



## ON TURING'S LEGACY IN MATHEMATICAL LOGIC AND THE FOUNDATIONS OF MATHEMATICS\*

## EL LEGADO DE TURING EN LA LÓGICA MATEMÁTICA Y LOS FUNDAMENTOS DE LAS MATEMÁTICAS

**Joan Bagaria**

ICREA (Institució Catalana De Recerca I Estudis Avançats) and  
Departament de Lògica, Història i Filosofia de la Ciència,  
Universitat de Barcelona.  
[joan.bagaria@icrea.cat](mailto:joan.bagaria@icrea.cat)

**Citation/Cómo citar este artículo:** Bagaria, J. (2013). "On Turing's legacy in mathematical logic and the foundations of mathematics". *Arbor*, 189 (764): a079. doi: <http://dx.doi.org/10.3989/arbor.2013.764n6002>

**Copyright:** © 2013 CSIC. This is an open-access article distributed under the terms of the Creative Commons Attribution-Non Commercial (by-nc) Spain 3.0 License.

Received: 10 July 2013. Accepted: 15 September 2013.

**ABSTRACT:** While Alan Turing is best known for his work on computer science and cryptography, his impact on the general theory of computable functions (recursion theory) and the foundations of mathematics is of equal importance. In this article we give a brief introduction to some of the ideas and problems arising from Turing's work in these areas, such as the analysis of the structure of Turing degrees and the development of ordinal logics.

**RESUMEN:** Alan Turing es conocido sobre todo por sus contribuciones a las ciencias de la computación y a la criptografía, pero el impacto de su trabajo en la teoría general de las funciones computables (teoría de la recursión) y en los fundamentos de la matemática es de igual importancia. En este artículo damos una breve introducción a algunas de las ideas y problemas matemáticos surgidos de la obra de Turing en estas áreas, como el análisis de la estructura de los grados de Turing y el desarrollo de las lógicas ordinales.

**KEYWORDS:** Alan Turing; Foundations of Mathematics.

**PALABRAS CLAVE:** Alan Turing; Fundamentos de la matemática.

\*This is the written version of the talk with the same title delivered at the "International Symposium: The Legacy of Alan Turing", held at the Fundación Ramón Areces, Madrid, on 23-24 October 2012.

## 1. PREAMBLE: THE FOUNDATIONS OF MATHEMATICS IN THE 1930'S

David Hilbert (1862-1943), one of the most prominent mathematicians of his time, developed in the 1920's an ambitious programme for laying the foundations of mathematics<sup>1</sup>. New firm foundations were much needed in the wake of the discovery of several paradoxes involving some of the most basic mathematical notions, such as those of infinite sets, or definability. At the beginning of his famous address on *The Foundations of Mathematics*, delivered at the Hamburg Mathematical Seminar in July 1927, he said (Hilbert, 1928):

I pursue a significant goal, for I should like to eliminate once and for all the questions regarding the foundations of mathematics [...] by turning every mathematical proposition into a formula that can be concretely exhibited and strictly derived. (Hilbert, 1927)

The need for formalization and strict derivation of mathematical statements according to explicitly stated logical rules was the only way, according to Hilbert, to be able to reason about *ideal elements*, such as actual infinite sets, while avoiding the paradoxes. To attain this goal, Hilbert proposed a foundation for mathematics that he called *proof theory*, whose goal was to build a formal system consisting of

- A *formal language*, in which every mathematical statement can be expressed by a formula.
- An *effectively given* list  $T$  of formulas, called *axioms*. These include logical axioms, axioms of equality, and mathematical axioms (including *axioms of number*, i.e., arithmetical axioms, as well as *explicit definitions*, and finite and transfinite *recursion axioms*).
- A finite set of *rules of inference* for deriving new formulas  $\varphi$  from  $T$ , written  $T \vdash \varphi$ , and read " $T$  proves  $\varphi$ ". In Hilbert's system [Hilbert, 1928] the only rule of inference is the *Modus Ponens*.

