Witold Marciszewski
University of Białystok

ON MECHANIZATION OF REASONING, DECIDABILITY OF LOGIC, AND UNCOMPUTABLE NUMBERS

1. Introduction

The concept of mechanization (in Turing's sense) is equivalent as to its scope with that of formalization (in Hilbert's sense), though they may differ in their pragmatic functions[1]. What has Turing done, it was the devising of a mathematical model, stylized as a machine, of the behaviour of a mathematician who acts strictly according to Hilbert's formalistic programme. Thus Turing made it precise what had been aimed at not only by Hilbert, but also Russell, Wittgenstein, and the whole Vienna Circle (and even some catholic writers who dreamed of mechanizing the proofs of God's existence). There are two curious facts about the idea of mechanizability of reasoning.

First, there is the psychological phenomenon that the formalists, now better recognizable under the denomination of computationalists, are so much enthusiastic about the claim that there does not exist any creativity in the world, be it mathematical, technological, philosophical, social, artistic or any other creativity. Thus they are bound to believe that the creation of algorihms, that is, mechanical procedures, or programs, does not require any invention as all. This does not appear to be a special title to pride for reasonable beings, as programmers etc. Moreover, this does not seem to agree with personal experiences of programmers and other mathematicians who happen to be extremely creative minds. The only solution might be as follows: all the insights, though felt to be creative, are in fact due to

[1] This contribution was supported with financial means of the State [Polish] Committee for Scientific Research as research project no. 2 H01A 030 25: *Nierozstrzygalność i algorytmiczna niedostępność w naukach społecznych* (Undecidability and Algorithmic Intractability in Social Sciences), planned for 2003-2006.

ISBN 83-89031-75-2 ISSN 0860-150X

some hidden algorithms in human brains, somehow produced by Nature or Evolution, while the said experience of creativity is a mere illusion.

The second riddle to be considered is as follows. The formalist, or computationalist, claim is maintained in spite of all the limitative results, concerning the limitations of algorithms, that have been demonstrated by Gödel, Tarski, Post, Church, Turing and others. The literature on AI, in which computationalism is defended, is enormous, but the problem can be conveniently restricted to that of decidability of logic which has been solved in the negative by Turing, Church and Post. It is also a consequence of Gödel's result concerning the incompletability of mathematics (this point being lucidly explained by Kneale 1962). Now the isssue is to the effect: how the statement that (A) the first-order logic is not (mechanically) decidable may be compatible with (B) the claim that all correct reasonings can be formalized (mechanized) with the means of that logic? In fact, this is the claim of strong AI.

The only way out which appears is to make explicit that supposedly implicit assumption in B that we are interested only in those correct reasonings which are really *useful* for our either theoretical or practical purposes. Such a restriction ought to be duly substantiated, and the argument could be based on the following Ontological Assumption (of Computationalism).

OAC: **In the empirical world do not appear uncomputable functions, that is, functions taking uncomputable numbers as their values.**

This assumption does not remove the air of strangeness as hinted in the first comment above, but would properly address the second comment, that about sufficiency of algoritmic procedures in dealing with the empirical world. Are they sufficient, then people can be replaced by machines both in cognitive and in practical dealing with the empirical world. In what follows, I am to discuss some points relevant to the OAC principle.

2. The undecidability of logic

2.1. The possibility to mechanize reasoning depends on whether the first-order logic, being the standard instrument of proving, is a decidable theory. And it is decidable if and only if there exists an algorithmic (or effective) method, allowing us to decide about any formula whether it is a law of logic or not. First-order logic is not decidable, and thus there exist formulae about which one cannot recognize (using algorithmic methods) if they are laws of logic. If a reasoning occurs, whose schema follows an undecidable formula, then the computer carrying out the operation will never stop, as stoppong means the solving of the problem in question.

That for every statement of logic there exists a decision procedure, still by Wittgenstein himself (so a great authority in philosophy of logic) was regarded as an unquestionable principle. Here are the solemn expressions in the "Tractatus".

> Our fundamental principle is that every question which can be decided at all by logic can be decided off-hand.... It is possible... to give at the outset a description of all "true" propositions. Hence there can *never* be surprises in logic.... Proof in logic is only a mechanical expedient to facilitate the recognition of tautology, where it is complicated. (*Tractatus Logico-Philosophicus*, 5.551, 6.125, 6.1251, 6.1262).

