions" and "recursive proof", i.e., definition by primitive recursion and proof by induction. Whereas the latter is indeed taken as a principle, the former is not really: for each operation or relation (via its characteristic function) an appropriate descriptive function in the sense of *Principia Mathematica*

[11] Here is the German formulation of this crucial condition: "Es muss nur bei jeder solchen Definition durch Rekursion eigens festgestellt werden, dass tatsächlich für jede Wertbestimmung der Argumente die Anwendung der Rekursionsformel ein Zahlzeichen als Funktionswert liefert."

has to be shown to have an unambiguous meaning, i.e., to be properly defined.[12] The actual mathematical development leads very carefully, and in much greater detail than in Hilbert and Bernays' lectures, to Euclid's theorem in the last section of the paper; the paper ends with reflections on cardinality. It is Skolem's explicit goal to avoid unrestricted quantification, as that would lead to "an infinite task — that means one that cannot be completed ..." (p. 310). In the Concluding Remark that was added to the paper at the time of its publication, Skolem makes a general point that is quite in the spirit of Hilbert: "The justification for introducing apparent variables ranging over infinite domains therefore seems very problematic; that is, one can doubt that there is any justification for the actual infinite or the transfinite." (p. 332) Skolem also announces the publication of another paper, he actually never published, in which the "formal cumbrousness" due to his reliance on *Principia Mathematica* would be avoided. "But that work, too," Skolem asserts, "is a consistently finitist one; it is built upon Kronecker's principle that a mathematical definition is a genuine definition if and only if it leads to the goal by means of a *finite* number of trials." (p. 333)

Implicit in these discussions is the specification of a class PR of functions that is obtained from the successor function by explicit definitions and the schema of (primitive) recursion. The definition of the class PR emerged in the 1920s; in Hilbert's *On the Infinite* (pp. 387–8) one finds it in almost the contemporary form: it is given inductively by specifying initial functions and closing under two definitional schemas, namely, what Hilbert calls *substitution* and *(elementary) recursion*. This can be done more precisely as follows: PR contains as its initial functions the zero-function Z, the successor function S, and the projection functions P_i^n for each n and each i with $1 \leq i \leq n$. These functions satisfy the equations $Z(x) = 0, S(x) = x'$, and $P_i^n(x_1, \ldots, x_n) = x_i$, for all $x, x_1, \ldots, x_n; x'$ is the successor of x. The class is closed under the *schema of composition*: Given an m-place function ψ in PR and n-place functions $\varphi_1, \ldots, \varphi_m$ in PR, the function ϕ defined by

$$\phi(x_1, \ldots, x_n) = \psi(\varphi_1(x_1, \ldots, x_n), \ldots, \varphi_m(x_1, \ldots, x_n))$$

is also in PR; ϕ is said to be obtained by *composition* from ψ and $\varphi_1, \ldots, \varphi_m$. PR is also closed under the *schema of primitive recursion*: Given an n-place function ψ in PR, and an $n + 24$-place function φ in PR, the function ϕ defined by

$$\begin{aligned} &\phi(x_1, \ldots, x_n, 0) = \psi(x_1, \ldots, x_n) \\ &\phi(x_1, \ldots, x_n, y') = \varphi(x_1, \ldots, x_n, y, \phi(x_1, \ldots, x_n, y)) \end{aligned}$$

is a function in PR; ϕ is said to be obtained by primitive recursion from ψ and φ. Thus, a function is primitive recursive if and only if it can be obtained from some initial functions by finitely many applications of the composition and recursion schemas. This definition was essentially given in Gödel's 1931 paper together with

[12] That is done for addition on p. 305, for the less-than relation on p. 307, and for subtraction on p. 314.

arguments that this class contains the particular functions that are needed for the arithmetic description of *Principia Mathematica* and related systems.

By an inductive argument on the definition of PR one can see that the values of primitive recursive functions can be determined, for any particular set of arguments, by a standardized calculation procedure; thus, all primitive recursive functions are in this sense calculable. Yet there are calculable functions, which are not primitive recursive. An early example is due to Hilbert's student Ackermann; it was published in 1928, but discussed already in [Hilbert, 1925]. Here is the definition of the *Ackermann function*:

$$\begin{aligned} \phi_0(x,y) &= S(y) \\ \phi_{n'}(x,0) &= \begin{cases} x & \text{if } n=0 \\ 0 & \text{if } n=1 \\ 1 & \text{if } n>1 \end{cases} \\ \phi_{n'}(x,y') &= \phi_n(x,\phi_{n'}(x,y)) \end{aligned}$$

Notice that ϕ_1 is addition, ϕ_2 is multiplication, ϕ_3 is exponentiation, etc; i.e., the next function is always obtained by iterating the previous one. For each n, the function $\phi_n(x,x)$ is primitive recursive, but $\phi(x,x,x)$ is not: Ackermann showed that it grows faster than any primitive recursive function. Herbrand viewed the Ackermann function in his [1931a] as finitistically calculable.

2.4 Formalizability and calculability

In lectures and publications from 1921 and 1922, Hilbert and Bernays established the consistency of an elementary part of arithmetic from their new finitist perspective. The work is described together with an *Ansatz* for its extension in [Hilbert, 1923]. They restrict the attention to the quantifier-free part of arithmetic that contains all primitive recursive functions and an induction rule; that part is now called *primitive recursive arithmetic* (*PRA*) and is indeed the system **F*** of Herbrand's discussed below, when the class F of finitist functions consists of exactly the primitive recursive ones.[13]

PRA has a direct finitist justification, and thus there was no programmatic need to establish its consistency. However, the proof was viewed as a stepping-stone towards a consistency proof for full arithmetic and analysis. It is indeed the first sophisticated proof-theoretic argument, transforming arbitrary derivations into configurations of variable-free formulas. The truth-values of these formulas can be effectively determined, because Hilbert and Bernays insist on the calculability of functions and the decidability of relations. Ackermann attempted in his dissertation, published as [Ackermann, 1924], to extend this very argument to analysis. Real difficulties emerged even before the article appeared and the

[13] Tait argues in his [1981] for the identification of finitist arithmetic with *PRA*. This a conceptually coherent position, but I no longer think that it reflects the historical record of considerations and work surrounding Hilbert's Program; cf. also Tait's [2002], the papers by Zach referred to, Ravaglia's Carnegie Mellon Ph.D. thesis, as well as our joint paper [2005].

validity of the result had to be restricted to a part of elementary number theory. The result is obtained also in von Neumann's [1927]. The problem of extending the restricted result was thought then to be a straightforward mathematical one. That position was clearly taken by Hilbert in his Bologna address of 1928, when he claims that the results of Ackermann and von Neumann cover full arithmetic and then asserts that there is an *Ansatz* of a consistency proof for analysis: "This [Ansatz] has been pursued by Ackermann already to such an extent that the remaining task amounts only to proving a purely arithmetic elementary finiteness theorem." (p. 4)

These difficulties were revealed, however, by the incompleteness theorems as "conceptual" philosophical ones. The straightforwardly mathematical consequence of the second incompleteness theorem can be formulated as follows: Under general conditions[14] on a theory T, T proves the conditional $(con_T \to G)$; con_T is the statement expressing the consistency of T, and G is the Gödel sentence. G states its own unprovability and is, by the first incompleteness theorem, not provable in T. Consequently, G would be provable in T, as soon as a finitist consistency proof for T could be formalized in T. That's why the issue of the formalizability of finitist considerations plays such an important role in the emerging discussion between von Neumann, Herbrand and Gödel. At issue was the extent of finitist methods and thus the reach of Hilbert's consistency program. That raises in particular the question, what are the finitistically calculable functions; it is clear that the primitive recursively defined functions are to be included. (Recall the rather general way in which recursive definitions were dicussed in Hilbert's lectures [1921/22].)

Herbrand's own [1931a] is an attempt to harmonize his proof theoretic investigations with Gödel's results. Gödel insisted in his paper that the second incompleteness theorem does not contradict Hilbert's "formalist viewpoint":

> For this viewpoint presupposes only the existence of a consistency proof in which nothing but finitary means of proof is used, and it is conceivable that there exist finitary proofs that *cannot* be expressed in the formalism of P (or of M and A).[15]

Having received the galleys of Gödel's paper, von Neumann writes in a letter of 12 January 1931:

> I absolutely disagree with your view on the formalizability of intuitionism. Certainly, for every formal system there is, as you proved, another formal one that is (already in arithmetic and the lower functional calculus) stronger. But that does not affect intuitionism at all.

[14] The general conditions on T include, of course, the representability conditions for the first theorem and the Hilbert-Bernays derivability conditions for the second theorem.

[15] *Collected Works I*, p. 195. P is a version of the system of *Principia Mathematica*, M the system of set theory introduced by von Neumann, and A classical analysis.

(Note that Herbrand and von Neumann, but also others at the time, use intuitionist as synonymous with finitist; even Gödel did as should be clear from his [1931a].) Denoting first-order number theory by A, analysis by M, and set theory by Z, von Neumann continues:

> Clearly, I cannot prove that every intuitionistically correct *construction* of *arithmetic* is formalizable in A or M or even in Z — for intuitionism is undefined and undefinable. But is it not a fact, that not a single construction of the kind mentioned is known that cannot be formalized in A, and that no living logician is in the position of naming such [[a construction]]? Or am I wrong, and you know an effective intuitionistic arithmetic construction whose formalization in A creates difficulties? If that, to my utmost surprise, should be the case, then the formalization should work in M or Z!

This line of argument was sharpened, when Herbrand wrote to Gödel on 7 April 1931. By then he had discussed the incompleteness phenomena extensively with von Neumann, and he had also read the galleys of [Gödel, 1931]. Herbrand's letter has to be understood, and Gödel in his response quite clearly did, as giving a sustained argument against Gödel's assertion that the second incompleteness theorem does not contradict Hilbert's formalist viewpoint.

Herbrand introduces a number of systems for arithmetic, all containing the axioms for predicate logic with identity and the Dedekind-Peano axioms for zero and successor. The systems are distinguished by the strength of the induction principle and by the class F of finitist functions for which recursion equations are available. The system with induction for all formulas and recursion equations for the functions in F is denoted here by **F**; if induction is restricted to quantifier-free formulas, I denote the resulting system by **F***. The axioms for the elements $f_1, f_2, f_3, \ldots$ in F must satisfy according to Herbrand's letter the following conditions:

1. The defining axioms for f_n contain, besides f_n, only functions of lesser index.
2. These axioms contain only constants and free variables.
3. We must be able to show, by means of intuitionistic proofs, that with these axioms it is possible to compute the value of the functions univocally for each specified system of values of their arguments.

As examples for classes F Herbrand considers the set E_1 of addition and multiplication, as well as the set E_2 of all primitive recursive functions. He asserts that many other functions are definable by his "general schema", in particular, the non-primitive recursive Ackermann function. He also argues that one can construct by diagonalization a finitist function that is not in E, if **E** contains axioms such that "one can always determine, whether or not certain defining axioms [for the elements of E] are among these axioms". It is here that the "double" use of finitist functions — straightforwardly as part of finitist mathematical practice and as a tool to describe formal theories — comes together to allow the definition of additional finitist functions; that is pointed out in Herbrand's letter to Gödel. Indeed, it is quite explicit also in Herbrand's almost immediate reaction to the incompleteness phenomena in his letter to Chevalley from 3 December 1930. (See [Sieg, 1994, 103–4].)

This fact of the open-endedness of any finitist presentation of the concept "finitist function" is crucial for Herbrand's conjecture that one cannot prove that all finitist methods are formalizable in *Principia Mathematica*. But he claims that, as a matter of fact, every finitist proof can be formalized in a system **F***, based on a suitable class F that depends on the given proof, thus in *Principia Mathematica*. Conversely, he insists that every proof in the quantifier-free part of **F*** is finitist. He summarizes his reflections by saying in the letter and with almost identical words in [1931a]:

> It reinforces my conviction that it is impossible to prove that every intuitionistic proof is formalizable in Russell's system, but that a counterexample will never be found. There we shall perhaps be compelled to adopt a kind of logical postulate.

Herbrand's conjectures and claims are completely in line with those von Neumann communicated to Gödel in his letters of November 1930 and January 1931. In the former letter von Neumann wrote:

> I believe that every intuitionistic consideration can be formally copied, because the "arbitrarily nested" recursions of Bernays-Hilbert are equivalent to ordinary transfinite recursions up to appropriate ordinals of the second number class. This is a process that can be formally captured, unless there is an intuitionistically definable ordinal of the second number class that could not be defined formally — which is in my view unthinkable. Intuitionism clearly has no finite axiom system, but that does not prevent its being a part of classical mathematics that does have one. (*Collected Works V*, p. 339)

We know of Gödel's response to von Neumann's dicta not through letters from Gödel, but rather through the minutes of a meeting of the Schlick or Vienna Circle that took place on 15 January 1931. According to these minutes Gödel viewed as questionable the claim that the totality of all intuitionistically correct proofs is contained in *one* formal system. That, he emphasized, is the weak spot in von Neumann's argumentation. (Gödel did respond to von Neumann, but his letters seem to have been lost. The minutes are found in the Carnap Archives of the University of Pittsburgh.)

When answering Herbrand's letter, Gödel makes more explicit his reasons for questioning the formalizability of finitist considerations in a single formal system like *Principia Mathematica*. He agrees with Herbrand on the indefinability of the concept "finitist proof". However, even if one accepts Herbrand's very schematic presentation of finitist methods and the claim that every finitist proof can be formalized in a system of the form **F***, the question remains "whether the intuitionistic proofs that are required in each case to justify the unicity of the recursion axioms are all formalizable in *Principia Mathematica*". Gödel continues:

> Clearly, I do not claim either that it is certain that some finitist proofs are not formalizable in *Principia Mathematica*, even though intuitively I tend toward this assumption. In any case, a finitist proof not formalizable in *Principia Mathematica* would have to be quite extraordinarily complicated, and on this purely practical ground there is very little prospect of finding one; but that, in my opinion, does not alter anything about the possibility in principle.

At this point there is a stalemate between Herbrand's "logical postulate" that no finitist proof outside of *Principia Mathematica* will be found and Gödel's "possibility in principle" that one might find such a proof.

By late December 1933 when he gave an invited lecture to the Mathematical Association of America in Cambridge (Massachusetts), Gödel had changed his views significantly. In the text for his lecture, [Gödel, 1933], he sharply distinguishes intuitionist from finitist arguments, the latter constituting the most restrictive form of constructive mathematics. He insists that the known finitist arguments given by "Hilbert and his disciples" can all be carried out in a certain system **A**. Proofs in **A**, he asserts, "can be easily expressed in the system of classical analysis and even in the system of classical arithmetic, and there are reasons for believing that this will hold for any proof which one will ever be able to construct". This observation and the second incompleteness theorem imply, as sketched above, that classical arithmetic cannot be shown to be consistent by finitist means. The system **A** is similar to the quantifier-free part of Herbrand's system **F***, except that the provable totality for functions in F is not mentioned and that **A** is also concerned with other inductively defined classes.[16] Gödel's reasons for conjecturing that **A** contains all finitist arguments are not made explicit.

Gödel discusses then a theorem of Herbrand's, which he considers to be the most far-reaching among interesting partial results in the pursuit of Hilbert's consistency program. He does so, as if to answer the question "How do current consistency proofs fare?" and formulates the theorem in this lucid and elegant way: "If we

[16] The restrictive characteristics of the system **A** are formulated on pp. 23 and 24 of [1933] and include the requirement that notions have to be decidable and functions must be calculable. Gödel claims that "such notions and functions can always be defined by complete induction". Definition by complete induction is to be understood as definition by recursion, which — for the integers — is not restricted to primitive recursion. The latter claim is supported by the context of the lecture and also by Gödel's remark at the very beginning of section 9 in his Princeton Lectures, where he explains that a version of the Ackermann function is "defined inductively". The actual definition is considered "as an example of a definition by induction with respect to two variables simultaneously". That is followed by the remark, "The consideration of various sorts of functions defined by induction leads to the question what one would mean by 'every recursive function'."

take a theory which is constructive in the sense that each existence assertion made in the axioms is covered by a construction, and if we add to this theory the non-constructive notion of existence and all the logical rules concerning it, e.g., the law of excluded middle, we shall never get into any contradiction." (This implies directly the extension of Hilbert's first consistency result from 1921/22 to the theory obtained from it by adding full classical first order logic, but leaving the induction principle quantifier-free.) Gödel conjectures that Herbrand's method might be generalized, but he emphasizes that "for larger systems containing the whole of arithmetic or analysis the situation is hopeless if you insist upon giving your proof for freedom from contradiction by means of the system **A**". As the system **A** is essentially the quantifier-free part of **F***, it is clear that Gödel now takes Herbrand's position concerning the impact of his second incompleteness theorem on Hilbert's Program.

Nowhere in the correspondence does the issue of *general* computability arise. Herbrand's discussion, in particular, is solely trying to explore the limits that are imposed on consistency proofs by the second theorem. Gödel's response focuses also on that very topic. It seems that he subsequently developed a more critical perspective on the character and generality of his theorems. This perspective allowed him to see a crucial open question and to consider Herbrand's notion of a finitist function as a first step towards an answer. A second step was taken in 1934 when Gödel lectured on his incompleteness theorems at Princeton. There one finds not only an even more concise definition of the class of primitive recursive functions, but also a crucial and revealing remark as to the pragmatic reason for the choice of this class of functions.

The very title of the lectures, *On undecidable propositions of formal mathematical systems*, indicates that Gödel wanted to establish his theorems in greater generality, not just for *Principia Mathematica* and related systems. In the introduction he attempts to characterize "formal mathematical system" by requiring that the rules of inference, and the definitions of meaningful [i.e., syntactically well-formed] formulas and axioms, be "constructive"; Gödel elucidates the latter concept as follows:

> ... for each rule of inference there shall be a finite procedure for determining whether a given formula B is an immediate consequence (by that rule) of given formulas $A_1, \ldots A_n$, and there shall be a finite procedure for determining whether a given formula A is a meaningful formula or an axiom. (p. 346)

That is of course informal and imprecise, mathematically speaking. The issue is addressed in section 7, where Gödel discusses conditions a formal system must satisfy so that the arguments for the incompleteness theorems apply to it. The first of five conditions is this:

> Supposing the symbols and formulas to be numbered in a manner similar to that used for the particular system considered above, then the

> class of axioms and the relation of immediate consequence shall be recursive (i.e., in these lectures, primitive recursive).
>
> This is a precise condition which in practice suffices as a substitute for the unprecise requirement of §1 that the class of axioms and the relation of immediate consequence be constructive. (p. 361)[17]

A principled precise condition for characterizing formal systems in general is needed. Gödel defines in §9 the class of "general recursive functions"; that is Gödel's second step alluded to above and the focus of the next section.

3 RECURSIVENESS AND CHURCH'S THESIS

In Section 2 I described the emergence of a broad concept of calculable function. It arose out of a mathematical practice that was concerned with effectiveness of solutions, procedures and notions; it was also tied in important ways to foundational discussions that took place already in the second half of the 19^{th} century with even older historical roots. I pointed to the sharply differing perspectives of Dedekind and Kronecker. It was the former who formulated in his [1888] the schema of primitive recursion in perfect generality. That all the functions defined in this way are calculable was of course clear, but not the major issue for Dedekind: he established that primitive recursive definitions determine unique functions in his logicist framework. From a constructive perspective, however, these functions have an autonomous significance and were used in the early work of Hilbert and Bernays, but also of Skolem, for developing elementary arithmetic in a deeply Kroneckerian spirit. Hilbert and Bernays viewed this as a part of finitist mathematics, their framework for meta-mathematical studies in general and for consistency proofs in particular.

An inductive specification of the class of primitive recursive functions is found in the *Zwischenbetrachtung* of section 3 in Gödel's [1931] and, even more standardly, in the second section of his [1934]. That section is entitled "Recursive functions and relations." In a later footnote Gödel pointed out that "recursive" in these lectures corresponds to "primitive recursive" as used now. It was a familiar fact by then that there are calculable functions, which are not in the class of primitive recursive functions, with Ackermann's and Sudan's functions being the best-known examples. Ackermann's results were published only in 1928, but they had been discussed extensively already earlier, e.g., in Hilbert's *On the infinite*. Herbrand's schema from 1931 defines a broad class of finitistically calculable functions including the Ackermann function; it turned out to be the starting-point of significant further developments.

Herbrand's schema is a natural generalization of the definition schemata for calculable functions that were known to him and built on the practice of the

[17] In the Postscriptum to [Gödel, 1934] Gödel asserts that exactly this condition can be removed on account of Turing's work.

Hilbert School. It could also be treated easily by the methods for proving the consistency of weak systems of arithmetic Herbrand had developed in his thesis. In a letter to Bernays of 7 April 1931, the very day on which he also wrote to Gödel, Herbrand contrasts his consistency proof with Ackermann's, which he mistakenly attributes to Bernays:

> In my arithmetic the axiom of complete induction is restricted, but one may use a variety of other functions than those that are defined by simple recursion: in this direction, it seems to me, that my theorem goes a little farther than yours [i.e., than Ackermann's].

The point that is implicit in my earlier discussion should be made explicit here and be contrasted with discussions surrounding Herbrand's schema by Gödel and van Heijenoort as to the programmatic direction of the schema[18]: the above is hardly a description of a class of functions that is deemed to be of fundamental significance for the question of "general computability". Rather, Herbrand's remark emphasizes that his schema captures a broader class of finitist functions and should be incorporated into the formal theory to be shown consistent.

Gödel considered the schema, initially and in perfect alignment with Herbrand's view, as a way of partially capturing the constructive aspect of mathematical practice. It is after all the classical theory of arithmetic with Herbrand's schema that is reduced to its intuitionistic version by Gödel in his [1933]; this reductive result showed that intuitionism provides a broader constructive framework than finitism. I will detail the modifications Gödel made to Herbrand's schema when introducing in [1934] the general recursive functions. The latter are the primary topic of this section, and the main issues for our discussion center around Church's Thesis.

3.1 Relative consistency

Herbrand proved in his [1931a], as I detailed above and at the end of section 2.4, the consistency of a system for classical arithmetic that included defining equations for all the finitistically calculable functions identified by his schema, but made the induction principle available only for quantifier-free formulas. In a certain sense that restriction is lifted in Gödel's [1933], where an elementary translation of full classical arithmetic into intuitionistic arithmetic is given. A system for intuitionistic arithmetic had been formulated in [Heyting, 1930a]. Gödel's central claim in the paper is this: *If a formula A is provable in Herbrand's system for classical arithmetic, then its translation A^* is provable in Heyting arithmetic.* A^* is obtained from A by transforming the latter into a classically equivalent formula not containing $\vee, \rightarrow, (\exists)$. The crucial auxiliary lemmata are the following:

[18] Van Heijenoort analyzed the differences between Herbrand's published proposals and the suggestion that had been made, according to [Gödel, 1934], by Herbrand in his letter to Gödel. References to this discussion in light of the actual letter are found in my paper [1994]; see in particular section 3.2 and the Appendix.

(i) For all formulas A^*, Heyting arithmetic proves $\neg\neg A^* \rightarrow A^*$; and

(ii) For all formulas A^* and B^*, Heyting arithmetic proves that $A^* \rightarrow B^*$ is equivalent to $\neg(A^* \& \neg B^*)$

The theorem establishes obviously the consistency of classical arithmetic relative to Heyting arithmetic. If the statement 0=1 were provable in classical arithmetic, then it would be provable in Heyting arithmetic, as (0=1)* is identical to 0=1. From an intuitionistic point of view, however, the principles of Heyting arithmetic can't lead to a contradiction. Gödel concludes his paper by saying (p. 294 in *Collected Works I*):

> The above considerations provide of course an intuitionistic consistency proof for classical arithmetic and number theory. However, the proof is not "finitist" in the sense Herbrand gave to the term, following Hilbert.

This implies a clear differentiation of intuitionistic from finitist mathematics, and the significance of this result cannot be overestimated. Ironically, it provided a basis and a positive direction for modifying Hilbert's Program: exploit in consistency proofs the constructive means of intuitionistic mathematics that go beyond finitist ones. Gödel's result is for that very reason important and was obtained, with a slightly different argument, also by Gentzen. The historical point is made forcefully by Bernays in his contribution on Hilbert to the *Encyclopedia of Philosophy*; the systematic point, and its relation to the further development of proof theory, has been made often in the context of a generalized reductive program in the tradition of the Hilbert school; see, for example, [Sieg and Parsons, 1995] or my [2002].

I discuss Gödel's result for two additional reasons, namely, to connect the specific developments concerning computability with the broader foundational considerations of the time and to make it clear that Gödel was thoroughly familiar with Herbrand's formulation when he gave the definition of "general recursive functions" in his 1934 Princeton Lectures. Herbrand's schema is viewed, in the reductive context, from the standpoint of constructive mathematical practice as opposed to its meta-mathematical use in the description of "formal theories". That is made clear by Gödel's remark, "The definition of number-theoretic functions by recursion is unobjectionable for intuitionism as well (see H_2, 10.03, 10.04). Thus all functions f_i (Axiom Group C) occur also in intuitionistic mathematics, and we consider the formulas defining them to have been adjoined to Heyting's axioms;..."[19] The meta-mathematical, descriptive use will become the focus of our investigation, as the general characterization of "formal systems" takes center stage and is pursued via an explication of "constructive" or "effective" procedures. We will then take on the problem of identifying an appropriate mathematical concept for this informal notion, i.e., issues surrounding Church's or Turing's Thesis. To get a

[19] *Collected Works I*, p. 290. The paper H_2 is [Heyting, 1930a], and the numbers 10.03 and 10.04 refer to not more and not less than the recursion equations for addition.

concrete perspective on the significance of the broad issues, let me mention claims formulated by Church and Gödel with respect to Turing's work, but also point to tensions and questions that are only too apparent.

Church reviewed Turing's *On computable numbers* for the Journal of Symbolic Logic just a few months after its publication. He contrasted Turing's notion for effective calculability (via idealized machines) with his own (via λ-definability) and with Gödel's (via the equational calculus). "Of these [notions]," Church remarked, "the first has the advantage of making the identification with effectiveness in the ordinary (not explicitly defined) sense evident immediately..." Neither in this review nor anywhere else did Church give reasons, why the identification *is* immediately evident for Turing's notion, and why it is *not* for the others. In contrast, Gödel seemed to capture essential aspects of Turing's considerations when making a brief and enigmatic remark in the 1964 postscript to the Princeton Lectures he had delivered thirty years earlier: "Turing's work gives an *analysis* of the concept of 'mechanical procedure'... This concept is *shown* to be equivalent with that of a 'Turing machine'."[20] But neither in this postscript nor in other writings did Gödel indicate the nature of Turing's analysis and prove that the analyzed concept is indeed equivalent to that of a Turing machine.

Gödel underlined the significance of Turing's analysis, repeatedly and emphatically. He claimed, also in [1964], that only Turing's work provided "a precise and unquestionably adequate definition of the general concept of formal system". As a formal system is for Gödel just a mechanical procedure for producing theorems, the adequacy of this definition rests squarely on the correctness of Turing's analysis of mechanical procedures. The latter lays the ground for the most general mathematical formulation and the broadest philosophical interpretation of the incompleteness theorems. Gödel himself had tried to arrive at an adequate concept in a different way, namely, by directly characterizing calculable number theoretic functions more general than primitive recursive ones. As a step towards such a characterization, Gödel introduced in his Princeton Lectures "general recursive functions" via his equational calculus "using" Herbrand's schema. I will now discuss the crucial features of Gödel's definition and contrast it with Herbrand's as discussed in Section 2.4.

3.2 Uniform calculations

In his Princeton Lectures, Gödel strove to make the incompleteness results less dependent on particular formalisms. Primitive recursive definability of axioms and inference rules was viewed as a "precise condition, which in practice suffices as a substitute for the unprecise requirement of §1 that the class of axioms and the relation of immediate consequence be constructive". A notion that would suffice *in principle* was needed, however, and Gödel attempted to arrive at a more general

[20]Gödel's *Collected Works I*, pp. 369–70. The emphases are mine. — In the context of this paper and reflecting the discussion of Church and Gödel, I consider effective and mechanical procedure as synonymous.

notion. Gödel considers the fact that the value of a primitive recursive function can be computed by a finite procedure for each set of arguments as an "important property" and adds in footnote 3:

> The converse seems to be true if, besides recursions according to the scheme (2) [i.e., primitive recursion as given above], recursions of other forms (e.g., with respect to two variables simultaneously) are admitted. This cannot be proved, since the notion of finite computation is not defined, but it can serve as a heuristic principle.