The formal system is not arbitrary, for it is intended to model mathematics in the following sense (Hilbert, 1928):

The axioms and provable propositions, that is the formulas that result from this procedure, are images (*Abbildung*) of the thoughts constituting customary mathematics as it has developed until now. (Hilbert, 1928)

In a famous passage from his 1925 address *On the Infinite*, Hilbert expressed the conviction of the solvability of all mathematical problems<sup>2</sup> (Hilbert, 1925):

... every mathematical problem can be solved. We are all convinced of that. After all, one of the things that attract us most when we apply ourselves to a mathematical problem is precisely that within us we always hear the call: here is the problem, search for the solution; you can find it by pure thoughts, for in mathematics there is no *ignorabimus*. (Hilbert, 1925)

Thus, it appears that Hilbert was convinced that his proposed system for the foundation of mathematics, or perhaps some extension of it, could be

**Complete:** Given a formula  $\varphi$ , either  $T \vdash \varphi$  or  $T \vdash \neg\varphi$ .

Further, as an essential requirement for the legitimacy of any formal system strong enough for the foundation of mathematics (hence involving ideal elements), Hilbert wanted a proof of consistency, a proof that should be obtained by purely finitary means (i.e., not involving ideal elements), for he believed that only finitary statements are firmly grounded. Thus, the system should be provably

**Consistent:** For no formula  $\varphi$  we have  $T \vdash \varphi$  and  $T \vdash \neg\varphi$ .

And the proof should be finitistic (hence arithmetical). As Hilbert (1925) acknowledged,

... my proof theory cannot specify a general method for solving every mathematical problem; that does not exist. (Hilbert, 1925)

He, however, does not give any argument of why this is so. Yet he seemed to believe that, in addition to being complete and consistent, some formal system based on first-order logic and strong enough to encompass all ordinary mathematics could be

**Decidable:** There is a *definite method, or mechanical process*, by which, given any formula  $\varphi$ , it can be determined whether or not  $\varphi$  is provable in the system.

The existence of such an effective method (for any given formal system) is known as the *Entscheidungsproblem* (the Decision Problem). The Problem had already been posed for different formal systems by Schröder (1895) and Löwenheim (1915). The formulation above was stated in the context of first-order logic by Hilbert and Ackermann in 1928. Let us observe that completeness implies a positive solution to the *Entscheidungsproblem* (assuming the axioms are given effectively).

### 1.1. Incompleteness

In the autumn of 1930, in his retirement address to the Society of German Scientists and Physicians, in Königsberg, and in response to the Latin maxim: *Ignoramus et ignorabimus* (We do not know, and will not know), Hilbert famously asserted:

*Wir müssen wissen. Wir werden wissen.* (We must know. We will know.)

Thus, Hilbert was still holding onto the belief that every mathematical problem could be solved, presumably by deriving it from a complete formal system. Ironically, just the day before, during the Conference on Epistemology held jointly with the Society meetings, Kurt Gödel had informally announced his incompleteness result, which represented a fatal blow to Hilbert's program, at least in the form outlined above. Gödel's First Incompleteness Theorem asserts that

**Theorem 1.** *Every consistent formal system that contains a small amount of arithmetic is incomplete, i.e., there are formulas  $\varphi$  such that neither  $T \vdash \varphi$  nor  $T \vdash \neg\varphi$ .*

The amount of arithmetic needed is indeed a very small fragment of Peano's Arithmetic. For instance, Robinson's finitely-axiomatizable theory  $Q$  suffices.

The second incompleteness theorem is even more dramatic.

**Theorem 2.** *No consistent formal system that contains a moderate amount of arithmetic can prove its own consistency. I.e., the arithmetical statement*

$$\text{CON}(T)$$

*that expresses the consistency of the system is not provable in the system itself. Formally,*

$$T \not\vdash \text{CON}(T).$$

In this case, the finitely-axiomatizable fragment of Peano's Arithmetic known as  $\Sigma_1$ -Induction suffices. Thus, no reasonable formal system for mathematics (not even for a basic fragment of arithmetic) can be both consistent and complete.