The same idea of mechanizability of demonstration was shared by Leibniz; as he did not even dream about electric current, he foresaw machines with gears and punched pieces of metal (to conduct or withhold mechanical impulses). The idea of formalization was also close to late scholastics from the circle of nominalism, whom Leibniz referred to in his polemic with antiformalism of the Cartesian school, as when he tried to free young Christian Wolff from under their influence (in which he succeeded). It is more difficult to say to what extent this medieval formalism drew from Aristotle himself, and to what extent it was an original current; if the former is true then the prehistory of the mechanization idea would go back to the beginnings of logic.

This story shows that it was the overcoming of the program of the mechanization of reasoning in the 1930s, and not the program itself, that bore the signs of a scientific revolution.

Overcoming it in the most visible way, that is with the use of the notion of a machine as an essential means of argumentation, was Turing's achievement. It is true that his work would have been impossible without the preceding works by Cantor (the idea of the diagonal demonstration), Skolem (elimination of quantifiers), Hilbert and Gödel and that an equivalent result was reached by Church at the same time, and that Gentzen's deductive methods proved invaluable for further development. However, regarding the role of the idea of the machine, confirmed by the developments in computer science, Turing's name fits best as a symbol of a breakthrough in the history of logic, a breakthrough which took place in 1936. Post, who published a similar analysis in 1936, could also claim this title, but his analysis was rather sketchy, and did not have such a wide continuation as Turing's work. Anyway, even if we do not persist in choosing Turing to become the symbol of that scientific revolution, the appearance in the same year of the three independent proofs of the impossibility of mechanizing reasoning makes that date crucial for the issue of mechanization.

2.2. The issue of the decidability of the classical first-order logic, which I shall shortly call logic, was stated in Hilbert and Ackermann's seminal textbook *Grundzûge der theoretischen Logik* (1928) which belongs to the classics of logic; it is referred to in this text as HA-1928. The issue came to be known under the German name of *Entscheidungsproblem*, in short EP. Paragraph 11 in HA-1928, devoted to that question, ends with the following sentence: "EP must be said to be the main problem of mathematical logic" (Das Entscheidungsproblem muss als das Hauptproblem der mathematischen Logik bezeichnet werden).

The next paragraph starts with the following words: "While in propositional calculus EP was not difficult to solve, finding a general procedure of determination for predicate calculus is a difficult issue, which has not been solved yet" (note how this cautiousness differs from Wittgenstein's naive confidence in algorithmic nature of logic).

That was how it was written in 1928. The solution came a few years later (Gödel 1931, Turing 1936, Church 1936). Against Hilbert's expectations an Wittgenstein's illusory certainty (see quotes above), however, it was negative. An important step towards that solution was the discovery by Gödel (1931) of the incompletability of formal arithmetic, here called in short (when referring to the system studied by Gödel) arithmetic.

In order to realize the link between the undecidability of logic and the incompletability of arithmetic, one must carry out the operation of eliminating functional symbols characteristic of the language of arithmetic, such as the symbol of the consequent, of addition and multiplication, replacing them in a certain way with predicates. Those predicates can be interpreted arithmetically. Owing to such a move, individual variables run through the set of individuals from a given field, while predicative symbols, introduced instead of functional symbols, can be interpreted in any domain whatever. That is how we obtain formulae of predicate logic.

2.3. There is the following relationship between the decidability of logic and the completability of arithmetic. *Had logic been decidable, then for any logical formula the question of its being a law of logic would have been answered through the use of an effective procedure.* Then any true arithmetic proposition, let us say P, could be proved on the basis of the conjunction K of axioms and of the implication $K \Rightarrow P$ being an appropriate law of logic. And that would suffice to demonstrate any given arithmetic truth. However, it is not so (Gödel's result that any arithmetic truth can be demonstrated on the basis of its axioms (the incompletability of the axiomatic system of arithmetic). Thus, logic is not decidable.

The above reasoning assumes that proposition K (the conjunction of axioms) is true, and thus is not inconsistent, for only then does the derivation of P (from K) prove its truth (the assumption of consistency of arithmetic). The clause given in italics in the preceding paragraph also requires a further comment. One should distinguish between the syntactic notion of provability and the semantic notion of logical consequence (a confusion is due to the fact that the ordinary langauge meaning of the word "proof" is not identical with the technical notion which is limited to syntax). What is meant in the said clause is obviously the purely syntactic notion. The lack of proof in this technical sense does not hinder an intuitive grasping of the entailment from K* to P*, which could be called a demonstration in the larger sense, also admitting non-syntactic methods of arriving at truth.