What other recursions might be admitted is discussed in the last section of the Notes under the heading "general recursive functions".

The general recursive functions are taken by Gödel to be those number theoretic functions whose values can be calculated via elementary substitution rules from an extended set of basic recursion equations. This is an extremely natural approach and properly generalizes primitive recursiveness: the new class of functions includes of course all primitive recursive functions and also those of the Ackermann type, defined by nested recursion. Assume, Gödel suggests, you are given a finite sequence $\psi_1, \ldots, \psi_k$ of "known" functions and a symbol ϕ for an "unknown" one. Then substitute these symbols "in one another in the most general fashions" and equate certain pairs of the resulting expressions. If the selected set of functional equations has exactly one solution, consider ϕ as denoting a "recursive" function.[21] Gödel attributes this broad proposal to define "recursive" functions mistakenly to Herbrand and proceeds then to formulate two restrictive conditions for his definition of "general recursive" functions:

(**1**) the l.h.s. of equations is of the form $\phi(\psi_{i_1}(x_1, \ldots, x_n), \ldots, \psi_{i_l}(x_1, \ldots, x_n))$, and

(**2**) for every l-tuple of natural numbers the value of ϕ is "computable in a calculus".

The first condition just stipulates a standard form of certain terms, whereas the important second condition demands that for every l-tuple $k_1, \ldots, k_l$ there is exactly one m such that $\phi(k_1, \ldots, k_l) = m$ is a "derived equation". The set of derived equations is specified inductively via elementary substitution rules; the basic clauses are:

(**A.1**) All numerical instances of a given equation are derived equations;

(**A.2**) All true equalities $\psi_{i_j}(x_1, \ldots, x_n) = m$ are derived equations.

The rules allowing steps from already obtained equations to additional ones are formulated as follows:

[21] Kalmar proved in his [1955] that these "recursive" functions, just satisfying recursion equations, form a strictly larger class than the general recursive ones.

(**R.1**) Replace occurrences of $\psi_{i_j}(x_1,\ldots,x_n)$ by m, if $\psi_{i_j}(x_1,\ldots,x_n) = m$ is a derived equation;

(**R.2**) Replace occurrences of $\phi(x_1,\ldots,x_l)$ on the right-hand side of a derived equation by m, if $\phi(x_1,\ldots,x_l) = m$ is a derived equation.

In addition to restriction (**1**) on the syntactic form of equations, we should recognize with Gödel two novel features in this definition when comparing it to Herbrand's: first, the precise specification of *mechanical* rules for deriving equations, i.e., for carrying out numerical computations; second, the formulation of the *regularity condition* requiring computable functions to be total, but without insisting on a finitist proof. These features were also emphasized by Kleene who wrote with respect to Gödel's definition that "it consists in specifying the form of the equations and the nature of the steps admissible in the computation of the values, and in requiring that for each given set of arguments the computation yield a unique number as value" [Kleene, 1936, 727]. Gödel reemphasized these points in later remarks, when responding to van Heijenoort's inquiry concerning the precise character of Herbrand's suggestion.

In a letter to van Heijenoort of 14 August 1964 Gödel asserts "it was exactly by specifying the rules of computation that a mathematically workable and fruitful concept was obtained". When making this claim Gödel took for granted that Herbrand's suggestion had been "formulated *exactly* as on page 26 of my lecture notes, i.e., without reference to computability". At that point Gödel had to rely on his recollection, which, he said, "is very distinct and was still very fresh in 1934". On the evidence of Herbrand's letter, it is clear that Gödel misremembered. This is not to suggest that Gödel was wrong in viewing the specification of computation rules as extremely important, but rather to point to the absolutely crucial step he had taken, namely, to disassociate general recursive functions from the epistemologically restricted notion of proof that is involved in Herbrand's formulation.

Gödel dropped later the regularity condition altogether and emphasized, "that the precise notion of mechanical procedures is brought out clearly by Turing machines producing partial rather than general recursive functions." At the earlier juncture in 1934 the introduction of the equational calculus with particular computation rules was important for the mathematical development of recursion theory as well as for the underlying conceptual motivation. It brought out clearly, what Herbrand — according to Gödel in his letter to van Heijenoort — had failed to see, namely "that the computation (for all computable functions) proceeds by exactly the same rules". Gödel was right, for stronger reasons than he put forward, when he cautioned in the same letter that Herbrand had *foreshadowed*, but not *introduced*, the notion of a general recursive function. Cf. the discussion in and of [Gödel, 193?] presented in Section 6.1.

Kleene analyzed the class of general recursive functions in his [1936] using Gödel's arithmetization technique to describe provability in the equational calculus. The uniform and effective generation of derived equations allowed Kleene to establish an important theorem that is most appropriately called "Kleene's normal form theorem": *for every recursive function* φ *there are primitive recursive functions* ψ *and* ρ *such that* $\varphi(x_1, \ldots, x_n)$ *equals* $\psi(\varepsilon y.\rho(x_1, \ldots, x_n, y) = 0)$, *where for every* n*-tuple* $x_1, \ldots, x_n$ *there is a* y *such that* $\rho(x_1, \ldots, x_n, y) = 0$. The latter equation expresses that y is (the code of) a computation from the equations that define φ for the arguments $x_1, \ldots, x_n$. The term $\varepsilon y.\rho(x_1, \ldots, x_n, y) = 0$ provides the smallest y, such that $\rho(x_1, \ldots, x_n, y) = 0$, if there is a y for the given arguments, and it yields 0 otherwise. Finally, the function ψ considers the last equation in the selected computation and determines the numerical value of the term on the r.h.s of that equation — which is a numeral and represents the value of φ for given arguments $x_1, \ldots, x_n$. This theorem (or rather its proof) is quite remarkable: the ease with which "it" allows to establish equivalences of different computability formulations makes it plausible that some stable notion has been isolated. What is needed for the proof is only that the inference or computation steps are all primitive recursive. Davis observes in his [1982, 11] quite correctly, "The theorem has made equivalence proofs for formalisms in recursive function theory rather routine, ..." The informal understanding of the theorem is even more apparent from Kleene's later formulation involving his T-predicate and result-extracting function U; see for example his *Introduction to Metamathematics*, p. 288 ff.

Hilbert and Bernays had introduced in the first volume of their *Grundlagen der Mathematik* a μ-operator that functioned in just the way the ε-operator did for Kleene. The μ-notation was adopted later also by Kleene and is still being used in computability theory. Indeed, the μ-operator is at the heart of the definition of the class of the so-called "μ-recursive functions". They are specified inductively in the same way as the primitive recursive functions, except that a third closure condition is formulated: if $\rho(x_1, \ldots, x_n, y)$ is μ-recursive and for every n-tuple $x_1, \ldots, x_n$ there is a y such that $\rho(x_1, \ldots, x_n, y) = 0$, then the function $\theta(x_1, \ldots, x_n)$ given by $\mu y.\rho(x_1, \ldots, x_n, y) = 0$ is also μ-recursive. The normal form theorem is the crucial stepping stone in proving that this class of functions is co-extensional with that of Gödel's general recursive ones.

This result was actually preceded by the thorough investigation of λ-definability by Church, Kleene and Rosser.[22] Kleene emphasized in his [1987, 491], that the approach to effective calculability through λ-definability had "quite independent roots (motivations)" and would have led Church to his main results "even if Gödel's paper [1931] had not already appeared". Perhaps Kleene is right, but I doubt it. The flurry of activity surrounding Church's *A set of postulates for the foundation of logic* (published in 1932 and 1933) is hardly imaginable without knowledge of Gödel's work, in particular not without the central notion of representability and, as Kleene points out, the arithmetization of meta-mathematics. The Princeton group knew of Gödel's theorems since the fall of 1931 through a lecture of von Neumann's. Kleene reports in [1987, 491], that through this lecture "Church and the rest of us first learned of Gödel's results". The centrality of representability

[22] For analyses of the quite important developments in Princeton from 1933 to 1937 see [Davis, 1982] and my [1997], but of course also the accounts given by Kleene and Rosser. [Crossley, 1975a] contains additional information from Kleene about this time.

for Church's considerations comes out clearly in his lecture on Richard's paradox given in December 1933 and published as [Church, 1934]. According to [Kleene, 1981, 59] Church had formulated his thesis for λ-definability already in the fall of 1933; so it is not difficult to read the following statement as an extremely cautious statement of the thesis:

> ... it appears to be possible that there should be a system of symbolic logic containing a formula to stand for every definable function of positive integers, and I fully believe that such systems exist.[23]

One has only to realize from the context that (i) 'definable' means 'constructively definable', so that the value of the functions can be calculated, and (ii) 'to stand for' means 'to represent'.

A wide class of calculable functions had been characterized by the concept introduced by Gödel, a class that contained all known effectively calculable functions. Footnote 3 of the Princeton Lectures I quoted earlier seems to express a form of Church's Thesis. In a letter to Martin Davis of 15 February 1965, Gödel emphasized that no formulation of Church's Thesis is implicit in that footnote. He wrote:

> ... The conjecture stated there only refers to the equivalence of "finite (computation) procedure" and "recursive procedure". However, I was, at the time of these lectures, not at all convinced that my concept of recursion comprises all possible recursions; and in fact the equivalence between my definition and Kleene's ... is not quite trivial.

At that time in early 1934, Gödel was equally unconvinced by Church's proposal to identify effective calculability with λ-definability; he called the proposal "thoroughly unsatisfactory". That was reported by Church in a letter to Kleene dated 29 November 1935 (and quoted in [Davis, 1982, 9]).

Almost a year later, Church comes back to his proposal in a letter to Bernays dated 23 January 1935; he conjectures that the λ-calculus may be a system that allows the representability of all constructively defined functions:

> The most important results of Kleene's thesis concern the problem of finding a formula to represent a given intuitively defined function of positive integers (it is required that the formula shall contain no other symbols than λ, variables, and parentheses). The results of Kleene are so general and the possibilities of extending them apparently so unlimited that one is led to conjecture that a formula can be found to represent any particular constructively defined function of positive integers whatever. It is difficult to prove this conjecture, however, or even to state it accurately, because of the difficulty in saying precisely what is meant by "constructively defined". A vague description can be

[23] [Church, 1934, 358]. Church assumed, clearly, the converse of this claim.

> given by saying that a function is constructively defined if a method can be given by which its values could be actually calculated for any particular positive integer whatever.

When Church wrote this letter, it was known in his group that all general recursive functions are λ-definable; Church established in collaboration with Kleene the converse by March 1935. (Cf. [Sieg, 1997, 156].) This mathematical equivalence result and the quasi-empirical adequacy through Kleene's and Rosser's work provided the background for the public articulation of Church's Thesis in the 1935 abstract to be discussed in the next subsection. The elementary character of the steps in computations made the normal form theorem and the equivalence argument possible. In the more general setting of his 1936 paper, Church actually tried to show that every informally calculable number theoretic function is indeed general recursive.

3.3 Elementary steps

Church, Kleene and Rosser had thoroughly investigated Gödel's notion and its connection with λ-definability by the end of March 1935; Church announced his thesis in a talk contributed to the meeting of the American Mathematical Society in New York City on 19 April 1935. I quote the abstract of the talk in full.

> Following a suggestion of Herbrand, but modifying it in an important respect, Gödel has proposed (in a set of lectures at Princeton, N.J., 1934) a definition of the term *recursive function*, in a very general sense. In this paper a definition of *recursive function of positive integers* which is essentially Gödel's is adopted. And it is maintained that the notion of an effectively calculable function of positive integers should be identified with that of a recursive function, since other plausible definitions of effective calculability turn out to yield notions that are either equivalent to or weaker than recursiveness. There are many problems of elementary number theory in which it is required to find an effectively calculable function of positive integers satisfying certain conditions, as well as a large number of problems in other fields which are known to be reducible to problems in number theory of this type. A problem of this class is the problem to find a complete set of invariants of formulas under the operation of conversion (see abstract 41.5.204). It is proved that this problem is unsolvable, in the sense that there is no complete set of effectively calculable invariants.

General recursiveness served, perhaps surprisingly, as the rigorous concept in this first published formulation of Church's Thesis. The surprise vanishes, however, when Rosser's remark in his [1984] about this period is seriously taken into account: "Church, Kleene, and I each thought that general recursivity seemed to embody the idea of effective calculability, and so each wished to show it equivalent to λ-definability" (p. 345). Additionally, when presenting his [1936a] to the American

Mathematical Society on 1 January 1936, Kleene made these introductory remarks (on p. 544): "The notion of a recursive function, which is familiar in the special cases associated with primitive recursions, Ackermann-Péter multiple recursions, and others, has received a general formulation from Herbrand and Gödel. The resulting notion is of especial interest, since the intuitive notion of a 'constructive' or 'effectively calculable' function of natural numbers can be identified with it very satisfactorily." λ-definability was not even mentioned.

In his famous 1936 paper *An unsolvable problem of elementary number theory* Church described the form of number theoretic problems to be shown unsolvable and restated his proposal for identifying the class of effectively calculable functions with a precisely defined class:

> There is a class of problems of elementary number theory which can be stated in the form that it is required to find an effectively calculable function f of n positive integers, such that $f(x_1, x_2, \ldots, x_n) = 2$ is a necessary and sufficient condition for the truth of a certain proposition of elementary number theory involving $x_1, x_2, \ldots, x_n$ as free variables.
> ...
>
> The purpose of the present paper is to propose a definition of effective calculability which is thought to correspond satisfactorily to the somewhat vague intuitive notion in terms of which problems of this class are often stated, and to show, by means of an example, that not every problem of this class is solvable. [f is the characteristic function of the proposition; that 2 is chosen to indicate 'truth' is, as Church remarked, accidental and non-essential.] (pp. 345–6)

Church's arguments in support of his proposal used again recursiveness; the fact that λ-definability was an equivalent concept added "... to the strength of the reasons adduced below for believing that they [these precise concepts] constitute as general a characterization of this notion [i.e. effective calculability] as is consistent with the usual intuitive understanding of it." (footnote 3, p. 90) Church claimed that those reasons, to be presented and examined in the next paragraph, justify the identification "so far as positive justification can ever be obtained for the selection of a formal definition to correspond to an intuitive notion". (p. 100) Why was there a satisfactory correspondence for Church? What were his reasons for believing that the most general characterization of effective calculability had been found?

To give a deeper analysis Church pointed out, in section 7 of his paper, that two methods to characterize effective calculability of number-theoretic functions suggest themselves. The first of these methods uses the notion of "algorithm", and the second employs the notion of "calculability in a logic". He argues that neither method leads to a definition that is more general than recursiveness. Since these arguments have a parallel structure, I discuss only the one pertaining to the second method. Church considers a logic **L**, that is a system of symbolic logic whose language contains the equality symbol =, a symbol { }( ) for the application of a unary function symbol to its argument, and numerals for the positive integers. For unary functions F he gives the definition:

> F is *effectively calculable* if and only if there is an expression f in the logic **L** such that: $\{f\}(\mu) = \nu$ is a theorem of **L** iff $F(m) = n$; here, μ and ν are expressions that stand for the positive integers m and n.

Church claims that F is recursive, assuming that **L** satisfies certain conditions which amount to requiring the theorem predicate of **L** to be recursively enumerable. Clearly, for us the claim then follows immediately by an unbounded search.

To argue for the recursive enumerability of **L**'s theorem predicate, Church starts out by formulating conditions *any* system of logic has to satisfy if it is "to serve at all the purposes for which a system of symbolic logic is usually intended". These conditions, Church notes in footnote 21, are "substantially" those from Gödel's Princeton Lectures for a formal mathematical system, I mentioned at the end of section 2.4. They state that (**i**) each rule must be an effectively calculable operation, (**ii**) the set of rules and axioms (if infinite) must be effectively enumerable, and (**iii**) the relation between a positive integer and the expression which stands for it must be effectively determinable. Church supposes that these conditions can be "interpreted" to mean that, via a suitable Gödel numbering for the expressions of the logic, (**i′**) each rule must be a recursive operation, (**ii′**) the set of rules and axioms (if infinite) must be recursively enumerable, and (**iii′**) the relation between a positive integer and the expression which stands for it must be recursive. The theorem predicate is then indeed recursively enumerable; but the crucial interpretative step is not argued for at all and thus seems to depend on the very claim that is to be established.

Church's argument in support of the thesis may appear to be viciously circular; but that would be too harsh a judgment. After all, the general concept of calculability is explicated by that of derivability in a logic, and Church uses (**i′**) to (**iii′**) to sharpen the idea that in a logical formalism one operates with an effective notion of immediate consequence.[24] The thesis is consequently appealed to only in a more special case. Nevertheless, it is precisely here that we encounter the major stumbling block for Church's analysis, and that stumbling block was quite clearly seen by Church. To substantiate the latter observation, let me modify a remark Church made with respect to the first method of characterizing effectively calculable functions: *If this interpretation* [what I called the "crucial interpretative step" in the above argument] *or some similar one is not allowed, it is difficult to see how the notion of a system of symbolic logic can be given any exact meaning at all.*[25] Given the crucial role this remark plays, it is appropriate to view and to

[24] Compare footnote 20 on p. 101 of [Church, 1936] where Church remarks: "In any case where the relation of immediate consequence is recursive it is possible to find a set of rules of procedure, equivalent to the original ones, such that each rule is a (one-valued) recursive operation, and the complete set of rules is recursively enumerable."

[25] The remark is obtained from footnote 19 of [Church, 1936, 101] by replacing "an algorithm" by "a system of symbolic logic".

formulate it as a normative requirement:

Church's central thesis. The steps of any effective procedure (governing derivations of a symbolic logic) must be recursive.

If this central thesis is accepted and a function is defined to be effectively calculable if, and only if, it is calculable in a logic, then what Robin Gandy called Church's "step-by-step argument" proves that all effectively calculable functions are recursive. These considerations can be easily adapted to Church's first method of characterizing effectively calculable functions via algorithms and provide another perspective for the "selection of a formal definition to correspond to an intuitive notion". The detailed reconstruction of Church's argument pinpoints the crucial difficulty and shows, first of all, that Church's methodological attitude is quite sophisticated and, secondly, that at this point in 1936 there is no major difference from Gödel's position. (A rather stark contrast is painted in [Davis, 1982] as well as in [Shapiro, 1991] and is quite commonly assumed.) These last points are supported by the directness with which Church recognized, in writing and early in 1937, the importance of Turing's work as making the identification of effectiveness and (Turing) computability "immediately evident".

3.4 Absoluteness

How can Church's Thesis be supported? — Let me first recall that Gödel defined the class of general recursive functions after discussion with Church and in response to Church's "thoroughly unsatisfactory" proposal to identify the effectively calculable functions with the λ-definable ones. Church published the thesis, as we saw, only after having done more mathematical work, in particular, after having established with Kleene the equivalence of general recursiveness and λ-definability. Church gives then two reasons for the thesis, namely, (i) the quasi-empirical observation that all known calculable functions can be shown to be general recursive, the *argument from coverage* and (ii) the mathematical fact of the equivalence of two differently motivated notions, the *argument from confluence*. A third reason comes directly from the 1936 paper and was discussed in the last subsection, (iii) the step-by-step *argument from a core conception*.

Remark. There are additional arguments of a more mathematical character in the literature. For example, in the Postscriptum to [1934] Gödel asserts that the question raised in footnote 3 of the Princeton Lectures, whether his concept of recursion comprises all possible recursions, can be "answered affirmatively" for recursiveness as given in section 10 "which is equivalent with general recursiveness as defined today". As to the contemporary definition he seems to point to μ-recursiveness. How could that *definition* convince Gödel that all possible recursions are captured? How could the *normal form theorem*, as Davis suggests in his [1982, 11], go "a considerable distance towards convincing Gödel" that all possible recursions are comprised by his concept of recursion? It seems to me that arguments answering these questions require crucially an appeal to Church's central thesis and are essentially reformulations of his semi-circular argument. That holds also for the appeal to the recursion theorem[26] in *Introduction to Metamathematics*, p. 352, when Kleene argues "Our methods ... are now developed to the point where they seem adequate for handling any effective definition of a function which might be proposed." After all, in the earlier discussion on p. 351 Kleene asserts: "We now have a general kind of 'recursion', in which the value $\varphi(x_1, \ldots, x_n)$ can be expressed as depending on other values of the same function in a quite arbitrary manner, provided only that the rule of dependence is describable by previously treated effective methods." Thus, to obtain a mathematical result, the "previously treated effective methods" must be identified via Church's central thesis with recursive ones. (End of Remark.)

All these arguments are in the end unsatisfactory. The quasi-empirical observation could be refuted tomorrow, as we might discover a function that is calculable, but not general recursive. The mathematical fact by itself is not convincing, as the ease with which the considerations underlying the proof of the normal form theorem allow one to prove equivalences shows a deep family resemblance of the different notions. The question, whether one or any of the rigorous notions corresponds to the informal concept of effective calculability, has to be answered independently. Finally, as to the particular explication via the core concept "calculability in a logic", Church's argument appeals semi-circularly to a restricted version of the thesis. A conceptual reduction has been achieved, but a mathematically convincing result only with the help of the central thesis. Before discussing Post's and Turing's reflections concerning calculability in the next section, I will look at important considerations due to Gödel and Hilbert and Bernays, respectively.

The concept used in Church's argument is extremely natural for number theoretic functions and is directly related to "Entscheidungsdefinitheit" for relations and classes introduced by Gödel in his [1931] as well as to the representability of functions used in his Princeton Lectures. The rules of the equational calculus allow the mechanical computation of the values of calculable functions; they must be contained in *any* system S that is adequate for number theory. Gödel made an important observation in the addendum to his brief 1936 note *On the length of proofs*. Using the general notion "f is computable in a formal system S" he considers a hierarchy of systems S_i (of order $i, 1 \leq i$) and observes that this notion of computability is independent of i in the following sense: If a function is computable in any of the systems S_i, possibly of transfinite order, then it is already computable in S_1. "Thus", Gödel concludes, "the notion 'computable' is in a certain sense 'absolute,' while almost all meta-mathematical notions otherwise known (for example, provable, definable, and so on) quite essentially depend upon the system adopted." For someone who stressed the type-relativity of provability as strongly as Gödel, this was a very surprising insight.

At the Princeton Bicentennial Conference in 1946 Gödel stressed the special

[26]In [Crossley, 1975a, 7], Kleene asserts that he had proved this theorem before June of 1935.

importance of general recursiveness or Turing computability and emphasized (*Collected Works II*, p. 150):

> It seems to me that this importance is largely due to the fact that with this concept one has for the first time succeeded in giving an *absolute* definition of an interesting epistemological notion, i.e., one not depending on the formalism chosen.

In the footnote added to this remark in 1965, Gödel formulated the mathematical fact underlying his claim that an absolute definition had been obtained, namely, "To be more precise: a function of integers is computable in any formal system containing arithmetic if and only if it is computable in arithmetic, where a function f is called computable in S if there is in S a computable term representing f." Thus not just higher-type extensions are considered now, but any theory that contains arithmetic, for example set theory. Tarski's remarks at this conference, only recently published in [Sinaceur, 2000], make dramatically vivid, how important the issue of the "intuitive adequacy" of general recursiveness was taken to be. The significance of his 1935 discovery was described by Gödel in a letter to Kreisel of 1 May 1968: "That my [incompleteness] results were valid for all possible formal systems began to be plausible for me (that is since 1935) only because of the *Remark* printed on p. 83 of 'The Undecidable' ... But I was completely convinced only by Turing's paper."[27]

If Gödel had been completely convinced of the adequacy of this notion at that time, he could have established the unsolvability of the decision problem for first-order logic. Given that mechanical procedures are exactly those that can be computed in the system S_1 or any other system to which Gödel's Incompleteness Theorem applies, the unsolvability follows from Theorem IX of [Gödel, 1931]. The theorem states that there are formally undecidable problems of predicate logic; it rests on the observation made by Theorem X that every sentence of the form $(\forall x)F(x)$, with F primitive recursive, can be shown in S_1 to be equivalent to the question of satisfiability for a formula of predicate logic. (This last observation has to be suitably extended to general recursiveness.)

Coming back to the conclusion Gödel drew from the absoluteness, he is right that the details of the formalisms extending arithmetic do not matter, but it is crucial that we are dealing with formalisms at all; in other words, a precise aspect of the unexplicated *formal* character of the extending theories has to come into play, when arguing for the absoluteness of the concept computability. Gödel did not prove that computability is an absolute concept, neither in [1946] nor in the earlier note. I conjecture that he must have used considerations similar to those for the proof of Kleene's normal form theorem in order to convince himself of the claim. The absoluteness was achieved then only relative to some effective

[27] In [Odifreddi, 1990, 65]. The content of Gödel's note was presented in a talk on June 19, 1935. See [Davis, 1982, 15, footnote 17] and [Dawson, 1986, 39]. "Remark printed on p. 83" refers to the remark concerning absoluteness that Gödel added in proof (to the original German publication) and is found in [Davis, 1965, 83].

description of the "formal" systems S and the stumbling block shows up exactly here. If my conjecture is correct, then Gödel's argument is completely parallel to Church's contemporaneous step-by-step argument for the co-extensiveness of effective calculability and general recursiveness. Church required, when explicating effective calculability as calculability in logical calculi, the inferential steps in such calculi not only to be effective, but to be general recursive. Some such condition is also needed for completing Gödel's argument.

3.5 Reckonable functions

Church's and Gödel's arguments contain a hidden and semi-circular condition on "steps", a condition that allows their parallel arguments to go through. This step-condition was subsequently moved into the foreground by Hilbert and Bernays's marvelous analysis of "calculations in deductive formalisms". However, before discussing that work in some detail, I want to expose some broad considerations by Church in a letter from 8 June 1937 to the Polish logician Josef Pepis. These considerations (also related in a letter to Post on the same day) are closely connected to Church's explication in his [1936]; they defend the central thesis in an indirect way and show how close his general conceptual perspective was to Gödel's.

In an earlier letter to Church, Pepis had described his project of constructing a number theoretic function that is effectively calculable, but not general recursive. Church explained in his response why he is "extremely skeptical". There is, he asserts, a minimal condition for a function f to be effectively calculable and "if we are not agreed on this then our ideas of effective calculability are so different as to leave no common ground for discussion". This minimal condition is formulated as follows: for every positive integer a there must exist a positive integer b such that the proposition $f(a) = b$ has a "valid proof" in mathematics. Indeed, Church argues, all existing mathematics is formalizable in *Principia Mathematica* or in one of its known extensions; consequently there must be a formal proof of a suitably chosen formal proposition. If f is not general recursive the considerations of [Church, 1936] ensure that for every definition of f within the language of *Principia Mathematica* there exists a positive integer a such that for no b the formal proposition corresponding to $f(a) = b$ is provable in *Principia Mathematica*. Church claims that this holds not only for all known extensions, but for "any system of symbolic logic whatsoever which to my knowledge has ever been proposed". To respect this quasi-empirical fact and satisfy the above minimal condition, one would have to find "an utterly new principle of logic, not only never before formulated, but also never before actually used in a mathematical proof".

Moreover, and here is the indirect appeal to the recursivity of steps, the new principle "must be of so strange, and presumably complicated, a kind that its metamathematical expression as a rule of inference was not general recursive", and one would have to scrutinize the "alleged effective applicability of the principle with considerable care". The dispute concerning a proposed effectively calculable, but non-recursive function would thus center for Church around the required new principle and its effective applicability as a rule of inference, i.e., what I called Church's central thesis. If the latter is taken for granted (implicitly, for example, in Gödel's absoluteness considerations), then the above minimal understanding of effective calculability and the quasi-empirical fact of formalizability block the construction of such a function. This is not a completely convincing argument, as Church admits, but does justify his extreme skepticism of Pepis's project. Church states "this [skeptical] attitude is of course subject to the reservation that I may be induced to change my opinion after seeing your work". So, in a real sense Church joins Gödel in asserting that in any "formal theory" (extending *Principia Mathematica*) only general recursive functions can be computed.