But how about the *Entscheidungsproblem*? To answer this, one needs to make precise the notion of *mechanical process*, which leads naturally to the notion of *computable function* on the natural numbers: Given a formal system, one needs a computable function  $f$  such that for any given formula  $\varphi$ ,  $f(\varphi) = 1$  if  $\varphi$  is provable in the system, and  $f(\varphi) = 0$ , otherwise. Since formu-

las in the relevant formal systems are finite sequences of symbols in a countable alphabet, by coding them by natural numbers, we may assume that  $f : \mathbb{N} \rightarrow \{0, 1\}$ . The question is thus to define precisely the notion of computable function on the natural numbers.

### 1.2. Princeton

In the 1930's, Princeton University was the centre of mathematical logic in the United States. The Institute for Advanced Studies (IAS), created in 1930 and housed in the same building as the Department of Mathematics, the old Fine Hall, attracted first-rate mathematicians such as Oswald Veblen and John von Neumann, as well as Albert Einstein. Kurt Gödel visited the IAS on several occasions, giving a series of lectures in 1934 on his incompleteness results, and becoming a permanent member in 1940. In Princeton, Alonzo Church (1903-1995) was the leading figure in Logic. Together with his bright students John Barkley Rosser and Stephen Kleene, Church developed the  $\lambda$ -calculus, a formal system designed to formalize the intuitive notion of *effectively calculable* function. Church and Kleene introduced a class of effectively calculable functions, called  $\lambda$ -definable, and Church formulated the so-called

**Church's Thesis (First unpublished version, 1934):** A function on the natural numbers is effectively calculable if and only if it is  $\lambda$ -definable.

In the meantime, Gödel had introduced his class of computable functions, known as the Gödel-Herbrand *general recursive* functions, and he had presented them during his 1934 visit to Princeton. Shortly afterwards, Kleene showed that the class of  $\lambda$ -definable functions coincides with the class of Herbrand-Gödel recursive functions, and also with the class of Kleene's recursive functions, now known simply as *recursive*. Thus, the first published version of Church's Thesis reads:

**Church's Thesis (1936):** A function on the natural numbers is effectively calculable if and only if it is recursive.

There is a notion of *normal form* for formulas of the  $\lambda$ -calculus, and in 1935, Church had proved the following:

**Theorem 3** (Church, 1936a). *There is no recursive function on the formulas of the  $\lambda$ -calculus such that on any formula  $C$  the value is 2 or 1 according as  $C$  has a normal form or not.*

As a consequence, Church obtains the following:

**Corollary 4** (Church, 1936b). *The answer to the Entscheidungsproblem is negative, even for first-order logic.*

That is, there is no effectively computable function on formulas  $C$  of first-order logic<sup>3</sup> whose value is 1 or 0 according as  $C$  is valid or not (equivalently, by Gödel's Completeness Theorem, provable or not in the first-order logic calculus).

But is this really a solution to the Entscheidungsproblem? Church's solution relies on Church's Thesis, which identifies computable and recursive (as well as  $\lambda$ -definable) functions. Gödel was not convinced. The problem is why should every computable function be recursive?

## 2. ALAN TURING ENTERS THE SCENE

Alan Matthison Turing (1912-1954)<sup>4</sup>, aged 22, was elected Fellow at King's College in Cambridge on the basis of a dissertation on the Gaussian Error Function, in which he proved the Central Limit Theorem (which he didn't know had already been proved in 1922 by J. W. Lindeberg). In the Spring of 1935 Turing took a course by Max Newman on the Foundations of Mathematics, including Gödel's incompleteness theorems and the Entscheidungsproblem. He immediately started working on the problem, producing in April of 1936 a manuscript entitled *On Computable Numbers, with an Application to the Entscheidungsproblem*. The paper consists of:

- The analysis of an (idealized) *human computer*, in the process of computing a computable function on the natural numbers.
- The description of a machine analogue of the human computer: the automatic machines, or a-machines, now known as *Turing Machines*.