Hilbert and Bernays provide in the second supplement[28] to *Grundlagen der Mathematik II* mathematical underpinnings for Gödel's absoluteness claim and Church's arguments *relative* to their recursiveness conditions ("Rekursivitäts-bedingungen"). They give a marvelous conceptual analysis and establish independence from particular features of formalisms in an even stronger sense than Gödel. The core notion of *calculability in a logic* is made directly explicit and a number-theoretic function is said to be reckonable ("regelrecht auswertbar") just in case it is computable (in the above sense) in *some* deductive formalism. Deductive formalisms must satisfy, however, three recursiveness conditions. The crucial one is an analogue of Church's central thesis and requires that the theorems of the formalism can be enumerated by a primitive recursive function or, equivalently, that the proof-predicate is primitive recursive. Then it is shown that a special number theoretic formalism (included in Gödel's S_1) suffices to compute the reckonable functions, and that the functions computable in this particular formalism are exactly the general recursive ones. Hilbert and Bernays's analysis is a natural capping of the development from *Entscheidungsdefinitheit* to an absolute notion of computability, because it captures the informal notion of rule-governed evaluation of number theoretic functions and explicitly isolates appropriate restrictive conditions. But this analysis does *not* overcome the major stumbling block, it puts it rather in plain view.

The conceptual work of Gödel, Church, Kleene and Hilbert and Bernays had intimate historical connections and is still of genuine and deep interest. It explicated calculability of functions by *one core notion*, namely, computability of their values in a deductive formalism via restricted elementary rules. But no one gave convincing reasons for the proposed restrictions on the steps permitted in computations. This issue was not resolved along Gödelian lines by generalizing recursions, but by a quite different approach due to Alan Turing and, to some extent, Emil Post. I reported in subsection 3.1 on Gödel's assessment of Turing's work in the Postscriptum to the 1934 Princeton Lectures. That Postscriptum was written on 3 June 1964; a few months earlier, on 28 August 1963, Gödel had formulated a brief note for the publication of the translation of his [1931] in [van Heijenoort, 1967]. That note is reprinted in *Collected Works I* (p. 195):

[28] The supplement is entitled, "Eine Präzisierung des Begriffs der berechenbaren Funktion und der Satz von Church über das Entscheidungsproblem."

> In consequence of later advances, in particular of the fact that due to A. M. Turing's work a precise and unquestionably adequate definition of the general notion of formal system can now be given, a completely general version of Theorems VI and XI is now possible. That is, it can be proved rigorously that in *every* consistent formal system that contains a certain amount of finitary number theory there exist undecidable arithmetic propositions and that, moreover, the consistency of any such system cannot be proved in the system.

To the first occurrence of "formal system" in this note Gödel attached a most informative footnote and suggested in it that the term "formal system" should never be used for anything but this notion. For example, the transfinite iterations of formal systems he had proposed in his contribution to the Princeton Bicentennial are viewed as "something radically different from formal systems in the proper sense of the term". The properly formal systems have the characteristic property "that reasoning in them, in principle, can be completely replaced by mechanical devices". That connects back to the remark he had made in [1933a] concerning the formalization of mathematics. The question is, what is it about Turing's notion that makes it an "unquestionably adequate definition of the general notion of formal system"? My contention is that a dramatic shift of perspective overcame the stumbling block for a fundamental conceptual analysis. Let us see what that amounts to: Turing's work is *the* central topic of the next section.

4 COMPUTATIONS AND COMBINATORY PROCESSES

We saw in the previous section that the work of Gödel, Church, Kleene and Hilbert and Bernays explicated calculability of number-theoretic functions as computability of their values in some deductive formalism via elementary steps. Church's direct argument for his thesis appeals to the central thesis asserting that the elementary steps in a computation (or deduction) should be recursive. There is no reason given, why that is a correct or motivated requirement. However, if the central thesis is accepted, then every effectively calculable function is indeed general recursive.

In some sense of elementary, the steps in deductive formalisms are *not* elementary at all. Consider Gödel's equational calculus contained in all of them: it allows the substitution of variables by arbitrary numerals in one step, and arbitrarily complex terms can be replaced by their numerical values, again, in one step. In general, a human calculator cannot carry out such mechanical steps without subdividing them into more basic ones. It was a dramatic shift of perspective, when Turing and Post formulated the most basic mechanical steps that underlie the effective determination of values of number-theoretic functions, respectively the execution of combinatory processes, and that can be carried out by a human computing agent. This shift of perspective made for real progress; it is contiguous with the other work, but it points the way towards overcoming, through Turing's reflections, the stumbling block for a fundamental conceptual analysis.

In the first subsection, *Machines and workers*, I present the mechanical devices or machines Turing introduced, and I'll discuss Post's human workers who operate robot-like in a "symbol space" of marked and unmarked boxes, carrying out extremely simple actions. It is perhaps surprising that Turing's model of computation, developed independently in the same year, is "identical". In contrast to Post, Turing investigated his machines systematically; that work resulted in the discovery of the universal machine, the proof of the unsolvability of the halting problem and, what is considered to be, the definitive resolution of the *Entscheidungsproblem*.

The contrast between the methodological approaches Post and Turing took is *prima facie* equally surprising, if not even more remarkable. For Post it is a "working hypothesis" that all combinatory processes can be effected by the worker's actions, and it is viewed as being in need of continual verification. Turing took the calculations of human computers as the starting-point of a detailed analysis to uncover the underlying symbolic operations, appealing crucially to the agent's sensory limitations. These operations are so basic that they cannot be further subdivided and essentially *are* the operations carried out by Turing machines. The general restrictive features can be formulated as *boundedness* and *locality* conditions. The analysis is the topic of section 4.2 entitled *Mechanical computors*.

Turing's reductive analysis will be critically examined in section 4.3 under the heading *Turing's central thesis*. Using Post's later presentation of Turing machines we can simplify and sharpen the restrictive conditions, but also return to the purely symbolic operations required for the general issues that were central before attention focused on the effective calculability of number theoretic functions. Here we are touching on the central reason why Turing's analysis is so appropriate and leads to an adequate notion. However, Turing felt that his arguments were mathematically unsatisfactory and thought, as late as 1954, that they had to remain so. Before addressing this pivotal point in Section 5, I am going to discuss in subsection 4.5 Church's "machine interpretation" of Turing's work, but also Gandy's proposal to characterize machine computability. Following Turing's broad approach, Gandy investigated in his [1980] the computations of machines or, to indicate better the scope of that notion, of "discrete mechanical devices". According to Gandy, machines can, in particular, carry out parallel computations. In spite of the great generality of his notion, Gandy was able to show that any machine computable function is also Turing computable.

This section is focused on a sustained conceptual analysis of human computability and contrasts it briefly with that of machine computability. Here lies the key to answering the question, "What distinguishes Turing's proposal so dramatically from Church's?" After all, the naïve examination of Turing machines hardly produces the conviction that Turing computability is provably equivalent to an analyzed notion of mechanical procedure (as Gödel claimed) or makes it immediately evident that Turing computability should be identified with effectiveness in the ordinary sense (as Church asserted). A tentative answer is provided; but we'll see that a genuine methodological problem remains. It is addressed in Section 5.

4.1 Machines and workers

The list of different notions in the argument from confluence includes, of course, Turing computability. Though confluence is at issue, there is usually an additional remark that Turing gave in his [1936] the most convincing analysis of effective calculability, and that his notion is truly adequate. What is the notion of computation that is being praised? — In the next few paragraphs I will describe a two-letter Turing machine, following [Davis, 1958] rather than Turing's original presentation. (The differences are discussed in Kleene's *Introduction to Metamathematics*, p. 361, where it is also stated that this treatment "is closer in some respects to [Post, 1936]".)

A Turing machine consists of a finite, but potentially infinite tape. The tape is divided into squares, and each square may carry a symbol from a finite alphabet, say, just the two-letter alphabet consisting of 0 and 1. The machine is able to scan one square at a time and perform, depending on the content of the observed square and its own internal state, one of four operations: print 0, print 1, or shift attention to one of the two immediately adjacent squares. The operation of the machine is given by a finite list of commands in the form of quadruples $q_i s_k c_l q_m$ that express the following: If the machine is in internal state q_i and finds symbol s_k on the square it is scanning, then it is to carry out operation c_l and change its state to q_m. The deterministic character of the machine's operation is guaranteed by the requirement that a program must not contain two different quadruples with the same first two components.

Gandy in his [1988] gave a lucid informal description of a Turing machine computation without using internal states or, as Turing called them, m-configurations: "The computation proceeds by discrete steps and produces a record consisting of a finite (but unbounded) number of cells, each of which is either blank or contains a symbol from a finite alphabet. At each step the action is local and is locally determined, according to a finite table of instructions" (p. 88). How Turing avoids the reference to internal states will be discussed below; why such a general formulation is appropriate will be seen in section 4.3.

For the moment, however, let me consider the Turing machines I just described. Taking for granted a representation of natural numbers in the two-letter alphabet and a straightforward definition of when to call a number-theoretic function *Turing computable*, I put the earlier remark before you as a question: Does this notion provide "an unquestionably adequate definition of the general concept of formal system"? Is it even plausible that every effectively calculable function is Turing computable? It seems to me that a naïve inspection of the restricted notion of Turing computability should lead to "No!" as a tentative answer to the second and, thus, to the first question. However, a systematic development of the theory of Turing computability convinces one quickly that it is a powerful notion.

One goes almost immediately beyond the examination of particular functions and the writing of programs for machines computing them; instead, one considers machines corresponing to operations that yield, when applied to computable functions, other functions that are again computable. Two such functional operations are crucial, namely, composition and minimization. Given these operations and the Turing computability of a few simple initial functions, the computability of all general recursive functions follows. This claim takes for granted Kleene's 1936 proof of the equivalence between general recursiveness and μ-recursiveness. Since Turing computable functions are readily shown to be among the μ-recursive ones, it seems that we are now in exactly the same position as before with respect to the evidence for Church's Thesis. This remark holds also for Post's model of computation.

Post's combinatory processes are generated by computation steps "identical" with Turing's; Post's model was published in the brief 1936 note, *Finite combinatory processes — Formulation 1.* Here we have a worker who operates in a *symbol space* consisting of

> a two way infinite sequence of spaces or boxes, i.e., ordinally similar to the series of integers .... The problem solver or worker is to move and work in this symbol space, being capable of being in, and operating in but one box at a time. And apart from the presence of the worker, a box is to admit of but two possible conditions, i.e., being empty or unmarked, and having a single mark in it, say a vertical stroke.[29]

The worker can perform a number of *primitive acts*, namely, make a vertical stroke $[V]$, erase a vertical stroke $[E]$, move to the box immediately to the right $[M_r]$ or to the left $[M_l]$ (of the box he is in), and determine whether the box he is in is marked or not $[D]$. In carrying out a particular combinatory process the worker begins in a special box (the *starting point*) and then follows directions from a finite, numbered sequence of instructions. The i-th direction, i between 1 and n, is in one of the following forms: (1) carry out act V, E, M_r, or M_l and then follow direction j_i, (2) carry out act D and then, depending on whether the answer was positive or negative, follow direction j_i' or j_i''. (Post has a special stop instruction, but that can be replaced by stopping, conventionally, in case the number of the next direction is greater than n.) Are there intrinsic reasons for choosing Formulation 1, except for its simplicity and Post's expectation that it will turn out to be equivalent to general recursiveness? An answer to this question is not clear (from Post's paper), and the claim that psychological fidelity is aimed for seems quite opaque. Post writes at the very end of his paper:

> The writer expects the present formulation to turn out to be equivalent to recursiveness in the sense of the Gödel–Church development. Its purpose, however, is not only to present a system of a certain logical potency but also, in its restricted field, of psychological fidelity. In the

[29] [Post, 1936, 289]. Post remarks that the infinite sequence of boxes can be replaced by a potentially infinite one, expanding the finite sequence as necessary.

> latter sense wider and wider formulations are contemplated. On the other hand, our aim will be to show that all such are logically reducible to formulation 1. We offer this conclusion at the present moment as a *working hypothesis*. And to our mind such is Church's identification of effective calculability with recursiveness. (p. 291)

Investigating wider and wider formulations and reducing them to the above basic formulation would change for Post this "hypothesis not so much to a definition or to an axiom but to a *natural law*".[30]

It is methodologically remarkable that Turing proceeded in *exactly* the opposite way when trying to support the claim that all computable numbers are machine computable or, in our way of speaking, that all effectively calculable functions are Turing computable. He did not try to extend a narrow notion reducibly and obtain in this way additional quasi-empirical support; rather, he attempted to analyze the intended broad concept and reduce it to the narrow one — *once and for all*. I would like to emphasize this, as it is claimed over and over that Post provided in his 1936 paper "much the same analysis as Turing". As a matter of fact, Post hardly offers an analysis of effective calculations or combinatory processes in this paper; it may be that Post took the context of his own work, published only much later, too much for granted.[31] There is a second respect in which Post's logical work differs almost tragically from Gödel's and Turing's, and Post recognized that painfully in the letters he wrote to Gödel in 1938 and 1939: these logicians obtained decisive mathematical results that had been within reach of Post's own investigations.[32]

By examining Turing's analysis and reduction we will find the key to answering the question I raised on the difference between Church's and Turing's proposals. Very briefly put it is this: Turing deepened Church's step-by-step argument by focusing on the mechanical operations underlying the elementary steps and by formulating well-motivated constraints that guarantee their recursiveness. Before presenting in the next subsection Turing's considerations systematically, with some simplification and added structure, I discuss briefly Turing's fundamental

---

[30] [*L.c.*, 291]

[31] The earlier remark on Post's analysis is from [Kleene, 1988, 34]. In [Gandy, 1988, 98], one finds this pertinent and correct observation on Post's 1936 paper: "Post does not analyze nor justify his formulation, nor does he indicate any chain of ideas leading to it." However, that judgment is only locally correct, when focusing on this very paper. To clarify some of the interpretative difficulties and, most of all, to see the proper context of Post's work that reaches back to the early 1920s, it is crucial to consider other papers of his, in particular, the long essay [1941] that was published only in [Davis, 1965] and the part that did appear in 1943 containing the central mathematical result (canonical production systems are reducible to normal ones). In 1994 Martin Davis edited Post's *Collected Works*. Systematic presentations of Post's approach to computability theory were given by Davis [1958] and Smullyan [1961] and [1993]. Brief, but very informative introductions can be found in [Davis, 1982, 18–22], [Gandy, 1988, 92–98], and [Stillwell, 2004]. Büchi continued in most interesting ways Post's investigations; see his *Collected Works*, in particular part 7 on computability with comments by Davis.

[32] The letters are found in volume V of Gödel's *Collected Works*; a very brief description of Post's work on canonical and normal systems is given in my Introductory Note to the correspondence.

mathematical results (in Kleene's formulation) and infer the unsolvability of the *Entscheidungsproblem*.

Let ψ_M be the unary number theoretic function that is computed by machine M, and let $T(z, x, y)$ express that y is a computation of a machine with Gödelnumber z for argument x; then $\psi_M(x) = U(\mu y.T(gn(M), x, y))$; U is the result-extracting function and $gn(M)$ the Gödelnumber of M. Both T and U are easily seen to be primitive recursive, in particular, when Turing machines are presented as Post systems; see subsection 4.3. Consider the binary function $\varphi(z, x)$ defined by $U(\mu y.T(z, x, y))$; that is a partial recursive function and is computable by a machine $\mathcal{U}$ such that $\psi_{\mathcal{U}}(z, x) = \varphi(z, x)$ on their common domain of definition. $\mathcal{U}$ can compute any unary total function f that is Turing computable: $f(x) = \psi_M(x)$, when M is the machine computing f; as $\psi_M(x) = U(\mu y.T(gn(M), x, y))$, $U(\mu y.T(gn(M), x, y)) = \varphi(gn(M), x)$, and $\varphi(gn(M), x) = \psi_{\mathcal{U}}(gn(M), x)$, we have $f(x) = \psi_{\mathcal{U}}(gn(M), x)$. Thus, $\mathcal{U}$ can be considered as a "universal machine".

A modification of the diagonal argument shows that Turing machines cannot answer particular questions concerning Turing machines. The most famous question is this: Does there exist an effective procedure implemented on a Turing machine that decides for any Turing machine M and any input x, whether the computation of machine M for input x terminates or halts? This is the *Halting Problem* as formulated by Turing in 1936; it is clearly a fundamental issue concerning computations and is unsolvable. The argument is classical and begins by assuming that there is an H that solves the halting problem, i.e., for any M and x, $\psi_H(gn(M), x) = 1$ iff M halts for argument x; otherwise $\psi_H(z, x) = 0$. It is easy to construct a machine H^* from H, such that H^* halts for x iff $\psi_H(x, x) = 0$. Let h^* be $gn(H^*)$; then we have the following equivalences: H^* halts for h^* iff $\psi_H(h^*, h^*) = 0$ iff $\psi_H(gn(H^*), h^*) = 0$ iff H^* does not halt for h^*, a contradiction. Turing used the unsolvability of this problem to establish the unsolvability of related machine problems, the self-halting and the printing problem. For that purpose he implicitly used a notion of effective reducibility; a problem P, identified with a set of natural numbers, is reducible to another problem Q just in case there is a recursive function f, such that for all $x : P(x)$ if and only if $Q(f(x))$. Thus, if we want to see whether x is in P we compute $f(x)$ and test its membership in Q. In order to obtain his negative answer to the decision problem Turing reduced in a most elegant way the halting problem to the decision problem. Thus, if the latter problem were solvable, the former problem would be.

The self-halting problem K is the simplest in an infinite sequence of increasingly complex and clearly undecidable problems, the so-called *jumps*. Notice that for a machine M with code e the set K can be defined arithmetically with Kleene's T-predicate by $(\exists y)T(e, e, y)$. K is indeed *complete* for sets A that are definable by formulas obtained from recursive ones by prefixing one existential quantifier; i.e., any such A is reducible to K. These A can be given a different and very intuitive characterization: A is either the empty set or the range of a recursive function. Under this characterization the A's are naturally called "recursively enumerable", or simply r.e.. It is not difficult to show that the recursive sets are exactly those that are r.e. and have an r.e. complement. Post's way of generating these sets by production systems thus opened a distinctive approach to recursion theory.[33]

Now that we have developed a small fraction of relevant computability theory, we return to the fundamental issue, namely, why was Turing's notion of computability exactly right to obtain a convincing negative solution of the decision problem and also for achieving a precise characterization of "formal systems"? That it was exactly right, well, that still has to be argued for. The examination of mathematical results and the cool shape of a definition certainly don't provide the reason. Let us look back at Turing's paper; it opens (p. 116) with a brief description of what is ostensibly its subject, namely, "computable numbers" or "the real numbers whose expressions as a decimal are calculable by finite means". Turing is quick to point out that the fundamental problem of explicating "calculable by finite means" is the same when considering calculable functions of an integral variable, calculable predicates, and so forth. So it is sufficient to address the question, what does it mean for a real number to be calculable by finite means? Turing admits:

[33]Coming back to complex sets, one obtains the *jump hierarchy* by relativizing the concept of computation to sets of natural numbers whose membership relations are revealed by "oracles". The jump K' of K, for example, is defined as the self-halting problem, when an oracle for K is available. This hierarchy can be associated to definability questions in the language of arithmetic: all jumps are definable by some arithmetical formula, and all arithmetically definable sets are reducible to some jump. A good survey of more current work can be found in [Griffor, 1999].

> This requires rather more explicit definition. No real attempt will be made to justify the definitions given until we reach §9. For the present I shall only say that the justification lies in the fact that the human memory is necessarily limited. (p. 117)

In §9 Turing claims that the operations of his machines "include all those which are used in the computation of a number". He tries to establish the claim by answering the real question at issue, "*What are the possible processes which can be carried out in computing a number?*" The question is implicitly restricted to processes that can be carried out by a human computer. Given the systematic context that reaches back to Leibniz's "Calculemus!" this is exactly the pertinent issue to raise: the general problematic *requires* an analysis of the mechanical steps a human computer can take; after all, a positive solution to the decision problem would be provided by a procedure that in principle can be carried out by us.

Gandy made a useful suggestion, namely, calling a human carrying out a computation a "computor" and referring by "computer" to some computing machine or other. In Turing's paper, "computer" is always used for a human computing agent who proceeds mechanically; his machines, our Turing machines, consistently are just machines. The Oxford English Dictionary gives this meaning of "mechanical" when applied to a person as "resembling (inanimate) machines or their operations; acting or performed without the exercise of thought or volition;...". When I want to stress strongly the machine-like behavior of a computor, I will even speak of a *mechanical computor*. The processes such a computor can carry out are being analyzed, and that is exactly Turing's specific and extraordinary approach: the computing agent is brought into the analysis. The question is thus no longer, "Which number theoretic functions can be calculated?" but rather, "Which number theoretic functions can be calculated by a mechanical computor?" Let's address that question with Turing and see, how his analysis proceeds. Gandy emphasizes in his [1988, 83–84], absolutely correctly as we will see, that "Turing's analysis makes no reference whatsoever to calculating machines. Turing machines appear as a result, as a codification, of his analysis of calculations by humans".

4.2 Mechanical computors

Turing imagines a computor writing symbols on paper that is divided into squares "like a child's arithmetic book". Since the two-dimensional character of this computing space is taken not to be an "essential of computation" (p. 135), Turing takes a one-dimensional tape divided into squares as the basic computing space. What determines the steps of the computor? And what elementary operations can he carry out? Before addressing these questions, let me formulate one crucial and normative consideration. Turing explicitly strives to isolate operations of the computor (p. 136) that are "so elementary that it is not easy to imagine them further divided". Thus it is crucial that symbolic configurations relevant to fixing the circumstances for the computor's actions can be recognized *immediately* or *at a glance*.

Because of Turing's first reductive step to a one-dimensional tape, we have to be concerned with either individual symbols or sequences of symbols. In the first case, only finitely many distinct symbols should be written on a square; Turing argues (p. 135) for this restriction by remarking, "If we were to allow an infinity of symbols, then there would be symbols differing to an arbitrarily small extent", and the computor could not distinguish at a glance between symbols that are "sufficiently" close. In the second and related case consider, for example, Arabic numerals like 178 or 99999999 as one symbol; then it is not possible for the computor to determine at one glance whether or not 9889995496789998769 is identical with 98899954967899998769. This restriction to finitely many observed symbols or symbol sequences will be the central part of condition (**B.1**) below and also constrains via condition (**L.1**) the operations a computor can carry out.

The behavior of a computor is determined uniquely at any moment by two factors, namely, the symbols or symbol sequences he observes, and his "state of mind" or "internal state"; what is uniquely determined is the action to be performed and the next state of mind to be taken.[34] This uniqueness requirement may be called *determinacy condition* (**D**) and guarantees that computations are deterministic. Internal states are introduced so that the computor's behavior can depend on earlier observations, i.e., reflect his experience.[35] A computor thus satisfies two *boundedness conditions*:

(**B.1**) There is a fixed finite bound on the number of symbol sequences a computor can immediately recognize;

(**B.2**) There is a fixed finite bound on the number of states of mind that need to be taken into account.

For a computor there are thus only boundedly many different relevant combinations of symbol sequences and internal states. Since the computor's behavior, according to (**D**), is uniquely determined by such combinations and associated operations, the computor can carry out at most finitely many different operations, and his behavior is fixed by a finite list of commands. The operations of a computor are restricted by *locality conditions*:

(**L.1**) Only elements of observed symbol sequences can be changed;

(**L.2**) The distribution of observed squares can be changed, but each of the new observed squares must be within a bounded distance L of a previously observed square.

Turing emphasizes that "the new observed squares must be immediately recognizable by the computor" and that means the distributions of the observed

[34]Turing argues in a similar way for bounding the number of states of mind, alleging confusion, if the states of mind were too close.

[35]Turing relates state of mind to memory in §1 for his machines: "By altering its m-configuration the machine can effectively remember some of the symbols which it has 'seen' (scanned) previously." Kleene emphasizes this point in [1988, 22]: "A person computing is not constrained to working from just what he sees on the square he is momentarily observing. He can remember information he previously read from other squares. This memory consists in a state of mind, his mind being in a different state at a given moment of time depending on what he remembers from before."

squares arising from changes according to (**L.2**) must be among the finitely many ones of (**B.1**). Clearly, the same must hold for the symbol sequences resulting from changes according to (**L.1**). Since some of the operations involve a change of state of mind, Turing concludes:

> The most general single operation must therefore be taken to be one of the following: (A) A possible change (a) of symbol [as in (L.1)] together with a possible change of state of mind. (B) A possible change (b) of observed squares [as in (L.2)] together with a possible change of state of mind. (p. 137)

With this restrictive analysis of the computor's steps it is rather straightforward to conclude that a Turing machine can carry out his computations. Indeed, Turing first considers machines that operate on strings ("string machines") and mimic directly the work of the computor; then he asserts referring to ordinary Turing machines ("letter machines"):

> The machines just described [string machines] do not differ very essentially from computing machines as defined in § 2 [letter machines], and corresponding to any machine of this type a computing machine can be constructed to compute the same sequence, that is to say the sequence computed by the computer. (p. 138)

It should be clear that the string machines, just as Gandy asserted, "appear as a result, as a codification, of his analysis of calculations by humans". Thus we seem to have, shifting back to the calculation of values of number-theoretic functions, an argument for the claim: Any number-theoretic function F calculable by a computor, who satisfies the conditions (**D**) and (**B.1**)–(**L.2**), is computable by a Turing machine.[36] Indeed, both Gandy in his [1988] and I in my [1994] state that Turing established a theorem by the above argument. I don't think anymore, as the reader will notice, that that is correct in general; it is correct, however, if one considers the calculations as being carried out on strings of symbols from the very beginning.

Because of this last remark and an additional observation, Turing's analysis can be connected in a straightforward way with Church's considerations discussed in section 3.3. The additional observation concerns the determinacy condition (**D**): it is not needed to guarantee the Turing computability of F in the above claim. More precisely, (**D**) was used in conjunction with (**B.1**) and (**B.2**) to argue that computors can carry out only finitely many operations; this claim follows also from conditions (**B.1**)–(**L.2**) *without* appealing to (**D**). Thus, the behavior of

[36] A similar analysis is presented in [Wang, 1974, 90–95]. However, Wang does not bring out at all the absolutely crucial point of grounding the boundedness and locality conditions in the limitations of the computing subject; instead he appeals to an abstract *principle of finiteness*. Post's remarks on "finite methods" on pp. 426–8 in [Davis, 1965] are also grappling with these issues.

computors can still be fixed by a finite list of commands (though it may exhibit non-determinism) and can be mimicked by a Turing machine. Consider now an effectively calculable function F and a non-deterministic computor who calculates, in Church's sense, the value of F in a logic $\mathbf{L}$. Using the (additional) observation and the fact that Turing computable functions are recursive, F is recursive.[37] *This* argument for F's recursiveness does no longer appeal to Church's Thesis, not even to the restricted central thesis; rather, such an appeal is replaced by the assumption that the calculation in the logic is done by a computor satisfying the conditions (**B.1**)–(**L.2**).

Both Church and Gödel state they were convinced by Turing's work that effective calculability should be identified with Turing computability and thus is also co-extensional with recursiveness and λ-definability. Church expressed his views in the 1937 review of Turing's paper from which I quoted in the introduction; on account of Turing's work the identification is considered as "immediately evident". We'll look at that review once more in subsection 4.4 when turning attention to machine computability, as Church emphasizes the machine character of Turing's model. As to Gödel I have not been able to find in his published papers any reference to Turing's paper before his [1946] except in the purely mathematical footnote 44 of [Gödel, 1944]; that paper was discussed in section 3.4 and does not give a distinguished role to Turing's analysis. Rather, the "great importance of the concept of general recursiveness" is pointed to and "Turing computability" is added disjunctively, indeed just parenthetically. As we saw, Gödel judged that the importance of the concept is "largely due" to its absoluteness.

There is some relevant discussion of Turing's work in unpublished material that is now available in the *Collected Works*, namely, in Gödel's [193?, 164—175] of *CW III*), the Gibbs lecture of 1951 (pp. 304–5 and p. 309 of *CW III*), and in the letter of 2 February 1957 that was addressed, but not sent, to Ernest Nagel (pp. 145–6 of *CW V*). The first written and public articulation of Gödel's views can be found in the 1963 Addendum to his [1931] (for its publication in [van Heijenoort, 1967]) and in the 1964 Postscriptum to the Princeton Lectures (for their publication in [Davis, 1965]). In the latter, more extended note, Gödel is perfectly clear about the structure of Turing's argument. "Turing's work", he writes, "*gives an analysis* [my emphasis] of the concept 'mechanical procedure' (*alias* 'algorithm' or 'computation procedure' or 'finite combinatorial procedure'). This concept is *shown* [my emphasis] to be equivalent with that of a 'Turing machine'." In a footnote attached to this observation he called "previous equivalent definitions of computability", referring to λ-definability and recursiveness, "much less suitable for our purpose". What is not elucidated by any remark of Gödel, as far as I know, is the *result* of Turing's analysis, i.e., the explicit formulation of restrictive conditions. There is consequently no discussion of the *reasons* for the correctness of these conditions or, for that matter, of the analysis; there is also no indication

[37] The proof is given via considerations underlying Kleene's normal form theorem. That is done in the most straightforward way if, as discussed in the next subsection, Turing machines are described as Post systems.

of a proof establishing the equivalence between the analyzed (and presumably rigorous) notion of mechanical procedure and the concept of a Turing machine. (Gödel's views are traced with many more details in my [2006].)