As is well-known, a Turing Machine (TM) is a device consisting of a *tape* (unbounded at both ends) divided into squares, a *head* that scans (reads), writes, and erases a symbol (0 or 1) on a given square of the tape, and a finite list of *configurations*. The TM performs two basic operations: (A) a possible change of symbol in the square being scanned, together with a possible change of configuration, and (B) a possible change of scanned square, together with a possible change of configuration. The operation being performed at any given time is determined by the current configuration and the observed symbol in the scanned square. A TM can thus be described completely by a finite *table*, detailing the operations, and therefore the table, hence the TM, can be coded by a natural number.

When set in motion starting from a blank tape, a TM produces a sequence of binary numbers. If the

sequence is infinite, the machine is called *circle-free*. Turing calls an infinite sequence of natural numbers *computable* if it can be computed by a circle-free TM.

- A proof of the existence of a *Universal Turing Machine*.

A Universal Turing Machine (UTM) is a TM that can be used to compute any computable sequence. That is, if the UTM  $\mathcal{U}$  is set in motion with the code of a TM  $\mathcal{M}$  written on the tape, then  $\mathcal{U}$  will produce the same sequence as  $\mathcal{M}$ . Turing's UTM has been regarded as the first description of the stored-program computer.

- A proof of the non-computability of the *Halting Problem*.

The Halting Problem is the problem of determining whether an arbitrary TM is circle-free or not. Turing proves that there is no circle-free TM that will compute the sequence of all codes of circle-free TM's. Should such a TM exist, there would be a TM  $\mathcal{M}$  computing the sequence  $1-f_n(n)$ , where  $f_n$  is the sequence computed by the  $n$ -th circle-free TM. If  $m$  is the code of  $\mathcal{M}$ , then  $f_m(m) = 1 - f_m(m)$ , which is impossible.

Turing also proves that there is no TM that, given the code of a TM  $\mathcal{M}$ , will determine whether the sequence computed by  $\mathcal{M}$  contains a 0 or not. And the same for 1. In fact, he shows that the computability of the Halting Problem is equivalent to the existence of such TM's.

- A "proof" that the *computable* functions (i.e., those computable by a Turing machine) are indeed those that "*would naturally be regarded as computable*".

Of course, a real (i.e., mathematical) proof of this assertion is not possible, and so the arguments given by Turing are of three kinds:

- (a) A direct appeal to intuition.
  - (b) A proof of the equivalence between computable (i.e., computable by a TM) and effectively calculable (or recursive).
  - (c) Giving examples of a large class of functions which are computable.
- A negative solution to the *Entscheidungsproblem*.

For each TM  $\mathcal{M}$ , he writes a first-order formula  $Un(\mathcal{M})$  which says: the symbol 0 appears in the sequence computed by  $\mathcal{M}$ . Then he proves that (1) if 0 appears on the sequence computed by  $\mathcal{M}$ , then  $Un(\mathcal{M})$  is provable, and that (2) if  $Un(\mathcal{M})$  is provable, then 0 appears in the sequence computed by  $\mathcal{M}$ . In this way a negative solution to the Entscheidungs-

sproblem for first-order logic follows from the non-computability of the Halting Problem.

It is hard to overestimate the importance of Turing's paper, both for the foundations of mathematics and the theory of computation. With regards to foundations, it is worth noticing Gödel's enthusiastic reaction (see Gödel, 1964).

Due to A. M. Turing's work, a precise and unquestionably adequate definition of the general concept of formal system can now be given ... Turing's work gives an analysis of the concept of 'mechanical procedure' (alias 'algorithm' or 'computation procedure' or 'finite combinatorial procedure'). ... A formal system can simply be defined to be any mechanical procedure for producing formulas, called provable formulas. (Gödel, 1964)

Thus, while Turing (and Church, independently) had given a negative solution to the Hilbert-Ackermann version of the Entscheidungsproblem, hence yet a further blow to Hilbert's Program, he had given for the first time, according to Gödel, a precise definition of the concept of formal system, thus opening the door to a possible revision of the Program.