A comparison of Gödel's concise description with Turing's actual argument raises a number of important issues, in particular one central question I earlier put aside: Isn't the starting-point of Turing's argument too vague and open, unless we take for granted that the symbolic configurations are of a certain kind, namely, symbol strings in our case? But even if that is taken for granted and Turing's argument is viewed as perfectly convincing, there remains a methodological problem. According to Gödel the argument consists of an analysis followed by a proof; how do we carve up matters, i.e., where does the analysis stop and the proof begin? Does the analysis stop only, when a string machine has fully captured the computor's actions, and the proof is just the proof establishing the reduction of computations by string machines to those by letter machines? Or does the analysis just lead to restrictive conditions for mechanical computors and the proof establishes the rest? To get a clearer view about these matters, I will simplify the argument and examine more closely the justificatory steps.

4.3 Turing's central thesis

The first section of this essay had the explicit purpose of exposing the broad context for the investigations of Herbrand, Gödel, Church, Kleene, Post, and Turing. There is no doubt that an analysis of human effective procedures on finite (symbolic) configurations was called for, and that the intended epistemological restrictions were cast in "mechanical" terms; *vide* as particularly striking examples the remarks of Frege and Gödel quoted in section 2.1. Thus, Turing's explication of *effective calculability* as *calculability by a mechanical computor* should be accepted. What are the general restrictions on calculation processes, and how are such constraints related to the nature of mechanical computors?

The justificatory steps in Turing's argument contain crucial appeals to boundedness and locality conditions. Turing claims that their ultimate justification lies in the necessary limitation of human memory. According to Gandy, Turing arrives at the restrictions "by considering the limitations of our sensory and mental apparatus". However, in Turing's argument only limitations of our sensory apparatus are involved, unless "state of mind" is given an irreducibly mental touch. That is technically unnecessary as Post's equivalent formulation makes clear. It is systematically also not central for Turing, as he describes in section 9 (III) of his paper, p. 139, a modified computor. There he avoids introducing "state of mind" by considering instead "a more physical and definite counterpart of it". (Indeed, if we take into account the quest for insuring "radical intersubjectivity" then internal, mental states should be externalized in any event.) Thus, Turing's analysis can be taken to appeal only to sensory limitations of the type I discussed at the beginning of section 4.2.[38] Such limitations *are* obviously operative when

[38]As Turing sees memory limitations as ultimately justifying the restrictive conditions, but

we work as purely mechanical computors.

Turing himself views his argument for the reduction of effectively calculable functions to functions computable by his machines as basically "a direct appeal to intuition". Indeed, he claims, p. 135, more strongly, "All arguments which can be given [for this reduction] are bound to be, fundamentally, appeals to intuition, and for that reason rather unsatisfactory mathematically." If we look at his paper [Turing, 1939], the claim that such arguments are "unsatisfactory mathematically" becomes at first rather puzzling, since he observes there that intuition is inextricable from mathematical reasoning. Turing's concept of intuition is much more general than that ordinarily used in the philosophy of mathematics. It is introduced in the 1939 paper explicitly to address the issues raised by Gödel's first incompleteness theorem; that is done in the context of work on ordinal logics or, what was later called, progressions of theories. The discussion is found in section 11:

> Mathematical reasoning may be regarded rather schematically as the exercise of a combination of two faculties, which we may call *intuition* and *ingenuity*. The activity of the intuition consists in making spontaneous judgements which are not the result of conscious trains of reasoning. These judgements are often but by no means invariably correct (leaving aside the question of what is meant by "correct"). ...The exercise of ingenuity in mathematics consists in aiding the intuition through suitable arrangements of propositions, and perhaps geometrical figures or drawings. It is intended that when these are really well arranged the validity of the intuitive steps which are required cannot seriously be doubted. (pp. 208–210)

Are the propositions in Turing's argument arranged with sufficient ingenuity so that "the validity of the intuitive steps which are required cannot seriously be doubted"? Or, does their arrangement allow us at least to point to central restrictive conditions with clear, adjudicable content?

To advance the further discussion, I simplify the formulation of the restrictive conditions that can be extracted from Turing's discussion by first eliminating internal states by "more physical counterparts" as Turing himself proposed. Then I turn machine operations into purely symbolic ones by presenting suitable Post productions as Turing himself did for obtaining new mathematical results in his [1950a], but also for a wonderful informal exposition of solvable and unsolvable

none of the conditions seems to be directly motivated by such a limitation, we should ask, how we can understand his claim. I suggest the following: If our memory were not subject to limitations of the same character as our sensory apparatus, we could scan (with the limited sensory apparatus) a symbolic configuration that is not immediately recognizable, read in sufficiently small parts so that their representations could be assembled in a unique way to a representation of the given symbolic configuration, and finally carry out (generalized) operations on that representation in memory. Is one driven to accept Turing's assertion as to the limitation of memory? I suppose so, if one thinks that information concerning symbolic structures is physically encoded and that there is a bound on the number of available codes.

problems in [1954]. Turing extended in the former paper Post's (and Markov's) result concerning the unsolvability of the word-problem for semi-groups to semi-groups with cancellation; on the way to the unsolvability of this problem, [Post, 1947] had used a most elegant way of describing Turing machines as production systems. The configurations of a Turing machine are given by *instantaneous descriptions* of the form $\alpha q_l s_k \beta$, where α and β are possibly empty strings of symbols in the machine's alphabet; more precisely, an *id* contains exactly one state symbol and to its right there must be at least one symbol. Such *id*s express that the current tape content is $\alpha s_k \beta$, the machine is in state q_l and scans (a square with symbol) s_k. Quadruples $q_i s_k c_l q_m$ of the program are represented by rules; for example, if the operation c_l is *print* 0, the corresponding rule is:

$$\alpha q_i s_k \beta => \alpha q_m 0 \beta.$$

Such formulations can be given, obviously, for all the different operations. One just has to append s_0 to $\alpha(\beta)$ in case c_l is the operation *move to the left* (*right*) and $\alpha(\beta)$ is the empty string; that reflects the expansion of the only potentially infinite tape by a blank square.

This formulation can be generalized so that machines operate directly on finite strings of symbols; operations can be indicated as follows:

$$\alpha \gamma q_l \delta \beta => \alpha \gamma^* q_m \delta^* \beta.$$

If in internal state q_l a *string machine* recognizes the string $\gamma\delta$ (i.e., takes in the sequence at one glance), it replaces that string by $\gamma^*\delta^*$ and changes its internal state to q_m. The rule systems describing string machines are semi-Thue systems and, as the latter, not deterministic, if their programs are just sequences of production rules. The usual non-determinism certainly can be excluded by requiring that, if the antecedents of two rules coincide, so must the consequents. But that requirement does not remove every possibility of two rules being applicable simultaneously: consider a machine whose program includes in addition to the above rule also the rule

$$\alpha \gamma^\sharp q_l \delta^\sharp \beta => \alpha \gamma^\perp q_n \delta^\perp \beta,$$

where $\delta^\sharp$ is an initial segment of δ, and $\gamma^\sharp$ is an end segment of γ; under these circumstances both rules would be applicable to $\gamma q_l \delta$. This non-determinism can be excluded in a variety of ways, e.g., by always using the applicable rule with the largest context. In sum, the Post representation joins the physical counterparts of internal states to the ordinary symbolic configurations and forms instantaneous descriptions, abbreviated as *id*. Any *id* contains exactly one such physical counterpart, and the immediately recognizable sub-configuration of an *id* must contain it. As the state symbol is part of the observed configuration, its internal shifting can be used to indicate a shift of the observed configuration. Given this compact description, the restrictive conditions are as follows:

(**B**) (Boundedness) There is a fixed finite bound on the number of symbol sequences (containing a state symbol) a computor can immediately recognize.

(**L**) (Locality) A computor can change only an *id*'s immediately recognizable sub-configuration.

These restrictions on computations are specifically and directly formulated for Post productions. Turing tried to give, as we saw, a more general argument starting with a broader class of symbolic configurations. Here is the starting-point of his considerations together with a dimension-lowering step to symbol sequences:

> Computing is normally done by writing certain symbols on paper. We may suppose this paper is divided into squares like a child's arithmetic book. In elementary arithmetic, the two-dimensional character of the paper is sometimes used. But such a use is always avoidable, and I think that it will be agreed that the two-dimensional character of paper is no essential of computation. I assume then that the computation is carried out on one-dimensional paper, *i.e.* on a tape divided into squares. (p. 135)

This last assumption, ... *the computation is carried out on one-dimensional paper* ..., is based on an appeal to intuition in Turing's sense and makes the general argument unconvincing as a rigorous proof. Turing's assertion that effective calculability can be identified with machine computability should thus be viewed as the result of asserting a central thesis and constructing a two-part argument: the central thesis asserts that the computor's calculations are carried out on symbol sequences; the first part of the argument (using the sensory limitations of the computor) yields the *claim* that every operation (and thus every calculation) can be carried out by a suitable string machine; the second part is the rigorous proof that letter machines can simulate these machines. The *claim* is trivial, as the computor's operations *are* the machine operations.

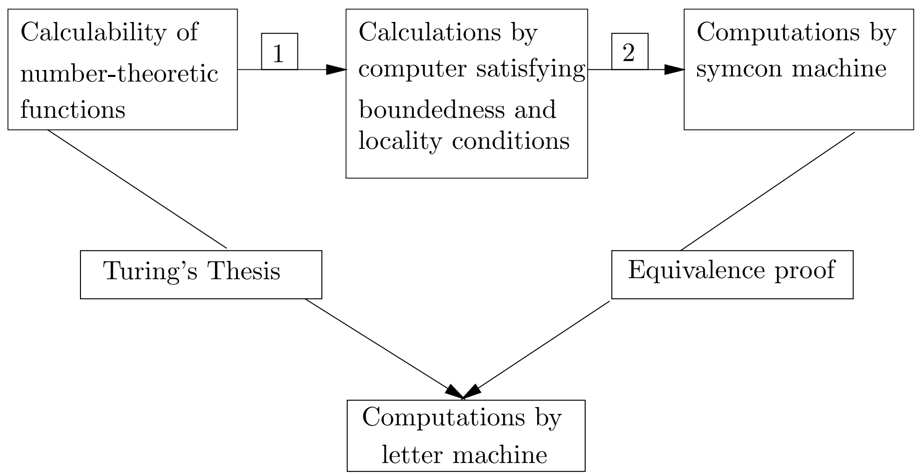
4.4 Stronger theses

The above argumentative structure leading from computor calculations to Turing machine computations is rather canonical, once the symbolic configurations are fixed as symbol sequences and given the computor's limitations. In the case of other, for example, two or three-dimensional symbolic configurations, I do not see such a canonical form of reduction, unless one assumes again that the configurations are of a very special regular or normal shape.[39] In general, an "argumentative structure" supporting a reduction will contain then a *central thesis* in a far stronger sense, namely, that the calculations of the computor can be carried out by a precisely described device operating on a particular class of symbolic configurations;

[39] This issue is also discussed in Kleene's *Introduction to Metamathematics*, pp. 376–381, in an informed and insightful defense of Turing's Thesis. However, in Kleene's way of extending configurations and operations, much stronger normalizing conditions are in place; e.g., when considering machines corresponding to our string machines the strings must be of the same length.

indeed, the devices should be viewed as generalized Post productions. These last considerations also indicate, how to carve up matters between analysis and proof; i.e., they allow us to answer the question asked at the end of subsection 4.2.

The diagram below represents these reflections graphically and relates them to the standard formulation of Turing's Thesis. Step 1 is given by conceptual analysis, whereas step 2 indicates the application of the central thesis for a particular class of symbolic configurations or *symcons*. (The symcon machines are Post systems operating, of course, on symcons.) The equivalence proof justifies an extremely simple description of computations that is most useful for mathematical investigations, from the construction of a universal machine and the formulation of the halting problem to the proof of the undecidability of the *Entscheidungsproblem*. It should be underlined that step 2, not the equivalence proof, is for Turing the crucial one that goes beyond the conceptual analysis; for me it is the problematic one that requires further reflection. I will address it in two different ways: inductively now and axiomatically in Section 5.



In order to make Turing's central thesis, quite in Post's spirit, inductively more convincing, it seems sensible to allow larger classes of symbolic configurations and more general operations on them. Turing himself intended, as we saw, to give an analysis of mechanical procedures on two-dimensional configurations already in 1936. In 1954 he considered even three-dimensional configurations and mechanical operations on them, starting out with examples of puzzles: square piece puzzles, puzzles involving the separation of rigid bodies or the transformation of knots, i.e., puzzles in two and three dimensions. He viewed Post production systems as *linear* or *substitution puzzles*. As he considered them as puzzles in "normal form", he was able to formulate a suitable version of "Turing's Thesis":

> Given any puzzle we can find a corresponding substitution puzzle which is equivalent to it in the sense that given a solution of the one we can

> easily find a solution of the other. ... A transformation can be carried out by the rules of the original puzzle if and only if it can be carried out by substitutions ....[40]

Turing admits that this formulation is "somewhat lacking in definiteness" and claims that it will remain so; he characterizes its status as lying between a theorem and a definition: "In so far as we know *a priori* what is a puzzle and what is not, the statement is a theorem. In so far as we do not know what puzzles are, the statement is a definition which tells us something about what they are." Of course, Turing continues, one could define puzzle by a phrase beginning with "a set of definite rules", or one could reduce its definition to that of computable function or systematic procedure. A definition of any of these notions would provide one for puzzles. Neither in 1936 nor in 1954 did Turing try to characterize mathematically more general configurations and elementary operations on them. I am going to describe briefly one particular attempt of doing just that by Byrnes and me in our [1996].

Our approach was influenced by Kolmogorov and Uspensky's work on algorithms and has three distinct components: the symbolic configurations are certain finite connected and labeled graphs, we call *K(olmogorov)-graphs*; K-graphs contain a unique *distinguished element* that corresponds to the scanned square of a Turing machine tape; the operations *substitute neighborhoods* of the distinguished element by appropriate other neighborhoods and are given by a finite list of generalized Post production rules. Though broadening Turing's original considerations, we remain clearly within his general analytic framework and prove that letter machines can mimic K-graph machines. Turing's central thesis expresses here that K-graph machines can do the work of computors directly. As a playful indication of how K-graph machines straightforwardly can carry out human and genuinely symbolic, indeed diagrammatic algorithms, we programmed a K-graph machine to do ordinary, two-dimensional column addition. In sum then, a much more general class of symbolic configurations and operations on them is considered, and the central thesis for K-graph machines seems even more plausible than the one for string machines.

The separation of conceptual analysis and mathematical proof is essential for recognizing that the correctness of *Turing's Thesis* (taken generically) rests on two pillars, namely, on the correctness of boundedness and locality conditions for computors and on the correctness of the pertinent central thesis. The latter asserts explicitly that calculations of a computor can be mimicked by a particular kind of machine. However satisfactory one may find this line of argument, there are two weak spots: the looseness of the restrictive conditions (What are symbolic configurations? What changes can mechanical operations effect?) and the corresponding vagueness of the central thesis. We are, no matter how we turn ourselves, in a position that is methodologically not fully satisfactory.

---

[40] [Turing, 1954, 15]

4.5 Machine computability

Before attacking the central methodological issue in Section 5 from a different angle that is however informed by our investigations so far, let us look at the case, where reflection on limitations of computing devices leads to an important general concept of parallel computation and allows us to abstract further from particular types of configurations and operations. These considerations are based on Gandy's work in his [1980] that in its broad methodological approach parallels Turing's. At issue is machine calculability. The machines Turing associates with the basic operations of a computor can be physically realized, and we can obviously raise the question, whether these devices (our desktop computers, for example) are just doing things faster than we do, or whether they are in a principled way computationally more powerful.

It is informative first to look at Church's perspective on Turing's work in his 1937 review for the *Journal of Symbolic Logic*. Church was very much on target, though there is one fundamental misunderstanding as to the relative role of computor and machine computability in Turing's argument. For Church, computability by a machine "occupying a finite space and with working parts of finite size" is analyzed by Turing; given that the Turing machine is the outcome of the analysis, one can then observe that "in particular, a human calculator, provided with pencil and paper and explicit instructions, can be regarded as a kind of Turing machine". On account of the analysis and this observation it is for Church then "immediately clear" that (Turing-) machine computability can be identified with effectiveness. This is re-emphasized in the rather critical review of Post's 1936 paper in which Church pointed to the essential finiteness requirements in Turing's analysis: "To define effectiveness as computability by an arbitrary machine, subject to restrictions of finiteness, would seem to be an adequate representation of the ordinary notion, and if this is done the need for a working hypothesis disappears." This is right, as far as emphasis on finiteness restrictions is concerned. But Turing analyzed, as we saw, a mechanical computor, and *that* provides the basis for judging the correctness of the finiteness conditions. In addition, Church is rather quick in his judgment that "certain further restrictions" can be imposed on such arbitrary machines to obtain Turing's machines; this is viewed "as a matter of convenience" and the restrictions are for Church "of such a nature as obviously to cause no loss of generality".

Church's apparent misunderstanding is rather common; see, as a later example, Mendelson's paper [1990]. It is Turing's student, Robin Gandy who analyzes machine computability in his 1980 paper *Church's thesis and principles for mechanisms* and proposes a particular mathematical description of *discrete mechanical devices* and their computations. He follows Turing's three-step-argument of analysis, formulation of restrictive principles and proof of a "reduction theorem". Gandy shows that everything calculable by a device satisfying the restrictive principles is already computable by a Turing machine. The central and novel aspect of Gandy's analysis is the fact that it incorporates parallelism and covers cellular automata directly. This is of real interest, as cellular automata do not satisfy the locality condition (**L**); after all, the configurations affected in a single computation step are potentially unbounded.

What are discrete mechanical devices "in general"? — Gandy introduces the term to make it clear that he does not deal with analogue devices, but rather with machines that are "discrete" (i.e., consist of finitely many parts) and proceed step-by-step from one state to the next. Gandy considers two fundamental physical constraints for such devices: (1) a lower bound on the size of atomic components; (2) an upper bound on the speed of signal propagation.[41] These two constraints together guarantee what the sensory limitations guarantee for computors, namely that in a given unit of time there are only a bounded number of different observable configurations (in a broad sense) and just a bounded number of possible actions on them. This justifies Gandy's contention that states of such machines "can be adequately described in finite terms", that calculations are proceeding in discrete and uniquely determined steps and, consequently, that these devices can be viewed, in a loose sense, as digital computers. If that's all, then it seems that without further ado we have established that machines in this sense are computationally not more powerful than computors, at least not in any principled way. However, if the concept of machine computability has to encompass "massive parallelism" then we are not done yet, and we have to incorporate that suitably into the mathematical description. And that can be done. Indeed, Gandy provided for the first time a conceptual analysis and a general description of parallel algorithms.

Gandy's characterization is given in terms of discrete dynamical systems $\langle \mathbf{S}, \mathbf{F} \rangle$, where $\mathbf{S}$ is the set of states and $\mathbf{F}$ governs the system's evolution. These dynamical systems have to satisfy four restrictive principles. The first principle pertains to the *form of description* and states that any machine $\mathbf{M}$ can be presented by such a pair $\langle \mathbf{S}, \mathbf{F} \rangle$, and that $\mathbf{M}$'s computation, starting in an initial state $\mathbf{x}$, is given by the sequence $\mathbf{x}$, $\mathbf{F}(\mathbf{x})$, $\mathbf{F}(\mathbf{F}(\mathbf{x}))$, ... Gandy formulates three groups of substantive principles, the first of which, *The Principle of Limitation of Hierarchy*, requires that the set theoretic rank of the states is bounded, i.e., the structural class $\mathbf{S}$ is contained in a fixed initial segment of the hierarchy of hereditarily finite and non-empty sets HF. Gandy argues (on p. 131) that it is natural or convenient to think of a machine in hierarchical terms, and that "for a given machine the maximum height of its hierarchical structure must be bounded". The second of the substantive principles, *The Principle of Unique Reassembly*, claims that any state can be "assembled" from "parts" of bounded size; its proper formulation requires care and a lengthy sequence of definitions. The informal idea, though, is wonderfully straightforward: any state of a concrete machine must be built up from (finitely many different types of) off-the-shelf components. Clearly, the components have a bound on their complexity. Both of these principles are

[41] Cf. [Gandy, 1980, 126, but also 135–6]. For a more detailed argument see [Mundici and Sieg, section 3], where physical limitations for computing devices are discussed. In particular, there is an exploration of how space-time of computations are constrained, and how such constraints prevent us from having "arbitrarily" complex physical operations.

concerned with the states in **S**; the remaining third and central principle, *The Principle of Local Causality*, puts conditions on (the local determination of) the structural operation **F**. It is formulated by Gandy in this preliminary way: "The next state, **Fx**, of a machine can be reassembled from its restrictions to overlapping 'regions' **s** and these restrictions are locally caused." It requires that the parts from which **F**(**x**) can be assembled depend only on bounded parts of **x**.

Gandy's Central Thesis is naturally the claim that any discrete mechanical device can be described as a dynamical system satisfying the above substantive principles. As to the set-up John Shepherdson remarked in his [1988, 586]: "Although Gandy's principles were obtained by a very natural analysis of Turing's argument they turned out to be rather complicated, involving many subsidiary definitions in their statement. In following Gandy's argument, however, one is led to the conclusion that that is in the nature of the situation." Nevertheless, in [Sieg and Byrnes, 1999] a greatly simplified presentation is achieved by choosing definitions appropriately, following closely the central informal ideas and using one key suggestion made by Gandy in the Appendix to his paper. This simplification does not change at all the form of presentation. However, of the four principles used by Gandy only a restricted version of the principle of local causality is explicitly retained. It is formulated in two separate parts, namely, as the principle of *Local Causation* and that of *Unique Assembly*. The separation reflects the distinction between the local determination of regions of the next state and their assembly into the next state.

Is it then correct to think that Turing's and Gandy's analyses lead to results that are in line with Gödel's general methodological expectations expressed to Church in 1934? Church reported that expectation to Kleene a year later and formulated it as follows:

> His [i.e. Gödel's] only idea at the time was that it might be possible, in terms of effective calculability as an undefined notion, to state a set of axioms which would embody the generally accepted properties of this notion, and to do something on that basis.[42]

Let's turn to that issue next.

5 AXIOMS FOR COMPUTABILITY.

The analysis offered by Turing in 1936 and re-described in 1954 was contiguous with the work of Gödel, Church, Kleene, Hilbert and Bernays, and others, but at the same time it was radically different and strikingly novel. They had explicated the calculability of number-theoretic functions in terms of their evaluation in calculi using only elementary and arithmetically meaningful steps; that put a stumbling-block into the path of a deeper analysis. Turing, in contrast, analyzed the basic processes that are carried out by computors and underlie the elementary

[42]Church in the letter to Kleene of November 29, 1935, quoted in [Davis, 1982, 9].

calculation steps. The restricted machine model that resulted from his analysis almost hides the fact that Turing deals with general symbolic processes.

Turing's perspective on such general processes made it possible to restrict computations by *boundedness* and *locality* conditions. These conditions are obviously violated and don't even make sense when the values of number theoretic functions are determined by arithmetically meaningful steps. For example, in Gödel's equational calculus the replacement operations involve quite naturally arbitrarily complex terms. However, for steps of general symbolic processes the conditions are convincingly motivated by the sensory limitations of the computing agent and the normative demand of immediate recognizability of configurations; the basic steps, after all, must not be in need of further analysis. Following Turing's broad approach Gandy investigated in [1980] the *computations of machines*. Machines can in particular carry out parallel computations, and physical limitations motivate restrictive conditions for them. In spite of the generality of his notion, Gandy was able to show that any machine computable function is also Turing computable.

These analyses are taken now as a basis for further reflections along Gödelian lines. In a conversation with Church that took place in early 1934, Gödel found Church's proposal to identify effective calculability with λ-definability "thoroughly unsatisfactory", but he did make a counterproposal. He suggested "to state a set of axioms which embody the generally accepted properties of this notion (i.e., effective calculability), and to do something on that basis". Gödel did not articulate what the generally accepted properties of effective calculability might be or what might be done on the basis of an appropriate set of axioms. Sharpening Gandy's work I will give an abstract characterization of "Turing Computors" and "Gandy Machines" as discrete dynamical systems whose evolutions satisfy some well-motivated and general axiomatic conditions. Those conditions express constraints, which have to be satisfied by computing processes of these particular devices. Thus, I am taking the axiomatic method as a tool to resolve the methodological problems surrounding Church's thesis for computors and machines. The mathematical formulations that follow in section 5.1 are given in greater generality than needed for Turing computors, so that they cover also the discussion of Gandy machines. (They are also quite different from the formulation in [Gandy, 1980] or in [Sieg and Byrnes, 1999a].)

5.1 Discrete dynamical systems

At issue is, how we can express those "well-motivated conditions" in a sharp way, as I clearly have not given an answer to the questions: What are symbolic configurations? What changes can mechanical operations effect? Nevertheless, some aspects can be coherently formulated for computors: (i) they operate deterministically on finite configurations; (ii) they recognize in each configuration exactly one pattern (from a bounded number of different kinds of such); (iii) they operate locally on the recognized patterns; (iv) they assemble the next configuration from the original one and the result of the local operation. Discrete dynamical systems provide an elegant framework for capturing these general ideas precisely. We consider pairs $\langle \mathbf{D}, \mathbf{F} \rangle$ where $\mathbf{D}$ is a class of states (ids or syntactic configurations) and $\mathbf{F}$ an operation from $\mathbf{D}$ to $\mathbf{D}$ that transforms a given state into the next one. States are finite objects and are represented by non-empty hereditarily finite sets over an infinite set $\mathbf{U}$ of atoms. Such sets reflect states of computing devices just as other mathematical structures represent states of nature, but this reflection is done somewhat indirectly, as only the $\in$-relation is available.

In order to obtain a more adequate mathematical framework free of ties to particular representations, we consider *structural classes* $\mathbf{S}$, i.e., classes of states that are closed under $\in$-isomorphisms. After all, any $\in$-isomorphic set can replace a given one in this reflective, representational role. That raises immediately the question, what invariance properties the state transforming operations $\mathbf{F}$ should have or how the $\mathbf{F}$-images of $\in$-isomorphic states are related. Recall that any $\in$-isomorphism π between states is a unique extension of some permutation on atoms, and let $\pi(\mathbf{x})$ or $\mathbf{x}^\pi$ stand for the result of applying π to the state $\mathbf{x}$. The lawlike connections between states are given by *structural operations* $\mathbf{G}$ from $\mathbf{S}$ to $\mathbf{S}$. The requirement on $\mathbf{G}$ will fix the dependence of values on just structural features of a state, not the nature of its atoms: for all permutations π on $\mathbf{U}$ and all $\mathbf{x} \in \mathbf{S}$, $\mathbf{G}(\pi(\mathbf{x}))$ is $\in$-isomorphic to $\pi(\mathbf{G}(\mathbf{x}))$, and the isomorphism has the additional property that it is the identity on the atoms occurring in the support of $\pi(\mathbf{x})$. $\mathbf{G}(\pi(\mathbf{x}))$ and $\pi(\mathbf{G}(\mathbf{x}))$ are said to be *$\in$-isomorphic over* $\pi(\mathbf{x})$, and we write $\mathbf{G}(\pi(\mathbf{x})) \cong_{\pi(\mathbf{x})} \pi(\mathbf{G}(\mathbf{x}))$. Note that we do not require the literal identity of $\mathbf{G}(\pi(\mathbf{x}))$ and $\pi(\mathbf{G}(\mathbf{x}))$; that would be too restrictive, as the state may be expanded by new atoms and it should not matter which new atoms are chosen. On the other hand, the requirement $\mathbf{G}(\pi(\mathbf{x}))$ is $\in$-isomorphic to $\pi(\mathbf{G}(\mathbf{x}))$ would be too loose, as we want to guarantee the physical persistence of atomic components. Here is the appropriate diagram:

$$\begin{tikzcd}
\mathbf{x} \arrow[r] \arrow[d] & \mathbf{G}(\mathbf{x}) \arrow[dr] & \\
\pi(\mathbf{x}) \arrow[r] & \mathbf{G}(\pi(\mathbf{x})) \cong_{\pi(\mathbf{x})} & \pi(\mathbf{G}(\mathbf{x}))
\end{tikzcd}$$

This mathematical framework addresses just point (i) in the above list of central aspects of mechanical computors. Now we turn to *patterns* and *local* operations. If $\mathbf{x}$ is a given state, regions of the next state are determined *locally* from particular *parts for* $\mathbf{x}$ on which the computor can operate.[43] *Boundedness* requires that there is only a bounded number of different kinds of parts, i.e., each part lies in one of

[43] A part $\mathbf{y}$ for $\mathbf{x}$ used to be in my earlier presentations a connected subtree $\mathbf{y}$ of the $\in$-tree for $\mathbf{x}$, briefly $\mathbf{y}<^*\mathbf{x}$, if $\mathbf{y}\neq\mathbf{x}$ and $\mathbf{y}$ has the same root as $\mathbf{x}$ and its leaves are also leaves of $\mathbf{x}$. More precisely, $\mathbf{y}\neq\mathbf{x}$ and $\mathbf{y}$ is a non-empty subset of $\{\mathbf{v} \mid (\exists\mathbf{z})(\mathbf{v}<^*\mathbf{z} \ \& \ \mathbf{z}\in\mathbf{x})\} \cup \{\mathbf{r} \mid \mathbf{r}\in\mathbf{x}\}$. Now it is just a subset, but I will continue to use the term "part" to emphasize that we are taking the whole $\in$-structure into account.

a finite number of isomorphism types or, using Gandy's terminology, *stereotypes*. So let $\mathbf{T}$ be a fixed finite class of stereotypes. A part for $\mathbf{x}$ that is a member of a stereotype of $\mathbf{T}$ is called, naturally enough, a $\mathbf{T}$-*part for* $\mathbf{x}$. A $\mathbf{T}$-part $\mathbf{y}$ for $\mathbf{x}$ is a *causal neighborhood for* $\mathbf{x}$ given by $\mathbf{T}$, briefly $\mathbf{y}\in\mathrm{Cn}(\mathbf{x})$, if there is no $\mathbf{T}$-part $\mathbf{y}^*$ for $\mathbf{x}$ such that $\mathbf{y}$ is $\in$-embeddable into $\mathbf{y}^*$. The causal neighborhoods for $\mathbf{x}$ will also be called *patterns in* $\mathbf{x}$. Finally, the local change is effected by a structural operation $\mathbf{G}$ that works on unique causal neighborhoods. Having also given points (ii) and (iii) a proper mathematical explication, the assembly of the next state has to be determined.

The values of $\mathbf{G}$ are in general not exactly what we need in order to assemble the next state, because the configurations may have to be expanded and that involves the addition and coordination of new atoms. To address that issue we introduce *determined regions* $\mathrm{Dr}(\mathbf{z},\mathbf{x})$ of a state $\mathbf{z}$; they are $\in$-isomorphic to $\mathbf{G}(\mathbf{y})$ for some causal neighborhood $\mathbf{y}$ for $\mathbf{x}$ and must satisfy a technical condition on the "newness" of atoms. More precisely, $\mathbf{v}\in\mathrm{Dr}(\mathbf{z},\mathbf{x})$ if and only if $\mathbf{v}<^*\mathbf{z}$ and there is a $\mathbf{y}\in\mathrm{Cn}(\mathbf{x})$, such that $\mathbf{G}(\mathbf{y})\cong_{\mathbf{y}}\mathbf{v}$ and $\mathrm{Sup}(\mathbf{v})\cap\mathrm{Sup}(\mathbf{x})\subseteq\mathrm{Sup}(\mathbf{y})$. The last condition for Dr guarantees that new atoms in $\mathbf{G}(\mathbf{y})$ correspond to new atoms in $\mathbf{v}$, and that the new atoms in $\mathbf{v}$ are new for $\mathbf{x}$. If one requires $\mathbf{G}$ to satisfy similarly $\mathrm{Sup}(\mathbf{G}(\mathbf{y}))\cap\mathrm{Sup}(\mathbf{x})\subseteq\mathrm{Sup}(\mathbf{y})$, then the condition $\mathbf{G}(\mathbf{y})\cong_{\mathbf{y}}\mathbf{v}$ can be strengthened to $\mathbf{G}(\mathbf{y})\cong_{\mathbf{x}}\mathbf{v}$. The new atoms are thus always taken from $\mathbf{U}\backslash\mathrm{Sup}(\mathbf{x})$.[44] One final definition: for given states $\mathbf{z}$ and $\mathbf{x}$ let $\mathrm{A}(\mathbf{z},\mathbf{x})$ stand for $\mathrm{Sup}(\mathbf{z})\backslash\mathrm{Sup}(\mathbf{x})$. Note that the number of new atoms introduced by $\mathbf{G}$ is bounded, i.e., $|\mathrm{A}(\mathbf{G}(\mathbf{y}),\mathrm{Sup}(\mathbf{x}))|<n$ for some natural number n (any $\mathbf{x}\in\mathbf{S}$ and any causal neighborhood $\mathbf{y}$ for $\mathbf{x}$).

So, how is the next state of a Turing computor assembled? By simple set theoretic operations, namely, difference $\backslash$ and union $\cup$. Recalling the boundedness and locality conditions for computors we define that $\mathbf{M}=\langle\mathbf{S};\mathbf{T},\mathbf{G}\rangle$ is a *Turing Computor on* $\mathbf{S}$, where $\mathbf{S}$ is a structural class, $\mathbf{T}$ a finite set of stereotypes, and $\mathbf{G}$ a structural operation on $\cup\,\mathbf{T}$, if and only if, for every $\mathbf{x}\in\mathbf{S}$ there is a $\mathbf{z}\in\mathbf{S}$, such that:

(**L.0**) $(\exists!\mathbf{y})\ \mathbf{y}\in\mathrm{Cn}(\mathbf{x})$,

(**L.1**) $(\exists!\mathbf{v}\in\mathrm{Dr}(\mathbf{z},\mathbf{x}))\ \mathbf{v}\cong_{\mathbf{x}}\mathbf{G}(\mathbf{cn}(\mathbf{x}))$,

(**A.1**) $\mathbf{z}=(\mathbf{x}\backslash\mathrm{Cn}(\mathbf{x}))\cup\mathrm{Dr}(\mathbf{z},\mathbf{x})$.

L stands for Locality and **A** for Assembly. $(\exists!\mathbf{y})$ is the existential quantifier expressing uniqueness. $\mathbf{cn}(\mathbf{x})$ denotes the sole causal neighborhood of $\mathbf{x}$ guaranteed by **L.0**, i.e., every state is required by **L.0** to contain exactly one pattern. This pattern in state $\mathbf{x}$ yields a unique determined region of a possible next state $\mathbf{z}$; that is expressed by **L.1**. The state $\mathbf{z}$ is obtained according to the assembly condition **A.1**. It is determined up to $\in$-isomorphism over $\mathbf{x}$. A *computation by* $\mathbf{M}$ is a finite sequence of transition steps via $\mathbf{G}$ that is halted when the operation on a state $\mathbf{w}$ yields $\mathbf{w}$ as the next state. This result, for input $\mathbf{x}$, is denoted by $\mathbf{M}(\mathbf{x})$.

[44] This selection of atoms new for $\mathbf{x}$ has in a very weak sense a "global" aspect; as $\mathbf{G}$ is a structural operation, the precise choice of the atoms does not matter.

A function **F** is (Turing) *computable* if and only if there is a Turing computor **M** whose computation results determine — under a suitable encoding and decoding — the values of **F** for any of its arguments. After all these definitions one can use a suitable set theoretic representation of Turing machines to establish one lemma, namely, that Turing machines are Turing computors. (See section 5.4.)

In the next subsection, we will provide a characterization of computations by machines that is as general and convincing as that of human computors. Gandy laid the groundwork in his thought-provoking paper *Church's Thesis and Principles for Mechanisms* — a rich and difficult, but unnecessarily and maddeningly complex paper. The structure of Turing's argument actually guided Gandy's analysis; however, Gandy realized through conversations with J. C. Shepherdson that the analysis "must take parallel working into account". In a comprehensive survey article published ten years after Gandy's paper, Leslie Lamport and Nancy Lynch argued that the theory of sequential computing "rests on fundamental concepts of computability that are independent of any particular computational model". They emphasized that the "fundamental formal concepts underlying distributed computing", if there were any, had not yet been developed. "Nevertheless", they wrote, "one can make some informal observations that seem to be important":

> Underlying almost all models of concurrent systems is the assumption that an execution consists of a set of discrete events, each affecting only part of the system's state. Events are grouped into processes, each process being a more or less completely sequenced set of events sharing some common locality in terms of what part of the state they affect. For a collection of autonomous processes to act as a coherent system, the processes must be synchronized. (p. 1166)

Gandy's analysis of parallel computation is conceptually convincing and provides a sharp mathematical form of the informal assumption(s) "underlying almost all models of concurrent systems". Gandy takes as the paradigmatic parallel computation, as I mentioned already, the evolution of the Game of Life or other cellular automata.

5.2 Gandy machines

Gandy uses, as Turing did, a *central thesis*: any discrete mechanical device satisfying some informal restrictive conditions can be described as a particular kind of dynamical system. Instead, I characterize a *Gandy Machine* axiomatically based on the following informal idea: the machine has to recognize the causal neighborhoods of a given state, act on them locally in parallel, and assemble the results to obtain the next state, which should be unique up to $\in$-isomorphism. In analogy to the definition of Turing computability, we call a function **F** *computable in parallel* if and only if there is a Gandy machine **M** whose computation results determine — under a suitable encoding and decoding — the values of **F** for any of its arguments. What then is the underlying notion of parallel computation?

Generalizing the above considerations for Turing computors, one notices quickly complications, when new atoms are introduced in the images of causal neighborhoods as well as in the next state: the different new atoms have to be "structurally coordinated". That can be achieved by a second local operation and a second set of stereotypes. Causal neighborhoods of type 1 are parts of neighborhoods of type 2 and the overlapping determined regions of type 1 must be parts of determined regions of type 2, so that they fit together appropriately. This generalization is absolutely crucial to allow the machine to assemble the determined regions. Here is the definition: $\mathbf{M} = \langle \mathbf{S}; \mathbf{T}_1, \mathbf{G}_1, \mathbf{T}_2, \mathbf{G}_2 \rangle$ is a *Gandy Machine on* $\mathbf{S}$, where $\mathbf{S}$ is a structural class, $\mathbf{T}_i$ a finite set of stereotypes, $\mathbf{G}_i$ a structural operation on $\cup\mathbf{T}_i$ ($i = 1$ or 2), if and only if for every $\mathbf{x} \in \mathbf{S}$ there is a $\mathbf{z} \in \mathbf{S}$ such that

(L.1) $(\forall \mathbf{y} \in \mathrm{Cn}_1(\mathbf{x}))(\exists! \mathbf{v} \in \mathrm{Dr}_1(\mathbf{z}, \mathbf{x}))\mathbf{v} \cong_{\mathbf{x}} \mathbf{G}_1(\mathbf{y})$;

(L.2) $(\forall \mathbf{y} \in \mathrm{Cn}_2(\mathbf{x}))(\exists \mathbf{v} \in \mathrm{Dr}_2(\mathbf{z}, \mathbf{x}))\mathbf{v} \cong_{\mathbf{x}} \mathbf{G}_2(\mathbf{y})$;

(A.1) $(\forall \mathbf{C})[\mathbf{C} \subseteq \mathrm{Dr}_1(\mathbf{z}, \mathbf{x}))\& \cap \{\ \mathrm{Sup}(\mathbf{v}) \cap \mathrm{A}(\mathbf{z}, \mathbf{x}) | \mathbf{v} \in \mathbf{C}\} \neq \varnothing \rightarrow$

$$(\exists \mathbf{w} \in \mathrm{Dr}_2(\mathbf{z}, \mathbf{x}))(\forall \mathbf{v} \in \mathbf{C})\mathbf{v} <^* \mathbf{w}];$$

(A.2) $\mathbf{z} = \cup\ \mathrm{Dr}_1(\mathbf{z}, \mathbf{x})$.

The condition $\cap\{\mathrm{Sup}(\mathbf{v}) \cap \mathrm{A}(\mathbf{z}, \mathbf{x}) | \mathbf{v} \in \mathbf{C}\} \neq \varnothing$ in (**A.1**) expresses that the determined regions $\mathbf{v}$ in $\mathbf{C}$ have new atoms in common, i.e., they *overlap*. — It might be helpful to the reader to look at section 5.4 and the description of the game of life as a Gandy machine one finds there.

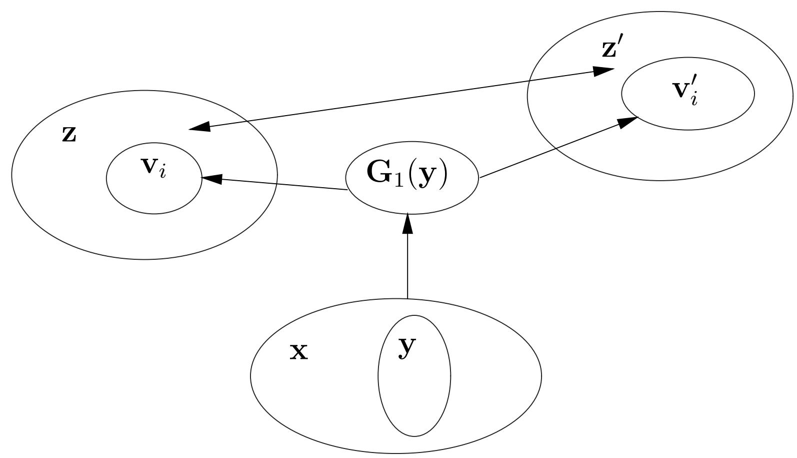
The restrictions for Gandy machines, as in the case of Turing computors, amount to boundedness and locality conditions. They are justified *directly* by two physical limitations, namely, a lower bound on the size of atomic components and an upper bound on the speed of signal propagation. I have completed now all the foundational work and can describe two important mathematical facts for Gandy machines: (i) the state $\mathbf{z}$ following $\mathbf{x}$ is determined uniquely up to $\in$-isomorphism over $\mathbf{x}$, and (ii) Turing machines can effect such transitions. The proof of the first claim contains the combinatorial heart of matters and uses crucially the assembly conditions. The proof of the second fact is rather direct. Only finitely many finite objects are involved in the transition, and all the axiomatic conditions are decidable. Thus, a search will allow us to find $\mathbf{z}$. This can be understood as a *Representation Theorem*: any particular Gandy machine is computationally equivalent to a two-letter Turing machine, as Turing machines are also Gandy machines. The first fact for Gandy machines, $\mathbf{z}$ is determined uniquely up to $\in$-isomorphism over $\mathbf{x}$, follows from the next theorem.[45] Before being able to formulate and prove

[45] In [Gandy, 1980] this uniqueness up to $\in$-isomorphism over $\mathbf{x}$ is achieved in a much more complex way, mainly, because parts of a state are proper subtrees, in general non-located. Given an appropriate definition of cover, a collection $\mathbf{C}$ is called an *assembly for* $\mathbf{x}$, if $\mathbf{C}$ is a cover for $\mathbf{x}$ and the elements of $\mathbf{C}$ are maximal. The fact that $\mathbf{C}$ is an assembly for exactly one $\mathbf{x}$, if indeed it is, is expressed by saying that $\mathbf{C}$ *uniquely assembles to* $\mathbf{x}$; see [Sieg and Byrnes, 1999a, 157]. In my setting, axiom (**A.2**) is equivalent to the claim that $\mathrm{Dr}_1(\mathbf{z},\mathbf{x})$ uniquely assembles to $\mathbf{z}$.

it, we need to introduce one more concept. A collection **C** of parts for **x** is a *cover for **x*** just in case $\mathbf{x} \subseteq \cup\, \mathbf{C}$.

THEOREM. Let **M** be $\langle \mathbf{S}; \mathbf{T}_1, \mathbf{G}_1, \mathbf{T}_2, \mathbf{G}_2 \rangle$ as above and $\mathbf{x} \in \mathbf{S}$; if there are **z** and $\mathbf{z}'$ in **S** satisfying principles (**L.1-2**), (**A.1**), and if $\mathrm{Dr}_1(\mathbf{z},\mathbf{x})$ and $\mathrm{Dr}_1(\mathbf{z}',\mathbf{x})$ cover **z** and $\mathbf{z}'$, then $\mathrm{Dr}_1(\mathbf{z},\mathbf{x}) \cong_{\mathbf{x}} \mathrm{Dr}_1(\mathbf{z}',\mathbf{x})$.

In the following Dr_1, Dr_1', A, and A$'$ will abbreviate $\mathrm{Dr}_1(\mathbf{z},\mathbf{x})$, $\mathrm{Dr}_1(\mathbf{z}',\mathbf{x})$, $\mathrm{A}(\mathbf{z},\mathbf{x})$, and $\mathrm{A}(\mathbf{z}',\mathbf{x})$ respectively. Note that Dr_1 and Dr_1' are finite. Using (**L.1**) and (**L.2**) one can observe that there is a natural number m and there are sequences $\mathbf{v}_i$ and $\mathbf{v}_i'$, $i < m$, such that $\mathrm{Dr}_1 = \{\mathbf{v}_i | i < m\}$, $\mathrm{Dr}_1' = \{\mathbf{v}_i' | i < m\}$, and $\mathbf{v}_i'$ is the unique part of $\mathbf{z}'$ with $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}_i'$ via permutations π_i (for all $i < m$). Here is a picture of the situation:



To establish the Theorem, we have to find a *single* permutation π that extends to an $\in$-isomorphism over **x** for all $\mathbf{v}_i$ and $\mathbf{v}_i'$ simultaneously. Such a π must obviously satisfy for all $i < m$:

$$\text{(i) } \mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}_i' \text{ via } \pi$$

and, consequently,

$$\text{(ii) } \pi[\mathrm{Sup}(\mathbf{v}_i)] = \mathrm{Sup}(\mathbf{v}_i').$$

As π is an $\in$-isomorphism over **x**, we have:

$$\text{(iii) } \pi[\mathrm{A}] = \mathrm{A}'.$$

Condition (ii) implies for all $i < m$ and all $\mathbf{r} \in \mathrm{A}$ the equivalence between $\mathbf{r} \in \mathrm{Sup}(\mathbf{v}_i)$ and $\mathbf{r}^{\pi} \in \mathrm{Sup}(\mathbf{v}_i')$. This can also be expressed by

$$\text{(ii*) } \mu(\mathbf{r}) = \mu'(\mathbf{r}^{\pi}), \text{ for all } \mathbf{r} \in \mathrm{A},$$

where $\mu(\mathbf{r}) = \{i|\mathbf{r} \in \mathrm{Sup}(\mathbf{v}_i)\}$ and $\mu'(\mathbf{r}) = \{i|\mathbf{r} \in \mathrm{Sup}(\mathbf{v}'_i)\}$; these are the *signatures* of $\mathbf{r}$ with respect to $\mathbf{z}$ and $\mathbf{z}'$.

To obtain such a permutation, the considerations are roughly as follows: (i) if the $\mathbf{v}_i$ do not overlap, then the π_i will do; (ii) if there is overlap, then an equivalence relation $\approx$ ($\approx'$) on A(A$'$) is defined by $\mathbf{r}_1 \approx \mathbf{r}_2$ iff $\mu(\mathbf{r}_1) = \mu(\mathbf{r}_2)$, and analogously for $\approx'$; (iii) then we prove that the "corresponding" equivalence classes $[\mathbf{r}]_\approx$ and $[\mathbf{s}]_{\approx'}$ (the signatures of their elements are identical) have the same cardinality. $[\mathbf{r}]_\approx$ can be characterized as $\cap\{\mathrm{Sup}(\mathbf{v}_i) \cap \mathrm{A}|i \in \mu(\mathbf{r})\}$; similar for $[\mathbf{s}]_{\approx'}$. This characterization is clearly independent of the choice of representative by the very definition of the equivalence relation(s). With this in place, a global $\in$-isomorphism can be defined. These considerations are made precise through the proofs of the combinatorial lemma and two corollaries in the next section.

5.3 Global assembly

All considerations in this section are carried out under the assumptions of the Theorem: $\mathbf{M} = \langle \mathbf{S}; \mathbf{T}_1, \mathbf{G}_1, \mathbf{T}_2, \mathbf{G}_2 \rangle$ is an arbitrary Gandy machine and $\mathbf{x} \in \mathbf{S}$ an arbitrary state; we assume furthermore that $\mathbf{z}$ and $\mathbf{z}'$ are in $\mathbf{S}$, the principles (**L.1-2**) and (**A.1**) are satisfied, and that Dr_1 and Dr'_1 cover $\mathbf{z}$ and $\mathbf{z}'$, because of (**A.2**). We want to show that $\mathrm{Dr}_1 \cong_{\mathbf{x}} \mathrm{Dr}'_1$, knowing already that there are sequences $\mathbf{v}_i$ and $\mathbf{v}'_i$ of length m, such that $\mathrm{Dr}_1 = \{\mathbf{v}_i |\ \mathrm{i{<}m}\}$, $\mathrm{Dr}'_1 = \{\mathbf{v}'_i |\ \mathrm{i{<}m}\}$ and $\mathbf{v}'_i$ is the unique part of $\mathbf{z}'$ with $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}'_i$ via permutations π_i (for all i<m). I start out with the formulation of a key lemma concerning overlaps.

LEMMA. (**Overlap Lemma**.) Let $\mathbf{r}_0 \in \mathrm{A}$ and $\mu(\mathbf{r}_0) \neq \varnothing$; then there is a permutation ρ on $\mathbf{U}$ with $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}'_i$ via ρ for all $i \in \mu(\mathbf{r}_0)$.

Proof. We have $\{\mathbf{v}_i|i \in \mu(\mathbf{r}_0)\} \subseteq \mathrm{Dr}_1$; as $\mathbf{r}_0$ is in A and in $\mathrm{Sup}(\mathbf{v}_i)$ for each $i \in \mu(\mathbf{r}_0) \neq \varnothing$, we have also that $\cap\{\mathrm{Sup}(\mathbf{v}_i) \cap \mathrm{A}|i \in \mu(\mathbf{r}_0)\} \neq \emptyset$. The antecedent of (**A.1**) is satisfied, and we conclude that there is a $\mathbf{w} \in \mathrm{Dr}_2$ such that $\mathbf{v}_i <^* \mathbf{w} <^* \mathbf{z}$, for all $i \in \mu(\mathbf{r}_0)$. Using (**L.2**) we obtain a $\mathbf{w}' \in \mathrm{Dr}'_2$ with $\mathbf{w} \cong_{\mathbf{x}} \mathbf{w}'$. This $\in$-isomorphism over $\mathbf{x}$ is induced by a permutation ρ and yields for all $i \in \mu(\mathbf{r}_0)$

$$\mathbf{v}_i^\rho <^* \mathbf{w}^\rho = \ \mathbf{w}' <^* \mathbf{z}'.$$

So we have, $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}_i^\rho$ and $\mathbf{v}_i^\rho <^* \mathbf{z}'$, thus — using (**L.1**) — $\mathbf{v}_i^\rho{=}\mathbf{v}'_i$; that holds for all $i \in \mu(\mathbf{r}_0)$. ■

Note that the condition $\mu(\mathbf{r}) \neq \varnothing$ is satisfied in our considerations for any $\mathbf{r} \in \mathrm{A}$, as $\mathrm{Dr}_1$is a cover of $\mathbf{z}$; so we have for any such $\mathbf{r}$ an appropiate *overlap permutation* $\rho^{\mathbf{r}}$ *for* $\mathbf{r}$. The crucial combinatorial lemma we have to establish is this:

LEMMA. (**Combinatorial Lemma**.) For $\mathbf{r}_0 \in \mathrm{A}: |\{\mathbf{r} \in \mathrm{A}|\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})\}| = |\{\mathbf{s} \in \mathrm{A}'|\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{s})\}|$.

Proof. Consider $\mathbf{r}_0 \in \mathrm{A}$. I establish first the claim

$$\rho[\{\mathbf{r} \in \mathrm{A}|\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})\}] \subseteq \{\mathbf{s} \in \mathrm{A}'|\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{s})\},$$

where ρ is an overlap permutation for $\mathbf{r}_0$. The claim follows easily from

$$\mathbf{r} \in \mathrm{A}\&\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r}) \to \mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{r}^\rho),$$

by observing that $\mathbf{r}^\rho$ is in A'. Assume, to establish this conditional indirectly, for arbitrary $\mathbf{r} \in \mathrm{A}$ that $\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})$ and $\neg(\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{r}^\rho))$. The first assumption implies that $\mathbf{r} \in \mathrm{Sup}(\mathbf{v}_i)$ for all $i \in \mu(\mathbf{r}_0)$, and the construction of ρ yields then:

$$(\heartsuit)\ \ \mathbf{r}^\rho \in \mathrm{Sup}(\mathbf{v}'_i) \text{ for all } i \in \mu(\mathbf{r}_0).$$

The second assumption implies that there is a k in $\mu(\mathbf{r}_0)\backslash\mu'(\mathbf{r}^\rho)$. Obviously, $k \in \mu(\mathbf{r}_0)$ and $k \notin \mu'(\mathbf{r}^\rho)$. The first conjunct $k \in \mu(\mathbf{r}_0)$ and $(\heartsuit)$ imply that $\mathbf{r}^\rho \in \mathrm{Sup}(\mathbf{v}'_k)$; as the second conjunct $k \notin \mu'(\mathbf{r}^\rho)$ means that $\mathbf{r}^\rho \notin \mathrm{Sup}(\mathbf{v}'_k)$, we have obtained a contradiction.

Now I'll show that $\rho[\{\mathbf{r} \in \mathrm{A}|\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})\}]$ cannot be a *proper* subset of $\{\mathbf{s} \in \mathrm{A}'|\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{s})\}$. Assume, to obtain a contradiction, that it is; then there is $\mathbf{s}^* \in \mathrm{A}'$ that satisfies $\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{s}^*)$ and is not a member of $\rho[\{\mathbf{r} \in \mathrm{A}|\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})\}]$. As $\mu(\mathbf{r}_0) \subseteq \mu'(\mathbf{s}^*), \mathbf{s}^*$ is in $\mathrm{Sup}(\mathbf{v}'_i)$ for all $i \in \mu(\mathbf{r}_0)$; the analogous fact holds for all $\mathbf{r} \in \mathrm{A}$ satisfying $\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r})$, i.e., all such $\mathbf{r}$ must be in $\mathrm{Sup}(\mathbf{v}_i)$ for all $i \in \mu(\mathbf{r}_0)$. As $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}'_i$ via ρ for all $i \in \mu(\mathbf{r}_0)$, $\mathbf{s}^*$ must be obtained as a ρ-image of some $\mathbf{r}^*$ in $\mathrm{Sup}(\mathbf{x})$ or in A (and, in the latter case, violating $\mu(\mathbf{r}_0) \subseteq \mu(\mathbf{r}^*)$). However, in either case we have a contradiction. The assertion of the Lemma is now immediate. ■

Next I establish two consequences of the Combinatorial Lemma, the second of which is basic for the definition of the global isomorphism π.