### 3. TURING'S DOCTORAL DISSERTATION

On the recommendation of Max Newman, Turing got accepted as a graduate student at Princeton. In a letter to Church, Newman writes:

I should mention that Turing's work is entirely independent; he has been working without any supervision or criticism from anyone. That makes it all the more important that he should come into contact as soon as possible with the leading workers on this line, so that he should not develop into a confirmed solitary.

Turing completed his doctoral dissertation in two years. His thesis, presented in 1938, was entitled *Systems of logic based on ordinals*. The motivation, clearly stated on the first paragraph, was to avoid as far as possible the effects of Gödel's incompleteness through the construction of sequences of formal systems of increasing degree of completeness (Turing, 1939):

The well-known theorem of Gödel [...] shows that every system of logic is in a certain sense incomplete, but at the same time it indicates means whereby from a system  $L$  of logic a more complete system  $L'$  may be obtained. By repeating the process we get a sequence  $L, L_1 = L', L_2 = L'_1, \dots$  each more complete than the pre-

ceding. A logic  $L_\omega$  may then be constructed in which the provable theorems are the totality of theorems provable with the help of the logics  $L, L_1, L_2, \dots$  [...] Proceeding in this way we can associate a system of logic with any constructive ordinal. It may be asked whether a sequence of logics of this kind is complete in the sense that to any problem  $A$  there corresponds an ordinal  $\alpha$  such that  $A$  is solvable by means of the logic  $L_\alpha$ . (Turing, 1939)

The following is an example of the formal systems (ordinal logics) studied in his thesis. Let  $T_0$  be the set of axioms of Peano's Arithmetic (PA). Given  $T_k$ , let  $T_{k+1} := T_k + CON(T_k)$ . Then let  $T_\omega := \bigcup_{k < \omega} T_k$ . One can continue into the transfinite by letting  $T_{\omega+1} := T_\omega + CON(T_\omega)$ , and so on. By Gödel's Second Incompleteness Theorem,  $T_{\alpha+1}$  is strictly stronger than  $T_\alpha$ , whenever  $T_\alpha$  is effectively given. But for  $T_\alpha$  to be effectively given, the ordinal  $\alpha$  must be computable. This leads to the notion of *ordinal notations*, i.e., a computable way to assign countable ordinals to natural numbers. Let us denote by  $|a|$  the ordinal assigned to the natural number  $a$ .

#### The Church-Kleene system $\mathcal{O}$ of ordinal notations:

The number 1 is a notation for the ordinal 0,  $2^e$  is a notation for the successor of  $|a|$  and  $3 \cdot 5^e$  is a notation for the supremum of the sequence  $|a_n|$ , whenever the sequence is strictly increasing and  $e$  is the code of a computable function  $\hat{e}$  such that  $\hat{e}(n) = a_n$ , for each  $n$ .

Observe that the set  $\mathcal{O}$  of ordinal notations, with the order given by  $a <_{\mathcal{O}} b$  if and only if  $|a| < |b|$ , forms a tree that splits only at limit levels, and whenever a branch on the tree splits it does it infinitely.

The system  $T^* := \langle T_\alpha \rangle_{\alpha \in \mathcal{O}}$  is an example of an *Ordinal Logic*. One might hope that every arithmetical statement  $\varphi$  could be decided by some  $T_\alpha$ . That is, either  $T_\alpha \vdash \varphi$  or  $T_\alpha \vdash \neg\varphi$ . Since the statement  $CON(T)$  is  $\Pi_1^0$ , that is, of the form  $\forall x R(x)$ , where  $R$  is some computable relation<sup>5</sup>, a natural question is whether every  $\Pi_1^0$  statement could be decided by some  $T_\alpha$ . Turing proves the following completeness theorem for  $\Pi_1^0$  statements:

**Theorem 5.** For every true  $\Pi_1^0$  sentence  $\varphi$ , there exists  $a \in \mathcal{O}$  with  $|a| = \omega + 1$  such that  $T_a \vdash \varphi$ .