COROLLARY 1. For any $I \subseteq \{0, 1, \ldots, m-1\}$ with $I \subseteq \mu(\mathbf{r}_0)$ for some $\mathbf{r}_0$ in A,

$$|\{\mathbf{r} \in \mathrm{A}|I \subseteq \mu(\mathbf{r})\}| = |\{\mathbf{s} \in \mathrm{A}'|I \subseteq \mu'(\mathbf{s})\}|.$$

Proof. Consider an arbitrary $I \subseteq \mu(\mathbf{r}_0)$ for some $\mathbf{r}_0$ in A. If $I = \mu(\mathbf{r}_0)$, then the claim follows directly from the Combinatorial Lemma. If $I \subset \mu(\mathbf{r}_0)$, let $\mathbf{r}^0, \ldots, \mathbf{r}^{k-1}$ be elements $\mathbf{r}$ of A with $I \subset \mu(\mathbf{r})$ and require that $\mu(\mathbf{r}^j) \neq \mu(\mathbf{r}^{j'})$, for all $j, j' < k$ and $j \neq j'$, and for every $\mathbf{r} \in \mathrm{A}$ with $I \subset \mu(\mathbf{r})$ there is a unique $j < k$ with $\mu(\mathbf{r}) = \mu(\mathbf{r}^j)$. The Combinatorial Lemma implies, for all $j < k$,

$$|\{\mathbf{r} \in \mathrm{A}|\mu(\mathbf{r}^j) \subseteq \mu(\mathbf{r})\}| = |\{\mathbf{s} \in \mathrm{A}'|\mu(\mathbf{r}^j) \subseteq \mu'(\mathbf{s})\}|.$$

Now it is easy to verify the claim of Corollary 1:

$$\begin{aligned}
&|\{\mathbf{r} \in \mathrm{A}|I \subseteq \mu(\mathbf{r})\}| = \\
&|\{\mathbf{r} \in \mathrm{A}|(\exists j < k)\mu(\mathbf{r}^j) \subseteq \mu(\mathbf{r})\}| = \\
&|\{\mathbf{s} \in \mathrm{A}'|(\exists j < k)\mu(\mathbf{r}^j) \subseteq \mu'(\mathbf{s})\}| = \\
&|\{\mathbf{s} \in \mathrm{A}'|I \subseteq \mu'(\mathbf{s})\}|.
\end{aligned}$$

This completes the proof of Corollary 1. ■

The second important consequence of the Combinatorial Lemma can be obtained now by an inductive argument.

COROLLARY 2. For any $I \subseteq \{0, 1, \ldots, m-1\}$ with $I \subseteq \mu(\mathbf{r}_0)$ for some $\mathbf{r}_0$ in A.

$$|\{\mathbf{r} \in \mathrm{A} | I = \mu(\mathbf{r})\}| = |\{\mathbf{s} \in \mathrm{A}' | I = \mu'(\mathbf{s})\}|.$$

Proof. (By downward induction on $|I|$). Abbreviating $|\{\mathbf{r} \in \mathrm{A} | I = \mu(\mathbf{r})\}|$ by ν_I and $|\{\mathbf{s} \in \mathrm{A}' | I = \mu'(\mathbf{s})\}|$ by ν'_I, the argument is as follows:
Base case ($|I| = m$): In this case there are no proper extensions I^* of I, and we have

$$\begin{aligned}
\nu_I \quad &= |\{\mathbf{r} \in \mathrm{A} | I = \mu(\mathbf{r})\}| \\
&= |\{\mathbf{r} \in \mathrm{A} | I \subseteq \mu(\mathbf{r})\}|, \quad && \text{as there is no proper extension of } I, \\
&= |\{\mathbf{s} \in \mathrm{A}' | I \subseteq \mu'(\mathbf{s})\}|, && \text{by Corollary 1,} \\
&= |\{\mathbf{s} \in \mathrm{A}' | I = \mu'(\mathbf{s})\}|, && \text{again, as there is no proper extension,} \\
&= \nu'_I
\end{aligned}$$

Induction step ($|I|) < m$): Assume that the claim holds for all I^* with $n+1 \leq |I^*| \leq m$ and show that it holds for I with $|I| = n$. Using the induction hypothesis we have, summing up over all proper extensions I^* of I:

$$(\clubsuit)\ \Sigma_{I^*} \nu_{I^*} = \Sigma_{I^*} \nu'_{I^*}.$$

Now we argue as before:

$$\begin{aligned}
\nu_I \quad &= |\{\mathbf{r} \in \mathrm{A} | I = \mu(\mathbf{r})\}| \\
&= |\{\mathbf{r} \in \mathrm{A} | I \subseteq \mu(\mathbf{r})\}| - \Sigma_{I^*} \nu_{I^*} \\
&= |\{\mathbf{s} \in \mathrm{A}' | I \subseteq \mu'(\mathbf{s})\}| - \Sigma_{I^*} \nu'_{I^*}, \quad && \text{by Corollary 1 and } (\clubsuit), \\
&= |\{\mathbf{s} \in \mathrm{A}' | I = \mu'(\mathbf{s})\}| \\
&= \nu'_I
\end{aligned}$$

This completes the proof of Corollary 2. ■

Finally, we can define an appropriate global permutation π. Given an atom $\mathbf{r} \in \mathrm{A}$, there is an overlap permutations ρ^r, which can be restricted to

$$[\mathbf{r}]_{\approx} = \cap \{\mathrm{Sup}(\mathbf{v}_i) \cap \mathrm{A} | i \in \mu(\mathbf{r})\};$$

let ρ^* denote this restriction. Because of Corollary 2, $\rho*$ is a bijection between $[\mathbf{r}]_{\approx}$ and $[\rho^*(\mathbf{r})]_{\approx'}$. The desired global permutation is now defined as follows for any atom $\mathbf{r} \in \cup\{\mathrm{Sup}(\mathbf{v}_i) | i < m\}$:

$$\pi(\mathbf{r}) = \begin{cases} \rho^*(\mathbf{r}) & \text{if } \mathbf{r} \in \cap\{\mathrm{Sup}(\mathbf{v}_i) \cap \mathrm{A} |\ \mathrm{i} \in \mu\ (\mathbf{r})\} \\ \mathbf{r} & \text{otherwise} \end{cases}$$

π is a well-defined bijection with $\pi[\mathrm{A}] = \mathrm{A}'$ and $\mu(\mathbf{r}) = \mu'(\mathbf{r}^{\pi})$. It remains to establish:

Claim: For all $i < m$, $\mathbf{v}_i \cong_{\mathbf{x}} \mathbf{v}'_i$ via π.
For the ***Proof*** consider an arbitrary $i < m$. By the basic set-up of our considerations, we have $\pi_i(\mathbf{v}_i) = \mathbf{v}'_i$. If $\mathbf{v}_i$ does not contain in its support an element of A, then π and π_i coincide; if $\mathbf{v}_i$'s support contains an element of A that is possibly even in an overlap, the argument proceeds as follows. Notice first of all that *all* elements of $[\mathbf{r}]_\approx$ are in $\mathrm{Sup}(\mathbf{v}_i)$ as soon as one $\mathbf{r} \in$ A is in $\mathrm{Sup}(\mathbf{v}_i)$. Taking this into account, we have by definition of π and $\mathbf{v}_i \uparrow [\mathbf{r}]_\approx$: $\pi(\mathbf{v}_i \uparrow [\mathbf{r}]_\approx) = \rho^*(\mathbf{v}_i \uparrow [\mathbf{r}]_\approx)$.[46] The definition of ρ^* and the fact that $\rho^r(\mathbf{v}_i) = \mathbf{v}'_i$ allow us to infer that $\rho^*(\mathbf{v}_i \uparrow [\mathbf{r}]_\approx) = \mathbf{v}'_i \uparrow [\rho^*(\mathbf{r})]_{\approx'}$. As $\mu'(\rho^*(\mathbf{r})) = \mu'(\pi_i(\mathbf{r}))[= \mu(\mathbf{r})]$ we can extend this sequence of identities by $\mathbf{v}'_i \uparrow [\rho^*(\mathbf{r})]_{\approx'} = \mathbf{v}'_i \uparrow [\pi_i(\mathbf{r})]_{\approx'}$. Consequently, as $\pi_i(\mathbf{v}_i) = \mathbf{v}'_i$, we have $\mathbf{v}'_i \uparrow [\pi_i(\mathbf{r})]_{\approx'} = \pi_i(\mathbf{v}_i \uparrow [\mathbf{r}]_\approx)$.
These considerations hold for all $\mathbf{r} \in \mathrm{Sup}(\mathbf{v}_i) \cap \mathrm{A}$; we can conclude $\pi(\mathbf{v}_i) = \pi_i(\mathbf{v}_i)$ and, with $\pi_i(\mathbf{v}_i) = \mathbf{v}'_i$, we have $\pi(\mathbf{v}_i) = \mathrm{v}'_i$.

This concludes, finally, the argument for the Theorem.

5.4 Models

There is a rich variety of models, as the game of life, other cellular automata and many artificial neural nets are Gandy machines. Let me first sketch a set theoretic presentation of a Turing machine as a Turing computor and then, even more briefly, that of the Game of Life as a Gandy machine. Consider a Turing machine with symbols $s_0, \ldots, s_k$ and internal states $q_0, \ldots, q_m$; its program is given as a finite list of quadruples of the form $q_i s_j c_k q_m$, expressing that the machine is going to perform action c_k and change into internal state q_m, when scanning symbol s_j in state q_i. The tape is identified with a set of overlapping pairs

$$\mathbf{Tp} := \{\langle b, b\rangle, \langle b, c\rangle, \ldots, \langle d, e\rangle, \langle e, e\rangle\}$$

where $b, c, \ldots, d, e$ are distinct atoms; c is the leftmost square of the tape with a possibly non-blank symbol on it, d its rightmost one. The symbols are represented by $\underline{s}_j := \{r\}^{(j+1)}, 0 \le j \le k$; the internal states are given by $\underline{q}_j := \{r\}^{(k+1)+(j+1)}, 0 \le j \le l$. The tape content is given by

$$\mathbf{Ct} := \{\langle \underline{s}_{j_0}, c\rangle, \ldots, \langle \underline{s}_{j_r}, d\rangle\}$$

and, finally, the *id* is represented as the union of $\mathbf{Tp}$, $\mathbf{Ct}$, and $\{\langle \underline{q}_i, r\rangle\}$ with r being a square of $\mathbf{Tp}$. So the structural set $\mathbf{S}$ of states is obtained as the set of all ids closed under $\in$-isomorphisms. Stereotypes (for each program line given by $q_i s_j$) consist of parts like

$$\{\langle \underline{q}_i, r\rangle, \langle \underline{s}_j, r\rangle, \langle t, r\rangle, \langle r, u\rangle\};$$

[46] $\uparrow$ is the pruning operation; it applies to an element $\mathbf{x}$ of HF and a subset $\mathbf{Y}$ of its support: $\mathbf{x} \uparrow \mathbf{Y}$ is the subtree of $\mathbf{x}$ that is built up exclusively from atoms in $\mathbf{Y}$. The $\in$-recursive definition is: $(\mathbf{x} \cap \mathbf{Y}) \cup [\{\mathbf{y} \uparrow (\mathbf{Y} \cap Tc(\mathbf{y})) | \mathbf{y} \in \mathbf{x}\} \backslash \{\emptyset\}]$. Cf. [Sieg and Byrnes, 1999a, 155–6].

these are the causal neighborhoods on which **G** operates. Consider the program line $q_i s_j s_k q_l$ (print s_k); applied to the above causal neighborhood **G** yields

$$\{\langle \underline{q}_l, r\rangle, \langle \underline{s}_k, r\rangle, \langle t, r\rangle, \langle r, u\rangle\}.$$

For the program line $q_i s_j R q_l$ (move Right) two cases have to be distinguished. In the first case, when r is not the rightmost square, **G** yields

$$\{\langle \underline{q}_l, u\rangle, \langle \underline{s}_j, r\rangle, \langle t, r\rangle, \langle r, u\rangle\};$$

in the second case, when r is the rightmost square, **G** yields

$$\{\langle \underline{q}_l, *\rangle, \langle \underline{s}_j, r\rangle, \langle \underline{s}_0, *\rangle, \langle t, r\rangle, \langle r, *\rangle, \langle *, u\rangle\};$$

where $*$ is a new atom. The program line $q_i s_j L q_l$ (move Left) is treated similarly. It is easy to verify that a Turing machine presented in this way is a Turing Computor.

Cellular automata introduced by Ulam and von Neumann operate in parallel; a particular cellular automaton was made popular by Conway, the Game of Life. A cellular automaton is made up of many identical cells. Typically, each cell is located on a regular grid in the plane and carries one of two possible values. After each time unit its values are updated according to a simple rule that depends on its own previous value and the previous values of the neighboring cells. Cellular automata of this sort can simulate universal Turing machines, but they also yield discrete simulations of very general and complex physical processes.

Gandy considered playing Conway's Game of Life as a paradigmatic case of parallel computing. It is being played on subsets of the plane, more precisely, subsets that are constituted by finitely many connected squares. For reasons that will be obvious in a moment, the squares are also called *internal cells*; they can be in two states, *dead* or *alive*. In my presentation the internal cells are surrounded by one layer of *border cells*; the latter, in turn, by an additional layer of *virtual cells*. Border and virtual cells are dead by convention. Internal cells and border cells are jointly called *real*. The layering ensures that each real cell is surrounded by a full set of eight neighboring cells. For real cells the game is played according to the rules:

1. living cells with 0 or 1 (living) neighbor die (from isolation);
2. living cells with 4 or more (living) neighbors die (from overcrowding);
3. dead cells with exactly 3 (living) neighbors become alive.
4. In all other cases the cell's state is unchanged.

A real cell a with neighbors $a_1, \ldots, a_8$ and state $s(a)$ is given by

$$\{a, s(a), \langle a_1, \ldots, a_8\rangle\}.$$

The neighbors are given in "canonical" order starting with the square in the left-most top corner and proceeding clockwise; $s(a)$ is $\{a\}$ in case a is alive, otherwise $\{\{a\}\}$. The $\mathbf{T}_1$-causal neighborhoods of real cells are of the form

$$\{\{a, s(a), \langle a_1, \ldots, a_8\rangle\}, \{a_1, s(a_1)\}, \ldots, \{a_8, s(a_8)\}\}.$$

It is obvious how to define the structural operation $\mathbf{G}_1$ on the causal neighborhoods of internal cells; the case of border cells requires attention. There is a big number of stereotypes that have to be treated, so I will discuss only one simple case that should, nevertheless, bring out the principled considerations. In the following diagram we start out with the cells that have letters assigned to them; the diagram should be thought of extending at the left and at the bottom. The v's indicate virtual cells, the b's border cells, the $\{a\}$'s internal cells that are alive, and the $*$'s new atoms that are added in the next step of the computation. Let's see how that comes about.

$*_0$	$*_1$	$*_2$	$*_3$	$*_4$	$*_5$	$*_6$	$*_7$		
v_0	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9
b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	v_{10}
$\{a_0\}$	$\{a_1\}$	$\{a_2\}$	$\{a_3\}$	$\{a_4\}$	$\{a_5\}$	$\{a_6\}$	$\{a_7\}$	b_9	v_{11}

Consider the darkly shaded square b_3 with its neighbors, i.e., its presentation

$$\{b_3, \{\{b_3\}\}, \langle v_2, \ldots, b_2\rangle\};$$

applying $\mathbf{G}_1$ to its causal neighborhood yields

$$\{\{b_3, \{b_3\}, \langle v_2, \ldots, b_2\rangle\}, \{v_3, \{\{v_3\}\}, \langle *_2, *_3, *_4, v_4, b_4, b_3, b_2, v_2\rangle\}\},$$

where $*_2, *_3$, and $*_4$ are new atoms (and v_3 has been turned from a virtual cell into a real one, namely a border cell). Here the second set of stereotypes and the second structural operation come in to ensure that the new squares introduced by applying $\mathbf{G}_1$ to "adjacent" border cells (whose neighborhoods overlap with the neighborhood of b_3) are properly identified in the next state. Consider as the appropriate $\mathbf{T}_2$-causal neighborhood the set consisting of the $\mathbf{T}_1$-causal neighborhoods of b_2, b_3, and b_4; $\mathbf{G}_2$ applied to it yields the set with presentations of the cells v_2, v_3, and v_4.

5.5 *Tieferlegung*

The above considerations constitute the mathematical core of this section. They lead to the conclusion that computability, when relativized to a particular kind of computing agent or device, has a perfectly standard methodological status: no thesis is needed, but rather the recognition that the axiomatic characterization is correct for the intended computing device. The recognition that the notions do not go beyond Turing computability is then an important mathematical fact. It seems to me that we have gained in Hilbert's broad terms a deepening of the foundations via the axiomatic method, a *Tieferlegung der Fundamente*. As I mentioned earlier, Gödel advocated such an approach in a conversation with Church in early 1934 and suggested "to state a set of axioms which would embody the generally accepted properties of this notion (i.e., effective calculability), and to do something on that basis."

The sharpened version of Turing's work and a thorough-going re-interpretation of Gandy's approach allow us to fill in the blanks of Gödel's suggestion; this resolves in my view the methodological issue raised at the end of section 4. Perhaps the remarks in the 1964 Postscriptum to the Princeton Lectures of 1934 echo his earlier considerations. "Turing's work gives," according to Gödel, "an analysis of the concept of 'mechanical procedure'.... This concept is shown to be equivalent with that of a 'Turing machine'." The work, on which I reported, substantiates these remarks in the following sense: it provides an axiomatic analysis of the concept "mechanical procedure" and shows that this concept is computationally equivalent to that of a Turing machine. Indeed, it does so for two such concepts, namely, when the computing agents are computors or discrete machines; and it does so by imposing constraints on the computations these agents carry out in steps. The natural and well-motivated constraints guarantee the effectiveness of the steps in the most direct way.

The axiomatic approach captures the essential nature of computation processes in an abstract way. The difference between the two types of calculators I have been describing is reduced to the fact that Turing computors modify *one* bounded part of a state, whereas Gandy machines operate in parallel on *arbitrarily many* bounded parts. The axiomatic conditions arise from underlying analyses that lead to a particular structural view. Of course, an appeal to some informal understanding can no more be avoided in this case than in any other case of an axiomatically characterized mathematical structure intended to model broad aspects of physical or intellectual reality. The general point is this: we don't have to face anything especially mysterious for the concept of calculability; rather, we have to face the ordinary issues for the adequacy of mathematical concepts and they are, of course, non-trivial.

I have been distinguishing in other writings two aspects of mathematical experience. The first, the *quasi-constructive* aspect, has to do with the recognition of laws for accessible domains; this includes, in particular, our recognition of the correctness of the Zermelo Fraenkel axioms in set theory and their extendibility by suitable axioms of infinity. The second, the *conceptional* aspect, deals with the uncovering of abstract, axiomatically characterized notions. These abstract notions are distilled from mathematical practice for the purpose of comprehending complex connections, of making analogies precise and of obtaining a more profound understanding. Bourbaki in their [1950] expressed matters quite in Dedekind and Hilbert's spirit, when claiming that the axiomatic method teaches us

> to look for the deep-lying reasons for such a discovery [that two or several quite distinct theories lend each other "unexpected support"], to find the common ideas of these theories, ... to bring these ideas forward and to put them in their proper light. (p. 223)

Notions like group, field, topological space and differentiable manifold are abstract in this sense. Turing's analysis shows, when properly generalized, that computability exemplifies the second aspect of mathematical experience. Although Gödel used "abstract" in a more inclusive way than I do here his broad claim is pertinent also for computability, namely, "that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding." [1972, 306]

6 OUTLOOK ON MACHINES AND MIND

Turing's notion of human computability is exactly right not only for obtaining a negative solution of the Entscheidungsproblem that is conclusive, but also for achieving a precise characterization of formal systems that is needed for the general formulation of Gödel's incompleteness theorems. I argued in sections 1 and 2 that the specific intellectual context reaches back to Leibniz and requires us to focus attention on effective, indeed mechanical procedures; these procedures are to be carried out by computors without invoking higher cognitive capacities. The axioms of section 5.1 are intended for this informal concept. The question whether there are strictly broader notions of effectiveness has of course been asked for both cognitive and physical processes. I am going to address this question not in any general and comprehensive way, but rather by focusing on one central issue: the discussion might be viewed as a congenial dialogue between Gödel and Turing on aspects of mathematical reasoning that transcend mechanical procedures.

I'll start in section 6.1 by returning more fully to Gödel's view on mechanical computability as articulated in his [193?]. There he drew a dramatic conclusion from the undecidability of certain Diophantine propositions, namely, that mathematicians cannot be replaced by machines. That theme is taken up in the Gibbs Lecture of 1951 where Gödel argues in greater detail that the human mind infinitely surpasses the powers of any finite machine; an analysis of the argument is presented in section 6.2 under the heading *Beyond calculation*. Section 6.3 is entitled *Beyond discipline* and gives Turing's perspective on intelligent machinery; it is devoted to the seemingly sharp conflict between Gödel's and Turing's views on mind. Their deeper disagreement really concerns the nature of machines, and I'll end with some brief remarks on (supra-) mechanical devices in section 6.4.

6.1 Mechanical computability

In section 4.2 I alluded briefly to the unpublished and untitled draft for a lecture Gödel presumably never delivered; it was written in the late 1930s. Here one finds the earliest extensive discussion of Turing and the reason why Gödel, at the time, thought Turing had established "beyond any doubt" that "this really is the correct definition of mechanical computability". Obviously, we have to clarify what "this" refers to, but first I want to give some of the surrounding context. Already in his [1933] Gödel elucidated, as others had done before him, the mechanical feature of effective procedures by pointing to the possibility that machines carry them out. When insisting that the inference rules of precisely described proof methods have to be "purely formal" he explains:

> [The inference rules] refer only to the outward structure of the formulas, not to their meaning, so that they could be applied by someone who knew nothing about mathematics, or by a machine. This has the consequence that there can never be any doubt as to what cases the rules of inference apply to, and thus the highest possible degree of exactness is obtained. [*Collected Works III*, p. 45]

During the spring term of 1939 Gödel gave an introductory logic course at Notre Dame. The logical decision problem is informally discussed and seen in the historical context of Leibniz's "Calculemus".[47] Before arguing that results of modern logic prevent the realization of Leibniz's project, Gödel asserts that the rules of logic can be applied in a "purely mechanical" way and that it is therefore possible "to construct a machine which would do the following thing":

> The supposed machine is to have a crank and whenever you turn the crank once around the machine would write down a tautology of the calculus of predicates and it would write down every existing tautology ... if you turn the crank sufficiently often. So this machine would really replace thinking completely as far as deriving of formulas of the calculus of predicates is concerned. It would be a thinking machine in the literal sense of the word. For the calculus of propositions you can do even more. You could construct a machine in form of a typewriter such that if you type down a formula of the calculus of propositions then the machine would ring a bell [if the formula is a tautology] and if it is not it would not. You could do the same thing for the calculus of monadic predicates.

[47] This is [Gödel 1939]. As to the character of these lectures, see [Dawson], p. 135.

Having formulated these positive results Gödel points out that "it is impossible to construct a machine which would do the same thing for the whole calculus of predicates". Drawing on the undecidability of predicate logic established by Church and Turing, he continues with a striking claim:

> So here already one can prove that Leibnitzens [sic!] program of the "calculemus" cannot be carried through, i.e. one knows that the human mind will never be able to be replaced by a machine already for this comparatively simple question to decide whether a formula is a tautology or not.

I mention these matters to indicate the fascination Gödel had with the mechanical realization of logical procedures, but also his *penchant* for overly dramatic formulations concerning the human mind. He takes obviously for granted here that a mathematically satisfactory definition of mechanical procedures has been given.

Such a definition, Gödel insists in [193?, 166], is provided by the work of Herbrand, Church and Turing. In that manuscript he examines the relation between mechanical computability, general recursiveness and machine computability. This is of special interest, as we will see that his methodological perspective here is quite different from his later standpoint. He gives, on pp. 167–8, a perspicuous presentation of the equational calculus that is "essentially Herbrand's" and defines general recursive functions. He claims outright that it provides "the correct definition of a computable function". Then he asserts, "That this really is the correct definition of mechanical computability was established beyond any doubt by Turing." Here the referent for "this" has finally been revealed: it is the definition of general recursive functions. How did Turing establish that this is also the correct definition of *mechanical* computability? Gödel's answer is as follows:

> He [Turing] has shown that the computable functions defined in this way [via the equational calculus] are exactly those for which you can construct a machine with a finite number of parts which will do the following thing. If you write down any number $n_1, \ldots, n_r$ on a slip of paper and put the slip of paper into the machine and turn the crank, then after a finite number of turns the machine will stop and the value of the function for the argument $n_1, \ldots, n_r$ will be printed on the paper. [*Collected Works III*, p. 168]

The implicit claim is clearly that a procedure is mechanical just in case it is executable by a machine with a finite number of parts. There is no indication of the structure of such machines except for the insistence that they have only finitely many parts, whereas Turing machines are of course potentially infinite due to the expanding tape.

The literal reading of the argument for the claim "this really is the correct definition of mechanical computability was established beyond any doubt by Turing" amounts to this. The equational calculus characterizes the computations of number-theoretic functions and provides thus "the correct definition of computable function". That the class of computable functions is co-extensional with that of *mechanically* computable ones is then guaranteed by "Turing's proof" of the equivalence between general recursiveness and machine computability.[48] Consequently, the definition of general recursive functions via the equational calculus characterizes correctly the mechanically computable functions. Without any explicit reason for the first step in this argument, it can only be viewed as a direct appeal to Church's Thesis.

If we go beyond the literal reading and think through the argument in parallel to Turing's analysis in his [1936], then we can interpret matters as follows. Turing considers arithmetic calculations done by a computor. He argues that they involve only very elementary processes; these processes can be carried out by a Turing machine operating on strings of symbols. Gödel, this interpretation maintains, also considers arithmetic calculations done by a computor; these calculations can be reduced to computations in the equational calculus. This first step is taken in parallel by Gödel and Turing and is based on a conceptual analysis; cf. the next paragraph. The second step connects calculations of a computor to computations of a Turing machine. This connection is established by mathematical arguments: Turing simply states that machines operating on finite strings can be proved to be equivalent to machines operating on individual symbols, i.e., to ordinary Turing machines; Gödel appeals to "Turing's proof" of the fact that general recursiveness and machine computability are equivalent.

Notice that in Gödel's way of thinking about matters at this juncture, the *mathematical theorem* stating the equivalence of general recursiveness and machine computability plays the pivotal role: It is not Turing's analysis that is appealed to by Gödel but rather "Turing's proof". The central analytic claim my interpretation attributes to Gödel is hardly argued for. On p. 13 Gödel just asserts, "... by analyzing in which manner this calculation [of the values of a general recursive function] proceeds you will find that it makes use only of the two following rules." The two rules as formulated here allow substituting numerals for variables and equals for equals. So, in some sense, Gödel seems to think that the rules of the equational calculus provide a way of "canonically" representing steps in calculations and, in addition, that his characterization of recursion is the most general one.[49] The latter is imposed by the requirement that function values have to be calculated, as pointed out in [1934, 369 top]; the former is emphasized much later in a letter to van Heijenoort of April 23, 1963, where Gödel distinguishes his definition from Herbrand's. His definition, Gödel asserts, brought out clearly what Herbrand had failed to see, namely "that the computation (for all computable functions) proceeds by exactly the same rules". [*Collected Works V*, p. 308] By

[48]In Turing's [1936] general recursive functions are not mentioned. Turing established in an Appendix to his paper the equivalence of his notion with λ-definability. As Church and Kleene had already proved the equivalence of λ-definability and general recursiveness, "Turing's Theorem" is thus established for Turing computability.

[49]This is obviously in contrast to the view he had in 1934 when defining general recursive functions; cf. section 3.2.

contrast, Turing shifts from arithmetically meaningful steps to symbolic processes that underlie them and can be taken to satisfy restrictive boundedness as well as locality conditions. These conditions cannot be imposed directly on arithmetic steps and are certainly not satisfied by computations in the equational calculus. So, we are back precisely at the point of the discussion in section 3.

6.2 Beyond calculation

In [193?] Gödel begins the discussion by reference to Hilbert's "famous words" that "for any precisely formulated mathematical question a unique answer can be found". He takes these words to mean that for any mathematical proposition A there is a proof of either A or not-A, "where by 'proof' is meant something which starts from evident axioms and proceeds by evident inferences". He argues that the incompleteness theorems show that something is lost when one takes the step from this notion of proof to a formalized one: "... it is not possible to formalise mathematical evidence even in the domain of number theory, but the conviction about which Hilbert speaks remains entirely untouched. Another way of putting the result is this: it is not possible to mechanise mathematical reasoning; ..." Then he continues, in a way that is similar to the striking remark in the Notre Dame Lectures, "i.e., it will never be possible to replace the mathematician by a machine, even if you confine yourself to number-theoretic problems." (pp. 164–5)

The succinct argument for this conclusion is refined in the Gibbs Lecture of 1951. In the second and longer part of the lecture, Gödel gave the most sustained defense of his Platonist standpoint drawing the "philosophical implications" of the situation presented by the incompleteness theorems.[50] "Of course," he says polemically, "in consequence of the undeveloped state of philosophy in our days, you must not expect these inferences to be drawn with mathematical rigor." The mathematical aspect of the situation, he claims, can be described rigorously; it is formulated as a disjunction, "Either mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule, that is to say, the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite machine, or else there exist absolutely unsolvable Diophantine problems of the type specified ..." Gödel insists that this fact is both "mathematically established" and of "great philosophical interest". He presents on pages 11–13 an argument for the disjunction and considers its conclusion as "inevitable".

The disjunction is called in footnote 15 a theorem that holds for finitists and intuitionists as an implication. Here is the appropriate implication: If the evident axioms of mathematics can be comprised in a finite rule, then there exist absolutely unsolvable Diophantine problems. Let us establish this implication by adapting Gödel's considerations for the disjunctive conclusion; the argument is

[50] That standpoint is formulated at the very end of the lecture as follows: p. 38 (CW III, 322/3): "Thereby [i.e., the Platonistic view] I mean the view that mathematics describes a non-sensual reality, which exists independently both of the acts and [[of]] the dispositions of the human mind and is only perceived, and probably perceived very incompletely, by the human mind."

brief. Assume the axioms that are evident for the human mind can be comprised in a finite rule "that is to say", for Gödel, a Turing machine can list them. Thus there exists a mechanical rule producing all the evident axioms for "subjective" mathematics, which is by definition the system of all humanly demonstrable mathematical propositions.[51] On pain of contradiction with the second incompleteness theorem, the human mind cannot prove the consistency of subjective mathematics. (This step is of course justified only if the inferential apparatus for subjective mathematics is given by a mechanical rule, and if subjective mathematics satisfies all the other conditions for the applicability of the second theorem.) Consequently, the Diophantine problem corresponding to the consistency statement cannot be proved either in subjective mathematics. That justifies Gödel's broader claim that it is undecidable "not just within some particular axiomatic system, but by *any* mathematical proof the human mind can conceive". (p. 13) In this sense the problem is *absolutely undecidable* for the human mind. So it seems that we have established the implication. However, the very first step in this argument, indicated by "that is to say", appeals to the precise concept of "finite procedure" as analyzed by Turing. Why is "that is to say" justified for Gödel? — To answer this question, I examine Gödel's earlier remarks about finite procedures and finite machines.[52]

Gödel stresses in the first paragraph of the Gibbs Lecture that the incompleteness theorems have taken on "a much more satisfactory form than they had had originally". The greatest improvement was made possible, he underlines, "through the precise definition of the concept of finite procedure, which plays a decisive role in these results". Though there are a number of different ways of arriving at such a definition which all lead to "exactly the same concept", the most satisfactory way is that taken by Turing when "reducing the concept of a finite procedure to that of a machine with a finite number of parts". Gödel does not indicate the character of, or an argument for, the reduction of finite procedures to procedures effected by a machine with a finite number of parts, but he states explicitly that he takes finite machine "in the precise sense" of a Turing machine. (p. 9) This reduction is pivotal for establishing the central implication rigorously, and it is thus crucial to understand and grasp its *mathematical* character. How else can we assent to the claim that the implication has been established mathematically as a theorem? In his [1964] Gödel expressed matters quite differently (and we discussed that later Gödelian perspective extensively in section 4): there he asserts that Turing in [1936] gave an analysis of mechanical procedures and showed that the analyzed

[51] This is in contrast to the case of "objective" mathematics, the system of all true mathematical propositions, for which one cannot have a "well-defined system of correct axioms" (given by a finite rule) that comprises all of it. In [Wang, 1974, 324–6], Gödel's position on these issues is (uncritically) discussed. The disjunction is presented as one of "two most interesting rigorously proved results about minds and machines" and is formulated as follows: "Either the human mind surpasses all machines (to be more precise: it can decide more number theoretic questions than any machine) or else there exist number theoretical questions undecidable for the human mind."

[52] Boolos' *Introductory Note* to the Gibbs Lecture, in particular section 3, gives a different perspective on difficulties in the argument.

concept is equivalent to that of a Turing machine. The claimed equivalence is viewed as central for obtaining "a precise and unquestionably adequate definition of the general concept of formal system" and for supporting, I would like to add in the current context, the mathematical cogency of the argument for the implication.

Gödel neither proved the mathematical conclusiveness of the reduction nor the correctness of the equivalence. So let us assume, for the sake of the argument, that the implication has been mathematically established and see what conclusions of great philosophical interest can be drawn. There is, as a first background assumption, Gödel's deeply rationalist and optimistic perspective that denies the consequent of the implication. That perspective, shared with Hilbert as we saw in section 6.1, was articulated in [193?], and it was still taken in the early 1970s. Wang reports in [1974, 324–5], that Gödel agreed with Hilbert in rejecting the possibility that there are number-theoretic problems undecidable for the human mind. Our task is then to follow the path of Gödel's reflections on the first alternative of his disjunction or the negated antecedent of our implication. That assertion states: There is no finite machine (i.e. no Turing machine) that lists all the axioms of mathematics which are evident to the human mind. Gödel argues for two related conclusions: i) the working of the human mind is not reducible to operations of the brain, and ii) the human mind infinitely surpasses the powers of any finite machine.[53]

A second background assumption is introduced to obtain the first conclusion: The brain, "to all appearances", is "a finite machine with a finite number of parts, namely, the neurons and their connections". (p. 15) As finite machines are taken to be Turing machines, brains are consequently also considered as Turing machines. That is reiterated in [Wang, 1974, 326], where Gödel views it as very likely that "The brain functions basically like a digital computer." Together with the above assertion this allows Gödel to conclude in the Gibbs Lecture, "the working of the human mind cannot be reduced to the working of the brain".[54] In [Wang] it is taken to be in conflict with the commonly accepted view, "There is no mind separate from matter." That view is for Gödel a "prejudice of our time, which will be disproved scientifically (perhaps by the fact that there aren't enough nerve cells to perform the observable operations of the mind)". Gödel uses the notion of a finite machine in an extremely general way when considering the brain as a finite machine with a finite number of parts. It is here that the identification of finite machines with Turing machines becomes evidently problematic: Is it at all plausible to think that the brain has a similarly fixed structure and fixed program as a particular Turing machine? The argumentation is problematic also on different grounds; namely, Gödel takes "human mind" in a more general way than just the

[53]This does not follow just from the fact that for every Turing machine that lists evident axioms there is another axiom evident to the human mind not included in the list. Turing had tried already in his 1939 paper, *Ordinal Logics*, to overcome the incompleteness results by strengthening theories systematically. He added consistency statements (or reflection principles) and iterated this step along constructive ordinals; Feferman perfected that line of investigation, cf. his [1988]. Such a procedure was also envisioned in [Gödel, 1946, 1–2].

[54]Cf. also note 13 of the Gibbs Lecture and the remark on p. 17.

mind of any one individual human being. Why should it be then that mind is realized through any particular brain?

The proposition that the working of the human mind cannot be reduced to the working of the brain is thus not obtained as a "direct" consequence of the incompleteness theorems, but requires additional substantive assumptions: i) there are no Diophantine problems the human mind cannot solve, ii) brains are finite machines with finitely many parts, and iii) finite machines with finitely many parts are Turing machines. None of these assumptions is uncontroversial; what seems not to be controversial, however, is Gödel's more open formulation in [193?] that it is not possible to mechanize mathematical reasoning. That raises immediately the question, what aspects of mathematical reasoning or experience defy formalization? In his note [1974] that was published in [Wang, 325–6], Gödel points to two "vaguely defined" processes that may lead to systematic and effective, but non-mechanical procedures, namely, the process of defining recursive well-orderings of integers for larger and larger ordinals of the second number class and that of formulating stronger and stronger axioms of infinity. The point was reiterated in a modified formulation [Gödel, 1972.3] that was published only later in *Collected Works II*, p. 306. The [1972.3] formulation of this note is preceded by [1972.2], where Gödel gives *Another version of the first undecidability theorem* that involves number theoretic problems of Goldbach type. This version of the theorem may be taken, Gödel states, "as an indication for the existence of mathematical yes or no questions undecidable for the human mind". (p. 305) However, he points to a *fact* that "weighs against this interpretation", namely, that "there *do* exist unexplored series of axioms which are analytic in the sense that they only explicate the concepts occurring in them". As an example he points also here to axioms of infinity, "which only explicate the content of the general concept of set". (p. 306) If the existence of such effective, non-mechanical procedures is taken as a fact or, more cautiously, as a third background assumption, then Gödel's second conclusion is established: The human mind, indeed, infinitely surpasses the power of any finite machine.

Though Gödel calls the existence of an "unexplored series" of axioms of infinity a *fact*, he also views it as a "vaguely defined" procedure and emphasizes that it requires further mathematical experience; after all, its formulation can be given only once set theory has been developed "to a considerable extent". In the note [1972.3] Gödel suggests that the process of forming stronger and stronger axioms of infinity does not yet form a "well-defined procedure which could actually be carried out (and would yield a non-recursive number-theoretic function)": it would require "a substantial advance in our understanding of the basic concepts of mathematics". In the note [1974], Gödel offers a *prima facie* startlingly different reason for not yet having a precise definition of such a procedure: it "would require a substantial deepening of our understanding of the basic operations of the mind". (p. 325)

Gödel's *Remarks before the Princeton bicentennial conference* in 1946 throw some light on this seeming tension. Gödel discusses there not only the role axioms of infinity might play in possibly obtaining an absolute concept of demonstrability, but he also explores the possibility of an absolute mathematical "definition of definability". What is most interesting for our considerations here is the fact that he considers a restricted concept of *human definability* that would reflect a human capacity, namely, "comprehensibility by our mind". That concept should satisfy, he thinks, the "postulate of denumerability" and in particular allow us to define (in this particular sense) only countably many sets. "For it has some plausibility that all things conceivable by us are denumerable, even if you disregard the question of expressibility in some language." (p. 3) That requirement, together with the related difficulty of the definability of the least indefinable ordinal, does not make such a concept of definability "impossible, but only [means] that it would involve some extramathematical element concerning the psychology of the being who deals with mathematics." Obviously, Turing brought to bear on his definition of computability, most fruitfully, an extramathematical feature of the psychology of a human computor.[55] Gödel viewed that definition in [1946], the reader may recall, as the first "absolute definition of an interesting epistemological notion". (p. 1) His reflections on the possibility of absolute definitions of demonstrability and definability were encouraged by the success in the case of computability. Can we obtain by a detailed study of *actual* mathematical experience a deeper "understanding of the basic operations of the mind" and thus make also a "substantial advance in our understanding of the basic concepts of mathematics"?

6.3 Beyond discipline

Gödel's brief exploration in [1972.3] of the issue of defining a non-mechanical, but effective procedure is preceded by a severe critique of Turing. The critical attitude is indicated already by the descriptive and harshly judging title of the note, *A philosophical error in Turing's work*. The discussion of Church's thesis and Turing's analysis is in general fraught with controversy and misunderstanding, and the controversy begins often with a dispute over what the intended informal concept is. When Gödel spotted a philosophical error in Turing's work, he *assumed* that Turing's argument in the 1936 paper was to show that "mental procedures cannot go beyond mechanical procedures". He considered the argument as inconclusive:

> What Turing disregards completely is the fact that *mind, in its use, is not static, but constantly developing*, i.e., that we understand abstract terms more and more precisely as we go on using them, and that more and more abstract terms enter the sphere of our understanding. [*Collected Works II*, p. 306]

Turing did not give a conclusive argument for Gödel's claim, but then it has to be added that he did not intend to argue for it. Simply carrying out a mechanical procedure does not, indeed, should not involve an expansion of our understanding. Turing viewed the restricted use of mind in computations undoubtedly as static;

[55]Cf. Parsons' informative remarks in the *Introductory Note* to [Gödel, 1946, 148].

after all, it seems that this feature contributed to the good reasons for replacing states of mind of the human computor by "more definite physical counterparts" in section 9, part III, of his classical paper.

Even in his work of the late 1940s and early 1950s that deals explicitly with mental processes, Turing does not argue that mental procedures cannot go beyond mechanical procedures. Mechanical processes are, as a matter of fact, still made precise as Turing machine computations; machines that might exhibit intelligence have, in contrast, a more complex structure than Turing machines. Conceptual idealization and empirical adequacy are now being sought for quite different purposes, and Turing is trying to capture clearly what Gödel found missing in his analysis for a broader concept of humanly effective calculability, namely, "... that mind, in its use, is not static, but constantly developing".[56] Gödel continued the above remark in this way:

> There may exist systematic methods of actualizing this development, which could form part of the procedure. Therefore, although at each stage the number and precision of the abstract terms at our disposal may be *finite*, both (and, therefore, also Turing's number of *distinguishable states of mind*) may *converge toward infinity* in the course of the application of the procedure.

The particular procedure mentioned as a plausible candidate for satisfying this description is the process of forming stronger and stronger axioms of infinity. We saw that the two notes, [1972-3] and [1974], are very closely connected. However, there is one subtle and yet substantive difference. In [1974] the claim that the number of possible states of mind may converge to infinity is obtained as a consequence of the dynamic development of mind. That claim is then followed by a remark that begins, in a superficially similar way, as the first sentence in the above quotation:

> Now there may exist systematic methods of accelerating, specializing, and uniquely determining this development, e.g. by asking the right questions on the basis of a mechanical procedure.

[56][Gödel, 1972.3] may be viewed, Gödel mentions, as a note to the word "mathematics" in the sentence, "Note that the results mentioned in this postscript do not establish any bounds of the powers of human reason, but rather for the potentialities of pure formalism in mathematics." This sentence appears in the 1964 Postscriptum to the Princeton Lectures Gödel gave in 1934; *Collected Works I*, pp. 369–371. He states in that Postscriptum also that there may be "finite non-mechanical procedures" and emphasizes, as he does in many other contexts, that such procedures would "involve the use of abstract terms on the basis of their meaning". (Note 36 on p. 370 of *Collected Works I*) Other contexts are found in volume III of the *Collected Works*, for example, the Gibbs Lecture (p. 318 and note 27 on that very page) and a related passage in "Is mathematics syntax of language?" (p. 344 and note 24) These are systematically connected to Gödel's reflections surrounding (the translation of) his Dialectica paper [1958] and [1972]. A thorough discussion of these issues cannot be given here; but as to my perspective on the basic difficulties, see the discussion in section 4 of my paper "Beyond Hilbert's Reach?".

Clearly, I don't have a full understanding of these enigmatic observations, but there are three aspects that are clear enough. First, mathematical experience has to be invoked when asking the right questions; second, aspects of that experience may be codified in a mechanical procedure and serve as the basis for the right questions; third, the answers may involve abstract terms that are incorporated into the non-mechanical mental procedure.

We should not dismiss or disregard Gödel's methodological remark that "asking the right questions on the basis of a mechanical procedure" may be part of a systematic method to push forward the development of mind. It allows us, even on the basis of a very limited understanding, to relate Gödel's reflections tenuously with Turing's proposal for investigating matters. Prima facie their perspectives are radically different, as Gödel proceeds by philosophical argument and broad, speculative appeal to mathematical experience, whereas Turing suggests attacking the problem largely by computational experimentation. That standard view of the situation is quite incomplete. In his paper *Intelligent machinery* written about ten years after [1939], Turing states what is really the central problem of cognitive psychology:

> If the untrained infant's mind is to become an intelligent one, it must acquire both discipline and initiative. So far we have been considering only discipline [via the universal machine, W.S.]. ... But discipline is certainly not enough in itself to produce intelligence. That which is required in addition we call initiative. This statement will have to serve as a definition. Our task is to discover the nature of this residue as it occurs in man, and to try and copy it in machines. (p. 21)

How can we transcend discipline? A hint is provided in Turing's 1939 paper, where he distinguishes between ingenuity and intuition. He observes that in formal logics their respective roles take on a greater definiteness. Intuition is used for "setting down formal rules for inferences which are always intuitively valid", whereas ingenuity is to "determine which steps are the more profitable for the purpose of proving a particular proposition". He notes:

> In pre-Gödel times it was thought by some that it would be possible to carry this programme to such a point that all the intuitive judgements of mathematics could be replaced by a finite number of these rules. The necessity for intuition would then be entirely eliminated. (p. 209)

The distinction between ingenuity and intuition, but also the explicit link of intuition to incompleteness, provides an entry to exploit through concrete computational work the "parallelism" of Turing's and Gödel's considerations. Copying the residue in machines is the task at hand. It is extremely difficult in the case of mathematical thinking, and Gödel would argue it is an impossible one, if machines are Turing machines. Turing would agree. Before we can start copying, we have to discover at least partially the nature of the residue, with an emphasis on "partially", through some restricted proposals for finding proofs in mathematics. Let us look briefly at the broad setting.

Proofs in a formal logic can be obtained uniformly by a patient search through an enumeration of all theorems, but additional intuitive steps remain necessary because of the incompleteness theorems. Turing suggested particular intuitive steps in his ordinal logics; his arguments are theoretical, but connect directly to the discussion of actual or projected computing devices that appears in his *Lecture to London Mathematical Society* and in *Intelligent Machinery*. In these papers he calls for intellectual searches (i.e., heuristically guided searches) and initiative (that includes, in the context of mathematics, proposing new intuitive steps). However, he emphasizes [1947, 122]:

> As regards mathematical philosophy, since the machines will be doing more and more mathematics themselves, the centre of gravity of the human interest will be driven further and further into philosophical questions of what can in principle be done etc.

Gödel and Turing, it seems, could have cooperated on the philosophical questions of what can in principle be done. They also could have agreed, so to speak terminologically, that there is a human mind whose working is not reducible to the working of any particular brain. Towards the end of *Intelligent Machinery* Turing emphasizes, "the isolated man does not develop any intellectual power", and argues:

> It is necessary for him to be immersed in an environment of other men, whose techniques he absorbs during the first twenty years of his life. He may then perhaps do a little research of his own and make a very few discoveries which are passed on to other men. From this point of view the search for new techniques must be regarded as carried out by the human community as a whole, rather than by individuals.

Turing calls this, appropriately enough, a *cultural search* and contrasts it with more limited, *intellectual searches*. Such searches, Turing says definitionally, can be carried out by individual brains. In the case of mathematics they would include searches through all proofs and would be at the center of "research into intelligence of machinery". Turing had high expectations for machines' progress in mathematics; indeed, he was unreasonably optimistic about their emerging capacities. Even now it is a real difficulty to have machines do mathematics on their own: work on Gödel's "theoretical" questions has to be complemented by sustained efforts to meet Turing's "practical" challenge. I take this to be one of the ultimate motivations for having machines find proofs in mathematics, i.e., proofs that reflect logical as well as mathematical understanding.

When focusing on proof search in mathematics it may be possible to use and expand logical work, but also draw on experience of actual mathematical practice. I distinguish two important features of the latter: i) the refined conceptual organization internal to a given part of mathematics, and ii) the introduction of new abstract concepts that cut across different areas of mathematics.[57] Logical formality per se does not facilitate the finding of arguments from given assumptions to a particular conclusion. However, strategic considerations can be formulated (for natural deduction calculi) and help to bridge the gap between assumptions and conclusion, suggesting at least a very rough structure of arguments. These logical structures depend solely on the syntactic form of assumptions and conclusion; they provide a seemingly modest, but in fact very important starting-point for strategies that promote automated proof search in mathematics.

Here is a pregnant general statement that appeals primarily to the first feature of mathematical practice mentioned above: *Proofs provide explanations of what they prove by putting their conclusion in a context that shows them to be correct.*[58] The deductive organization of parts of mathematics is the classical methodology for specifying such contexts. "Leading mathematical ideas" have to be found, proofs have to be planned: I take this to be the axiomatic method turned dynamic and local.[59] This requires undoubtedly the introduction of heuristics that reflect a deep understanding of the underlying mathematical subject matter. The broad and operationally significant claim is, that we have succeeded in isolating the leading ideas for a part of mathematics, if that part can be developed by machine — automatically, efficiently, and in a way that is furthermore easily accessible to human mathematicians.[60] This feature can undoubtedly serve as a springboard for the second feature I mentioned earlier, one that is so characteristic of the developments in modern mathematics, beginning in the second half of the 19^{th} century: the introduction of abstract notions that do not have an intended interpretation, but rather are applicable in many different contexts. (Cf. section 5.5.) The above general statement concerning mathematical explanation can now be directly extended to incorporate also the second feature of actual mathematical experience. Turing might ask, whether machines can be educated to make such reflective moves on their own.

It remains a deep challenge to understand better the very nature of reasoning. A marvelous place to start is mathematics; where else do we find such a rich body of systematically and rigorously organized knowledge that is structured for intelligibility and discovery? The appropriate logical framework should undoubtedly include a s*tructure theory of (mathematical) proofs*. Such an extension of mathematical logic and in particular of proof theory interacts directly with a

[57] That is, it seems to me, still far removed from the introduction of "abstract terms" in Gödel's discussions. They are also, if not mainly, concerned with the introduction of new mathematical objects. Cf. note 10.

[58] That is a classical observation; just recall the dual experiences of Hobbes and Newton with the Pythagorean Theorem, when reading Book 1 of Euclid's *Elements*.

[59] Saunders MacLane articulated such a perspective and pursued matters to a certain extent in his Göttingen dissertation. See his papers [1935] and [1979].

[60] To mention one example: in an abstract setting, where representability and derivability conditions, but also instances of the diagonal lemma are taken for granted as axioms, Gödel's proofs can be found fully automatically; see [Sieg and Field]. The leading ideas used to extend the basic logical strategies are very natural; they allow moving between object and meta-theoretic considerations via provability elimination and introduction rules.

sophisticated automated search for humanly intelligible proofs. How far can this be pushed? What kind of broader leading ideas will emerge? What deeper understanding of basic operations of the mind will be gained? — We'll hopefully find out and, thus, uncover with strategic ingenuity part of Turing's residue and capture also part of what Gödel considered as "humanly effective", but not mechanical — "by asking the right questions on the basis of a mechanical procedure".

6.4 (Supra-) Mechanical devices

Turing machines codify directly the most basic operations of a human computor and can be realized as physical devices, up to a point. Gödel took for granted that finite machines just are (computationally equivalent to) Turing machines. Similarly, Church claimed that Turing machines are obtained by natural restrictions from machines occupying a finite space and with working parts of finite size; he viewed the restrictions "of such a nature as obviously to cause no loss of generality". (Cf. section 4.5.) In contrast to Gödel and Church, Gandy did not take this equivalence for granted and certainly not as being supported by Turing's analysis. He characterized machines informally as discrete mechanical devices that can carry out massively parallel operations. Mathematically Gandy machines are discrete dynamical systems satisfying boundedness and locality conditions that are physically motivated; they are provably not more powerful than Turing machines. (Cf. section 5.2.) Clearly one may ask: Are there plausible broader concepts of computations for physical systems? If there are systems that carry out supra-Turing processes they cannot satisfy the physical restrictions motivating the boundedness and locality conditions for Gandy machines. I.e., such systems must violate either the upper bound on signal propagation or the lower bound on the size of distinguishable atomic components.[61]

In *Paper machines*, Mundici and I diagnosed matters concerning physical processes in the following way. Every mathematical model of physical processes comes with at least two problems, "How accurately does the model capture physical reality, and how efficiently can the model be used to make predictions?" What is distinctive about modern developments is the fact that, on the one hand, computer simulations have led to an emphasis on algorithmic aspects of scientific laws and, on the other hand, physical systems are being considered as computational devices that process information much as computers do. It seems, ironically, that the mathematical inquiry into paper machines has led to the point where (effective) mathematical descriptions of nature and (natural) computations for mathematical problems coincide.

[61] For a general and informative discussion concerning "hypercomputation", see Martin Davis's paper [2004]. A specific case of "computations" beyond the Turing limit is presented through Siegelmann's ANNs (artificial neural nets): they perform hypercomputations only if arbitrary reals are admitted as weights. Finally, there is the complex case of quantum computations; if I understand matters correctly, they allow a significant speed-up for example in Shore's algorithm, but the current versions don't go beyond the Turing limit.

How could we have physical processes that allow *supra-Turing computations*? If harnessed in a machine, we would have a genuinely supra-mechanical device. However, we want to be able to *effectively determine* mathematical states from other such states — that "parallel" physical states, i.e., we want to make predictions and do that in a sharply intersubjective way. If that would not be the case, why would we want to call such a physical process a computation and not just an oracle? Wouldn't that undermine the radical intersubjectivity computations were to insure? There are many fascinating open issues concerning mental and physical processes that may or may not have adequate computational models. They are empirical, broadly conceptual, mathematical and, indeed, richly interdisciplinary.

ACKNOWLEDGMENTS

The presentation given here has evolved over almost two decades, and I have drawn systematically on my earlier publications, in particular on [1994], [1996], [1997], [2002a,b] and [2007]. Indeed, parts 5.2 and 5.3 are taken from [2002b]; part 6 was published as [2007]. Pioneering papers from Dedekind, Kronecker and Hilbert through Church, Gödel, Kleene, Post and Turing to Gandy have been a source of continuing inspiration. The historical accounts by Davis, Gandy, Kleene and Rosser have been helpful in clarifying many developments, so has the correspondence between Gödel and Herbrand, as well as that between Bernays and Church. A detailed review of classical arguments for Church's and Turing's theses is found in Kleene's *Introduction to Metamathematics*, in particular, sections 62, 63 and 70; section 6.4 of [Shoenfield, 1967] contains a careful discussion of Church's Thesis. The first chapter of [Odifreddi, 1989] and Cooper's essay [1999] provide a broad perspective for the whole discussion, as does [Soare, 1999]. Much of the material was presented in talks and seminars, and I am grateful to critical responses by the many audiences; much of the work was done in collaboration, and I owe particular debts to John Byrnes, Daniele Mundici, Mark Ravaglia and last, but undoubtedly not least, Guglielmo Tamburrini. Finally, the material was organized for four seminars I gave in November of 2004 at the University of Bologna; I am grateful to Rossella Lupacchini and Giorgio Sandri for their invitation, critical support and warm hospitality.

BIBLIOGRAPHY

[Ackermann, 1925] W. Ackermann. Begründung des "tertium non datur" mittels der Hilbertschen Theorie der Widerspruchsfreiheit. *Mathematische Annalen*, 93: 1–26, 1925.

[Ackermann, 1928] W. Ackermann. Zum Hilbertschen Aufbau der rellen Zahlen. *Mathematische Annalen*, 99: 118–133, 1928.

[Baldwin, 2004] J. Baldwin. Review of [Wolfram, 2002]. *Bulletin of Symbolic Logic*, 10(1): 112–114, 2004.

[Behmann, 1922] H. Behmann. Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem. *Mathematische Annalen*, 86: 163–229, 1922.

[Benacerraf and Putnam, 1983] P. Benacerraf and H. Putnam. *Philosophy of Mathematics - Selected Readings* (Second edition). Cambridge University Press, 1983.

[Bernays, 1918] P. Bernays. *Beiträge zur axiomatischen Behandlung des Logik-Kalküls*. Habilitationsschrift. Göttingen, 1918.

[Bernays, 1922] P. Bernays. Über Hilberts Gedanken zur Grundlegung der Arithmetik. *Jahresbericht der Deutschen Mathematiker Vereinigung*, 31: 10–19, 1922.

[Bernays, 1926] P. Bernays. Axiomatische Untersuchung des Aussagen-Kalkuls der "Principia mathematica". *Mathematische Zeitschrift*, 25: 305–320, 1926.

[Bernays, 1967] P. Bernays. Hilbert, David. In P. Edwards (Editor in Chief), *The Encyclopedia of Philosophy*, vol. 3, pages 496–504, 1967.

[Bernays, 1976] P. Bernays. *Abhandlungen zur Philosophie der Mathematik*. Wissenschaftliche Buchgesellschaft, Darmstadt, 1976.

[Boniface and Schappacher, 2001] J. Boniface and N. Schappacher. Sur le concept de nombre en mathématique - Cours inédit de Leopold Kronecker à Berlin (1891). *Revue d'histoire des mathématiques*, 7: 207–275, 2001.

[Büchi, 1990] J. R. Büchi. *The Collected Works of J. Richard Büchi*. S. Mac Lane and D. Siefkes (eds.), Springer Verlag, 1990.

[Byrnes and W. Sieg, 1996] J. Byrnes and W. Sieg. A graphical presentation of Gandy's parallel machines (Abstract). *Bulletin of Symbolic Logic*, 2: 452–3, 1996.

[Carnap, 1931] R. Carnap. Die logizistische Grundlegung der Mathematik. *Erkenntnis*, 2: 91-105, 1931. (Translation in [Benacerraf and Putnam]).

[Church, 1932] A. Church. A set of postulates for the foundation of logic I. *Annals of Mathematics*, 33(2): 346–366, 1932.