The proof of the Theorem is quite ingenious (see Feferman, 2012), but as Turing himself recognized, it is disappointing because it is uninformative. Given a  $\Pi_1^0$  sentence  $\varphi$ , he does find an  $a$  such that, if  $\varphi$  is true, then  $a \in \mathcal{O}$  with  $|a| = \omega + 1$  and  $T_a \vdash \varphi$ . But this just shifts the question about the truth of  $\varphi$  to the ques-

tion about the number  $a$  belonging to  $\mathcal{O}$ , a question that is actually more complex than any arithmetical statement.

### 3.1. Oracles

In section 3 of the Thesis, Turing goes on to consider what he calls *number-theoretic* statements (i.e.,  $\Pi_2^0$ ), namely, those of the form  $\forall x \exists y R(x, y)$ , where  $R$  is a computable relation. However, he does not succeed in proving a similar completeness theorem for  $\Pi_2^0$  statements; not surprisingly, for S. Feferman showed in 1962 that this is impossible (Feferman, 1962). Then in section 4, and in order to produce concrete examples of problems that are more complex than  $\Pi_2^0$ , Turing introduces the notion of computation relative to an *oracle*.

Let us suppose that we are supplied with some unspecified means of solving number-theoretic [i.e.,  $\Pi_2^0$ ] problems; a kind of oracle as it were. [...] With the help of the oracle we could form a new kind of machine (call them *o*-machines), having as one of its fundamental processes that of solving a given number-theoretic problem. (Turing, 1939)

Turing then defines precisely the notion of *o*-machine (for a fixed oracle  $o$ ) and formulates the *Halting Problem for o-machines*, showing that it is not computable by any *o*-machine, with an argument entirely analogous to the non-computability of the Halting Problem. Since every  $\Pi_2^0$  statement is easily seen to be equivalent to one of the form “ $\mathcal{M}$  is a circle-free TM”, if  $o$  is an oracle that decides every  $\Pi_2^0$  formula, then it follows that the Halting Problem for *o*-machines is not  $\Pi_2^0$ .

Although the notion of *o*-machine is not pursued any further in Turing's Thesis, with hindsight it is perhaps the most important notion introduced in that work, for it is the starting point of the study of relative computability.

### 3.2. Relative computability

The theory of computability relative to an oracle, was developed by Emil Post (1897-1984) and Kleene into the theory of degrees of unsolvability, also known as *Turing Degrees*.

A set of natural numbers  $A$  is *Turing reducible* to a set of natural numbers  $B$  if  $A$  is computable with oracle  $B$ . Written  $A \leq_T B$ .

Two sets of natural numbers  $A$  and  $B$  are *Turing equivalent*, written  $A \equiv_T B$ , if and only if  $A \leq_T B$  and  $B \leq_T A$ .

A *Turing Degree* is an  $\equiv_T$ -equivalence class. Let  $\mathcal{D}$  denote the set of Turing Degrees with the order induced by  $\leq_T$ .

The analysis of the structure of  $\mathcal{D}$  has produced many fascinating new ideas and techniques, such as the *priority method*, and is still one of the central areas of modern computability theory. Much is known about  $\mathcal{D}$ . For example: each degree is countable; there are  $2^{\aleph_0}$ -many degrees; each degree has only countably-many predecessors;  $\mathcal{D}$  is not dense; for every degree  $a$  there is a degree  $b$  incomparable with  $a$ ; there are  $2^{\aleph_0}$ -many incomparable degrees; every two degrees have a least upper bound, but they need not have a greatest lower bound; etc.