[Church, 1933] A. Church. A set of postulates for the foundation of logic II. *Annals of Mathematics*, 34(2): 839–864, 1933.

[Church, 1934] A. Church. The Richard Paradox. *American Mathematical Monthly*, 41: 356–361, 1934.

[Church, 1935] A. Church. An unsolvable problem of elementary number theory. Preliminary report (abstract). *Bulletin of the American Mathematical Society*, 41: 332–333, 1935.

[Church, 1936] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58: 345–363, 1936. (Reprinted in [Davis, 1965].)

[Church, 1936a] A. Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1(1): 40–41, 1936.

[Church, 1937] A. Church. Review of [Turing, 1936]. *Journal of Symbolic Logic*, 2(1): 42–43, 1937.

[Church, 1937a] A. Church. Review of [Post, 1936]. *Journal of Symbolic Logic*, 2(1): 43, 1937.

[Church and Kleene, 1935] A. Church and S. C. Kleene. Formal definitions in the theory of ordinal numbers. *Bulletin of the American Mathematical Society*, 41: 627, 1935.

[Church and Kleene, 1936] A. Church and S. C. Kleene. Formal definitions in the theory of ordinal numbers. *Fundamenta Mathematicae*, 28: 11–21, 1936.

[Colvin, 1997] S. Colvin. *Intelligent Machinery: Turing's Ideas and Their Relation to the Work of Newell and Simon*. M.S. Thesis, Carnegie Mellon University, Department of Philosophy, 63 pp., 1997.

[Cooper, 1999] S. B. Cooper. Clockwork or Turing Universe — Remarks on causal determinism and computability. In S. B. Cooper and J. K. Truss (eds.), *Models and Computability*, London Mathematical Society, Lecture Note Series 259, Cambridge University Press, pages 63–116, 1999.

[Copeland, 2004] J. Copeland. *The Essential Turing*. Oxford University Press, 2004.

[Crossley, 1975] J. N. Crossley (ed.). *Algebra and Logic; Papers from the 1974 Summer Research Institute of the Australasian Mathematical Society; Monash University*. Springer Lecture Notes in Mathematics 450, 1975.

[Crossley, 1975a] J. N. Crossley. Reminiscences of logicians. In [Crossley, 1975], 1–62, 1975.

[Davis, 1958] M. Davis. *Computability and Unsolvability*. McGraw-Hill, New York, 1958. (A Dover edition was published in 1973 and 1982.)

[Davis, 1965] M. Davis (ed.). *The Undecidable, Basic papers on undecidable propositions, unsolvable problems and computable functions*. Raven Press, Hewlett, New York, 1965.

[Davis, 1973] M. Davis. Hilberts tenth problem is unsolvable. *American Mathematical Monthly*, 80: 233–269, 1973. (Reprinted in the second Dover edition of [Davis, 1958].)

[Davis, 1982] M. Davis. Why Gödel didnt have Church's Thesis. *Information and Control*, 54: 3–24, 1982.

[Davis, 2004] M. Davis. The myth of hypercomputation. In C. Teuscher (ed.), *Alan Turing: Life and Legacy of a Great Thinker*. Springer, 195–211, 2004.

[Dawson, 1986] J. Dawson. A Gödel chronology. In [Gödel, 1986, 37–43].

[Dawson, 1991] J. Dawson. Prelude to recursion theory: the Gödel-Herbrand correspondence. Manuscript, 1991.

[Dawson, 1997] J. Dawson. *Logical Dilemmas*. A. K. Peters, Wellesley, Massachusetts, 1997.

[De Pisapia, 2000] N. De Pisapia. *Gandy Machines: An Abstract Model of Parallel Computation for Turing Machines, the Game of Life, and Artificial Neural Networks*. M.S. Thesis, Carnegie Mellon University, Department of Philosophy, 75 pp., 2000.

[Dedekind, 1888] R. Dedekind. *Was sind und was sollen die Zahlen?* Vieweg, Braunschweig, 1888. (Translation in [Ewald, 1996].)

[Ewald, 1996] W. Ewald (ed.). *From Kant to Hilbert: A Source Book in the Foundations of Mathematics*. Two volumes. Oxford University Press, 1996.

[Feferman, 1988] S. Feferman. Turing in the land of O(z). In [Herken, 1988, 113–147].

[Frege, 1879] G. Frege. *Begriffsschrift*. Verlag Nebert, Halle, 1879.

[Frege, 1893] G. Frege. *Grundgesetze der Arithmetik, begriffsschriftlich abgeleitet*. Jena, 1893. (Translation in [Geach and Black].)

[Frege, 1969] G. Frege. In H. Hermes, F. Kambartel, and F. Kaulbach (eds.), *Nachgelassene Schriften*. Meiner Verlag, Hamburg, 1969.

[Frege, 1984] G. Frege. In B. McGuinness (ed.), *Collected Papers on Mathematics, Logic, and Philosophy*. Oxford University Press, 1984.

[Gandy, 1980] R. Gandy. Churchs Thesis and principles for mechanisms. In J. Barwise, H. J. Keisler, and K. Kunen (eds.), *The Kleene Symposium*. North-Holland Publishing Company, Amsterdam, 123–148, 1980.

[Gandy, 1988] R. Gandy. The confluence of ideas in 1936. In [Herken, 1988, 55–111].

[Geach and Block, 1977] Geach and Block. *Translations from the Philosophical Writings of Gottlob Frege*. Blackwell, Oxford, 1977.

[Gödel, 1931] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38: 173–198, 1931. (Translation in [Davis, 1965], [van Heijenoort, 1967] and *Collected Works I*.)

[Gödel, 1931a] K. Gödel. Diskussion zur Grundlegung der Mathematik. In *Collected Works I*, 200–204, 1931.

[Gödel, 1933] K. Gödel. Zur intuitionistischen Arithmetik und Zahlentheorie. In *Collected Works I*, 286–295, 1933.

[Gödel, 1933a] K. Gödel. The present situation in the foundations of mathematics. In *Collected Works III*, 45–53, 1933.

[Gödel, 1934] K. Gödel. On undecidable propositions of formal mathematical systems. In *Collected Works I*, 346–371, 1934.

[Gödel, 1936] K. Gödel. Über die Länge von Beweisen. In *Collected Works I*, 396–399, 1936.

[Gödel, 193?] K. Gödel. Undecidable Diophantine propositions. In *Collected Works III*, 164–175, 193?.

[Gödel, 1938] K. Gödel. Vortrag bei Zilsel. In *Collected Works III*, 86–113, 1938.

[Gödel, 1946] K. Gödel. Remarks before the Princeton bicentennial conference on problems in mathematics. In *Collected Works II*, 150–153, 1946.

[Gödel, 1951] K. Gödel. Some basic theorems on the foundations of mathematics and their implications. In *Collected Works III*, 304–323, 1951.

[Gödel, 1963] K. Gödel. Postscriptum for [1931]. In *Collected Works I*, 195, 1963.

[Gödel, 1964] K. Gödel. Postscriptum for [1934]. In *Collected Works I*, 369-371, 1964.

[Gödel, 1972] K. Gödel. Some remarks on the undecidability results. In *Collected Works II*, 305–306, 1972.

[Gödel, 1972.1] K. Gödel. The best and most general version of the unprovability of consistency in the same system. First of the three notes [1972].

[Gödel, 1972.2] K. Gödel. Another version of the first undecidability result. Second of the three notes [1972].

[Gödel, 1972.3] K. Gödel. A philosophical error in Turings work. Third of the three notes [1972].

[Gödel, 1974] K. Gödel. Note in [Wang, 1974, 325–6].

[Gödel, 1986] K. Gödel. *Collected Works I*. Oxford University Press, 1986.

[Gödel, 1990] K. Gödel. *Collected Works II.* Oxford University Press, 1990.
[Gödel, 1995] K. Gödel. *Collected Works III.* Oxford University Press, 1995.
[Gödel, 2003] K. Gödel. *Collected Works IV-V.* Oxford University Press, 2003.
[Griffor, 1999] E. R. Griffor (ed.). *Handbook of Computability Theory.* Elsevier, 1999.
[Herbrand, 1928] J. Herbrand. On proof theory. 1928. In [Herbrand, 1971, 29–34].
[Herbrand, 1929] J. Herbrand. On the fundamental problem of mathematics. 1929. In [Herbrand, 1971, 41–43].
[Herbrand, 1930] J. Herbrand. Investigations in proof theory. 1930. In [Herbrand, 1971, 44–202].
[Herbrand, 1931] J. Herbrand. On the fundamental problem of mathematical logic. 1931. In [Herbrand, 1971, 215–259].
[Herbrand, 1931a] J. Herbrand. On the consistency of arithmetic. 1931. In [Herbrand, 1971, 282–298].
[Herbrand, 1971] J. Herbrand. *Logical Writings*. W. Goldfarb (ed.). Harvard University Press, 1971.
[Herken, 1988] R. Herken (ed.). *The Universal Turing Machine — A Half-Century Survey.* Oxford University Press, 1988.
[Herron, 1995] T. Herron. *An Alternative Definition of Pushout Diagrams and their Use in Characterizing K-Graph Machines*. Carnegie Mellon University, 1995.
[Heyting, 1930] A. Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, physikalisch-mathematische Klasse*, 42–56, 1930.
[Heyting, 1930a] A. Heyting. Die formalen Regeln der intuitionistischen Mathematik. Ibid., 57–71, 158-169, 1930.
[Heyting, 1931] A. Heyting. Die intuitionistische Grundlegung der Mathematik. *Erkenntnis*, 2: 106–115, 1931. (Translation in [Benacerraf and Putnam, 1983].)
[Hilbert, 1899] D. Hilbert. *Grundlagen der Geometrie*. Teubner, Leipzig, 1899.
[Hilbert, 1900] D. Hilbert. Über den Zahlbegriff. *Jahresbericht der Deutschen Mathematiker Vereinigung*, 8: 180–194, 1900.
[Hilbert, 1901] D. Hilbert. Mathematische Probleme Vortrag, gehalten auf dem internationalen Mathematiker-Kongreß zu Paris 1900. *Archiv der Mathematik und Physik*, 1: 44-63 and 213–237, 1901. (Partial translation in [Ewald, 1996].)
[Hilbert, 1917*] D. Hilbert. Prinzipien der Mathematik. Lectures given by Hilbert and Bernays in the winter term of the academic year 1917/18. 1917*.
[Hilbert, 1921*] D. Hilbert. Grundlagen der Mathematik. Lectures given by Hilbert and Bernays in the winter term of the academic year 1921/22. 1921*.
[Hilbert, 1922] D. Hilbert. Neubegründung der Mathematik. *Abhandlungen aus dem mathematischen Seminar der Hamburgischen Universität*, 1: 157–177, 1922.
[Hilbert, 1923] D. Hilbert. Die logischen Grundlagen der Mathematik. *Mathematische Annalen*, 88: 151–165, 1923.
[Hilbert, 1926] D. Hilbert. Über das Unendliche. *Mathematische Annalen*, 95: 161–190, 1926.
[Hilbert, 1928] D. Hilbert. Die Grundlagen der Mathematik. *Abhandlungen aus dem mathematischen Seminar der Hamburgischen Universität*, 6: 65–85, 1928.
[Hilbert, 1929] D. Hilbert. Probleme der Grundlegung der Mathematik. *Mathematische Annalen*, 102: 1–9, 1929.
[Hilbert, 1931] D. Hilbert. Die Grundlegung der elementaren Zahlenlehre. *Mathematische Annalen*, 104: 485–494, 1931.
[Hilbert and Ackermann, 1928] D. Hilbert and W. Ackermann. *Grundzüge der theoretischen Logik*. Springer Verlag, Berlin, 1928.
[Hilbert and Bernays, 1934] D. Hilbert and P. Bernays. *Grundlagen der Mathematik I.* Springer Verlag, Berlin, 1934.
[Hilbert and Bernays, 1939] D. Hilbert and P. Bernays. *Grundlagen der Mathematik II.* Springer Verlag, Berlin, 1939.
[Kalmar, 1955] L. Kalmar. Über ein Problem, betreffend die Definition des Begriffes der allgemein-rekursiven Funktion. *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 1: 93–5, 1955.
[Kleene, 1935] S. C. Kleene. General recursive functions of natural numbers. *Bulletin of the American Mathematical Society*, 41: 489, 1935.
[Kleene, 1935a] S. C. Kleene. λ-definability and recursiveness. *Bulletin of the American Mathematical Society*, 41: 490, 1935.

[Kleene, 1936] S. C. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112: 727–742, 1936. (Reprinted in [Davis, 1965].)

[Kleene, 1936a] S. C. Kleene. A note on recursive functions. *Bulletin of the American Mathematical Society*, 42: 544–546, 1936.

[Kleene, 1952] S. C. Kleene. *Introduction to Metamathematics*. Elsevier, Groningen, 1952.

[Kleene, 1981] S. C. Kleene. Origins of recursive function theory. *Annals of the History of Computing*, 3: 52–66, 1981.

[Kleene, 1987] S. C. Kleene. Reflections on Church's Thesis. *Notre Dame Journal of Formal Logic*, 28: 490–498, 1987.

[Kolmogorov and Uspensky, 1963] A. Kolmogorov and V. Uspensky. On the definition of an algorithm. *AMS Translations*, 21(2): 217–245, 1963.

[Kronecker, 1887] L. Kronecker. Über den Zahlbegriff. In *Philosophische Aufsätze*, Eduard Zeller zu seinem fünfzigjährigen Doctorjubiläum gewidmet. Fues, Leipzig, 271–74, 1887. (Translation in [Ewald, 1996].)

[Kronecker, 1891] L. Kronecker. Über den Zahlbegriff in der Mathematik. 1891. In [Boniface and Schappacher, 2001].

[Lamport and Lynch, 1990] L. Lamport and N. Lynch. Distributed Computing: Models and Methods. In J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science*. Elsevier, Groningen, 1990.

[Löwenheim, 1915] L. Löwenheim. Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76: 447–470. (Translation in [van Heijenoort, 1967].)

[MacLane, 1934] S. MacLane. *Abgekürzte Beweise im Logikkalkul*. Inaugural-Dissertation, Göttingen, 1934.

[MacLane, 1935] S. MacLane. A logical analysis of mathematical structure. *The Monist*, 118–130, 1935.

[MacLane, 1979] S. MacLane. A late return to a thesis in logic. In I. Kaplansky (ed.), *Saunders MacLane Selected Papers*. Springer-Verlag, 1979.

[Mancosu, 1999] P. Mancosu. Between Russell and Hilbert: Behmann on the foundations of mathematics. *Bulletin of Symbolic Logic*, 5(3): 303–330, 1999.

[Mancosu, 2003] P. Mancosu. The Russellian influence on Hilbert and his school. *Synthese*, 137: 59–101, 2003.

[Mates, 1986] B. Mates. *The Philosophy of Leibniz*. Oxford University Press, 1986.

[Mendelson, 1990] E. Mendelson. Second thoughts about Church's Thesis and mathematical proofs. *Journal of Philosophy*, 87(5): 225–33, 1990.

[Mundici and Sieg, 1995] D. Mundici and W. Sieg. Paper machines. *Philosophia Mathematica*, 3: 5–30, 1995.

[Odifreddi, 1989] Odifreddi. *Classical Recursion Theory*. North Holland Publishing Company, Amsterdam, 1989.

[Odifreddi, 1990] Odifreddi. *About Logics and Logicians — A Palimpsest of Essays by Georg Kreisel*. Volume II: Mathematics; manuscript, 1990.

[Parsons, 1995] C. D. Parsons. Quine and Gödel on analyticity. In P. Leonardi and M. Santambrogio (eds.), *On Quine*. Cambridge University Press, 1995.

[Post, 1936] E. Post. Finite combinatory processes. Formulation I. *Journal of Symbolic Logic*, 1: 103–5, 1936.

[Post, 1941] E. Post. Absolutely unsolvable problems and relatively undecidable propositions — Account of an anticipation. 1941. (In [Davis, 1965, 340–433].)

[Post, 1943] E. Post. Formal reductions of the general combinatorial decision problem. *Amererican Journal of Mathematics*, 65(2): 197–215, 1943.

[Post, 1947] E. Post. Recursive unsolvability of a problem of Thue. *Journal of Symbolic Logic*, 12: 1–11, 1947.

[Post, 1994] E. Post. *Solvability, Provability, Definability: The Collected Works of Emil L. Post*. M. Davis (ed.). Birkhäuser, 1994.

[Ravaglia, 2003] M. Ravaglia. *Explicating the Finitist Standpoint*. Ph.D. Thesis; Department of Philosophy, Carnegie Mellon University, 2003.

[Rosser, 1935] B. Rosser. A mathematical logic without variables. *Annals of Mathematics*, 36: 127–150, 1935.

[Rosser, 1936] B. Rosser. Extensions of some theorems of Gödel and Church. *Journal of Symbolic Logic*, 1: 87–91, 1936.

[Rosser, 1984] B. Rosser. Highlights of the history of the lambda-calculus. *Annals of the History of Computing*, 6(4): 337–349, 1984.

[Shanker, 1987] S. G. Shanker. Wittgenstein versus Turing on the nature of Church's Thesis. *Notre Dame Journal of Formal Logic*, 28(4): 615–649, 1987.

[Shanker, 1998] S. G. Shanker. *Wittgenstein's Remarks on the Foundations of AI*. Routledge, London and New York, 1998.

[Shapiro, 1983] S. Shapiro. Remarks on the development of computability. *History and Philosophy of Logic*, 4: 203–220, 1983.

[Shapiro, 1994] S. Shapiro. Metamathematics and computability. In I. Grattan-Guinness (ed.), *Encyclopedia of the History and Philosophy of the Mathematical Sciences*. Routledge, London, 644–655, 1994.

[Shapiro, 2006] S. Shapiro. Computability, proof, and open-texture. In A. Olszewski, J. Wolenski, and R. Janusz (eds.), *Church's Thesis after 70 Years*. Logos Verlag, Berlin, 355–390, 2006.

[Shepherdson, 1988] J. Shepherdson. Mechanisms for computing over arbitrary structures. In [Herken, 1988, 581–601].

[Shoenfield, 1967] J. Shoenfield. *Mathematical Logic*. Addison-Wesley, Reading, Massachusetts, 1967.

[Sieg, 1994] W. Sieg. Mechanical procedures and mathematical experience. In A. George (ed.), *Mathematics and Mind*. Oxford University Press, 71–117, 1994.

[Sieg, 1996] W. Sieg. Aspects of mathematical experience. In E. Agazzi and G. Darvas (eds.), *Philosophy of mathematics today*. Kluwer, 195–217, 1996.

[Sieg, 1997] W. Sieg. Step by recursive step: Churchs analysis of effective calculability. *Bulletin of Symbolic Logic*, 3: 154–80, 1997.

[Sieg, 1999] W. Sieg. Hilberts programs: 1917–1922. *Bulletin of Symbolic Logic*, 5(1): 1–44, 1999.

[Sieg, 2002] W. Sieg. Beyond Hilberts Reach? In D. B. Malalment (ed.), *Reading Natural Philosophy*. Open Court, Chicago, 363–405, 2002.

[Sieg, 2002a] W. Sieg. Calculations by man and machine: conceptual analysis. *Lecture Notes in Logic*, 15: 390–409, 2002.

[Sieg, 2002b] W. Sieg. Calculations by man and machine: mathematical presentation. In P. Gärdenfors, J. Wolenski and K. Kijania-Placek (eds.), *In the Scope of Logic, Methodology and Philosophy of Science*, volume one of the 11th International Congress of Logic, Methodology and Philosophy of Science, Cracow, August 1999. Kluwer, Synthese Library volume 315: 247–262, 2002.

[Sieg, 2005] W. Sieg. Only two letters. *Bulletin of Symbolic Logic*, 11(2): 172–184, 2005.

[Sieg, 2006] W. Sieg. Gödel on computability. *Philosophia Mathematica*, 14: 189–207, 2006.

[Sieg, 2007] W. Sieg. On mind and Turing's machines. *Natural Computing*, 6: 187–205, 2007.

[Sieg and Byrnes, 1996] W. Sieg and J. Byrnes. K-graph machines: generalizing Turing's machines and arguments. In P. Hajek (ed.), *Gödel '96*. Lecture Notes in Logic 6, Springer Verlag, 98–119, 1996.

[Sieg and Byrnes, 1999] W. Sieg and J. Byrnes. Gödel, Turing, and K-graph machines. In A. Cantini, E. Casari, P. Minari (eds.), *Logic and Foundations of Mathematics*. Synthese Library 280, Kluwer, 57–66, 1999.

[Sieg and Byrnes, 1999a] W. Sieg and J. Byrnes. An abstract model for parallel computations: Gandy's Thesis. *The Monist*, 82(1): 150–64, 1999.

[Sieg and Field, 2005] W. Sieg and C. Field. Automated search for Gödel's proofs. *Annals of Pure and Applied Logic*, 133: 319–338, 2005.

[Sieg and Parsons, 1995] W. Sieg and C. D. Parsons. Introductory Note to [Gödel, 1938]. In *Gödel's Collected Works III*, 62–85, 1995.

[Sieg and Ravaglia, 2005] W. Sieg and M. Ravaglia. David Hilbert and Paul Bernays, Grundlagen der Mathematik. In I. Grattan-Guinness (ed.), *Landmark Writings in Western Mathematics, 1640-1940*. Elsevier, 981–999, 2005.

[Sieg and Schlimm, 2005] W. Sieg and D. Schlimm. Dedekind's analysis of number: Systems and axioms. *Synthese*, 147: 121–170, 2005.

[Sieg *et al.*, 2002] W. Sieg, R. Sommer, and C. Talcott (eds.). *Reflections on the Foundations of Mathematics — Essays in Honor of Solomon Feferman*. Association for Symbolic Logic, Lecture Notes in Logic 15, 2002.

[Siegelmann, 1997] H. T. Siegelmann. *Neural Networks and Analog Computation — Beyond the Turing Limit*. Birkhäuser, 1997.

[Sinaceur, 2000] H. Sinaceur. Address at the Princeton University Bicentennial Conference on Problems of Mathematics (December 17-19, 1946), by Alfred Tarski. *Bulletin of Symbolic Logic*, 6(1): 1–44, 2000.

[Skolem, 1923] T. Skolem. Begründung der elementaren Arithmetik durch die rekurrierende Denkweise ohne Anwendung scheinbarer Vernderlichen mit unendlichem Ausdehnungsbereich; Skrifter utgit av Videnskapsselskapet i Kristiana, I. Matematisk-naturvidenskabelig klasse, no. 6, 1–38. (Translation in [van Heijenoort, 1967].)

[Skolem, 1923a] T. Skolem. Einige Bemerkungen zur axiomatischen Begründung der Mengenlehre. In *Matematikerkongressen I Helsingfors den 4-7 Juli 1922*. Akademiska Bokhandeln, Helsinki, 217–232, 1923. (Translation in [van Heijenoort, 1967].)

[Spruit and G. Tamburrini, 1991] L. Spruit and G. Tamburrini. Reasoning and computation in Leibniz. *History and Philosophy of Logic*, 12: 1–14, 1991.

[Smullyan, 1961] R. Smullyan. *Theory of formal systems*. Annals of Mathematics Studies 47, Princeton University Press, 1961. (A revised edition was published in 1968.)

[Smullyan, 1993] R. Smullyan. *Recursion theory for metamathematics*. Oxford University Press, 1993.

[Soare, 1996] R. Soare. Computability and recursion. *Bulletin of Symbolic Logic*, 2(3): 284–321, 1996.

[Soare, 1999] R. Soare. The history and concept of computability. In [Griffor, 1999, 3-36].

[Stillwell, 2004] J. Stillwell. Emil Post and his anticipation of Gödel and Turing. *Mathematics Magazine*, 77(1): 3–14, 2004.

[Sudan, 1927] G. Sudan. Sur le nombre transfini ω^ω. *Bulletin mathématique de la Société roumaine des sciences*, 30: 11–30, 1927.

[Tait, 1981] W. W. Tait. Finitism. *Journal of Philosophy*, 78: 524–546, 1981.

[Tait, 2002] W. W. Tait. Remarks on finitism. In [Sieg, Sommer, and Talcott, 2002, 410–419].

[Tamburrini, 1987] G. Tamburrini. *Reflections on Mechanism*. Ph.D. Thesis, Department of Philosophy, Columbia University, New York, 1987.

[Tamburrini, 1997] G. Tamburrini. Mechanistic theories in cognitive science: The import of Turing's Thesis. In M. L. Dalla Chiara, K. Doets, D. Mundici, and J. van Benthem (eds.), *Logic and Scientific Methods*. Synthese Library 259, Kluwer, 239–57, 1997.

[Turing, 1936] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, series 2, 42: 230–265, 1936. (Reprinted in [Davis, 1965].)

[Turing, 1939] A. Turing. Systems of logic based on ordinals. *Proceedings of the London Mathematical Society*, series 2, 45: 161–228, 1939. (Reprinted in [Davis, 1965].)

[Turing, 1947] A. Turing. Lecture to the London Mathematical Society on 20 February 1947. In D. C. Ince (ed.), *Collected Works of A. M. Turing — Mechanical Intelligence*. North Holland, 87–105, 1992.

[Turing, 1948] A. Turing. Intelligent Machinery. 1948. In D. C. Ince (ed.), *Collected Works of A. M. Turing — Mechanical Intelligence*. North Holland, 107–127, 1992.

[Turing, 1950] A. Turing. Computing machinery and intelligence. *Mind*, 59: 433–460, 1950.

[Turing, 1950a] A. Turing. The word problem in semi-groups with cancellation. *Annals of Mathematics*, 52: 491–505, 1950.

[Turing, 1954] A. Turing. Solvable and unsolvable problems. *Science News*, 31: 7–23, 1954.

[Uspensky, 1992] V. A. Uspensky. Kolmogorov and mathematical logic. *Journal of Symbolic Logic*, 57: 385–412, 1992.

[Uspensky and Semenov, 1981] V. A. Uspensky and A. L. Semenov. What are the gains of the theory of algorithms: Basic developments connected with the concept of algorithm and with its application in mathematics. In A. P. Ershov and D. E. Knuth (eds.), *Algorithms in Modern Mathematics and Computer Science*. Lecture Notes in Computer Science, 122: 100–235, 1981.

[van Heijenoort, 1967] J. van Heijenoort (ed.). *From Frege to Gödel - A Source Book in Mathematical Logic, 1879-1931*. Harvard University Press, 1967.

[van Heijenoort, 1985] J. van Heijenoort. *Selected Essays*. Bibliopolis, Naples, 1985.

[van Heijenoort, 1985a] J. van Heijenoort. Jacques Herbrand's work in logic and its historical context. In [van Heijenoort, 1985, 99–122].

[von Neumann, 1927] J. von Neumann. Zur Hilbertschen Beweistheorie. *Mathematische Zeitschrift*, 26: 1–46, 1927.

[von Neumann, 1931] J. von Neumann. Die formalistische Grundlegung der Mathematik. *Erkenntnis*, 2: 116–121, 1931. (Translation in [Benacerraf and Putnam, 1983].)

[Wang, 1974] H. Wang. *From Mathematics to Philosophy*. Routledge & Kegan Paul, London, 1974.

[Wang, 1981] H. Wang. Some facts about Kurt Gödel. *Journal of Symbolic Logic*, 46: 653–659, 1981.

[Whitehead and Russell, 1910] A. N. Whitehead and B. Russell. *Principia Mathematica, vol. 1*. Cambridge University Press, 1910.

[Whitehead and Russell, 1912] A. N. Whitehead and B. Russell. *Principia Mathematica, vol. 2*. Cambridge University Press, 1912.

[Whitehead and Russell, 1913] A. N. Whitehead and B. Russell. *Principia Mathematica, vol. 3*. Cambridge University Press, 1913.

[Wittgenstein, 1980] L. Wittgenstein. *Remarks on the philosophy of psychology, vol. 1*. G. E. M. Anscombe and G. H. van Wright (eds.). Blackwell, Oxford, 1980.

[Wolfram, 2002] S. Wolfram. *A New Kind of Science*. Wolfram Media, Inc., Champaign, 2002.

[Zach, 1999] R. Zach. Completeness before Post: Bernays, Hilbert, and the development of propositional logic. *Bulletin of Symbolic Logic*, 5: 331–366, 1999.

[Zach, 2003] R. Zach. The practice of finitism: Epsilon calculus and consistency proofs in Hilbert's program. *Synthese*, 137: 211–259, 2003.