The structure of  $\mathcal{D}$  is indeed very complex. For one thing, every countable partial ordering can be embedded in  $\mathcal{D}$ . Moreover, the first-order theory of  $\mathcal{D}$  and the set of theorems of second-order arithmetic are recursively isomorphic (S. Simpson, 1977). Perhaps the most outstanding still open problem about  $\mathcal{D}$  is whether it is rigid, that is, whether  $\mathcal{D}$  has a non-trivial automorphism.

### 3.3. On intuition and ingenuity

In section 11 of the Thesis, entitled “The purpose of ordinal logics”, Turing singles out two faculties of mathematical reasoning he calls *intuition* and *ingenuity*.

The activity of the intuition consists in making spontaneous judgements which are not the result of conscious trains of reasoning. (Turing, 1939)

Whereas

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 validity of the intuitive steps which are required cannot seriously be doubted. (Turing, 1939)

By the introduction of formal logic, the need for intuition is

... greatly reduced by setting down formal rules for carrying out inferences which are always intuitively valid. (Turing, 1939)

But since it is not possible to find a formal logic that would eliminate the use of intuition completely, because of Gödel's incompleteness, one is naturally forced to consider “non-constructive” systems of logic, such as ordinal logics. With an ordinal logic we

are in a position to prove theorems by the intuitive steps of recognizing a number as a notation for an ordinal, and the mechanical (yet requiring ingenuity) steps of applying the logical rules.

Further study of ordinal logics, rechristened as *transfinite recursive progressions of transfinite theories* was carried out in the 1960's by Feferman (1962).

## ACKNOWLEDGMENTS

The work was partially supported by the Spanish Ministry of Science and Innovation under grant MTM2011-25229, and by the *Generalitat de Catalunya* (Catalan Government) under grant 2009 SGR 187.

## NOTES

- 1 In the 1920's, many other people collaborated either jointly with Hilbert or independently in the development of what is now known as *Hilbert's Program*, such as P. Bernays, W. Ackermann, J. Herbrand, or J. von Neumann.
- 2 Address delivered in München on 4 June 1925 at a meeting organized by the Westphalian Mathematical Society to honor Weierstrass.
- 3 A language having only two unary predicate symbols, or just one binary relation symbol suffices.
- 4 We refer to A. Hodges' excellent biography (Hodges, 1983) of A. Turing for all details about his life.
- 5 A unary relation, or predicate, i.e., a set of natural numbers, is computable if and only if its characteristic function is computable.

## REFERENCES

- Church, A. (1936a). "An unsolvable problem of elementary number theory". *Amer. J. of Math*, 58 (2), pp. 345-363.
- Church, A. (1936b). "A Note on the Entscheidungsproblem". *The Journal of Symbolic Logic*, 1 (1), pp. 40-41.
- Feferman, S. (1962). "Transfinite recursive progressions of transfinite theories". *J. Symbolic Logic*, 27, pp. 259-316.
- Feferman, S. (2012). "Turing's Thesis". In *Alan Turing's Systems of Logic. The Princeton Thesis*. Edited by Andrew W. Appel. Princeton University Press.
- Gödel, K. (1964). "Postscriptum". In Davis, M. (ed.), *The Undecidable*. New York: Raven, 1965.
- Hilbert, D. (1925). "On the Infinite", English translation in J. van Heijenoort (Ed.), *From Frege to Gödel. A Source Book in Mathematical Logic, 1879-1931*. Harvard University Press, 1967.
- Hilbert, D. (1928) "Die Grundlagen der Mathematik". *Abhandlungen aus den mathematischen Seminar der Hamburgischen Universität* 6. English translation in J. van Heijenoort (ed.), *From Frege to Gödel. A Source Book in Mathematical Logic, 1879-1931*. Harvard University Press, 1967.
- Hilbert, D. and Ackermann, W. (1928). *Grundzüge der theoretischen Logik*. Springer-Verlag.
- Hodges, A. (1983). *Alan Turing, the Enigma*. New York: Simon and Schuster Inc.
- Turing, A. (1939). "Systems of Logic based on ordinals". *Proc. London Math. Soc.*, (2), pp. 161-228.