One would reach the same negative result, if there existed evidence for the incompleteness of another theory which uses first-order logic in the course of demonstration and which also deserves the assumption of consistency, as arithmetic does. Thus arithmetic is de iure not distinguished in any way in the discussion regarding the problem of the decidability of logic; however, it de facto serves the role particularly well.

2.5. And that is how we find one of the reasons why not all reasoning can be mechanized. It is a key reason, in the sense that it is sanctioned by a strict metamathematical result (while other reasons, to be mentioned later, are just intuitive ones). Namely, the program for a digital machine, that is a computer, is a translation of a certain algorithm into the language of the machine; first, it is a translation into one of the programming languages, and then this translation is rendered (automatically through a special program) in the internal machine language of the computer in question.

In the case that we are dealing with, the role of the algorithm should be performed by the procedure deciding about the conformity of reasoning with the laws of logic. We enjoy possessing such a procedure only when we can decide about every logical formula used in that reasoning, whether it is a law of logic. That this necessary condition is not fulfilled is the essence of the theorem about the undecidability of logic. And since that condition is not satisfied, the necessary condition of the mechanization of reasoning is not satisfied either.

The term "decidability" has until now been used in a sense which was possible to capture intuitively, but which lacked a closer analysis, to explain the term "procedure" which appears in the intuitive idea of decidability. A precise explanation can be found in several different, but equivalent, theories such as the theory of algorithms, the theory of recursive functions, the lambda calculus and the conception of the Turing machine. The last one is particularly useful for the discussion of our problem. For it not only gives a precise definition of decidability, as the other teories do as well, but it also provides a certain model of mind which helps to state the issue of scope and limits of mechanizability of reasoning.

3. When a machine works without halting

3.1. The notion of undecidability was elaborated by Turing in 1936 with the help of a construction which he called a machine. Church called it Turing machine. This became a technical term, a key concept in both logic and informatics.

Such a machine is like a computer program, that is a set of instructions to ensure the solving of a problem in a finite number of precisely enumerated steps, the solution being achieved through operations on well-definied physical objects, especially symbols made out of a material stuff. However. that is a significant difference for which we rather speak of a machine than of a program or an algorithm.

The difference consists in equipping such an entity with memory, and (as one puts it in cybernetics) with receptors, effectors and internal states. The receptor reads symbols on a moving tape, while the effector writes or effaces the symbols, according to the instructions included in the program. Such instructions take into account both the current state of the machine and the symbol being perceived by the receptor, so that each next move is strictly determined as a function of these two variables.

Such description of the machine allows for a more precise definition of the notion of effective (algorithmic) procedure, which was formerly used in a rather intuitive way. For it is now possible to behaviorally define the process controlled by the program, dividing it into steps visible from the outside (the movement of the tape, the writing of symbols etc.), being elementary components, as if atoms of the proceeding, such that no simpler or more tangible elements might be found.

This method of describing provides us with the following definition of decidability. The problem is solved when the machine stops, that is, when it writes the answer on the tape, and afterwards has nothing else to do. It would seem that the introduction of the term "stop" does not add anything new, for instead of the stopping of the machine, we can speak about its solving the problem. However, it turns out that this way of speaking about the procedure carries a fertile theoretical idea. Namely it guides us in looking for the reason why in some situations the machine does not halt.

One of such situations is the case when the machine is to test the satisfying of a certain condition by an infinite number of objects, particularly natural numbers. The lack of a stop signifies that the machine never reaches a counter-example. Such a statement about the lack of stop would be e.g. Goldbach's hypothesis that every even number is the sum of two prime numbers; if it is true, then the machine checking particular even numbers will never find a counter-example, and thus it will never stop working.

3.2. To exemplify what is said above, let us examine Fermat's big theorem:

$$x^n + y^n = z^n$$

The theorem is to the efect that the above equation has no solutions in the domain of natural numbers, when n is greater than 2, and x, y, z are greater than 0. This is a general statement, in which after the universal quantifiers binding x, y, z, n (with the restriction $n > 2$) there follows the negation symbol, and then the formula mentioned. Is the statement true, then if we substitute consecutive natural numbers, our machine will never find a counter-example (that is, three numbers which with a given n will be the solution of the equation). Thus it will never stop. Let us refer to that machine used for calculating particular substitutes of Fermat's formula as M_k, that is the machine number k in an infinite sequence of numbered machines, starting with the smallest one (e.g. in the sense of having the smallest number of symbols).

However, we cannot know that M_k will never stop if Fermat's statement is true (as it encounters no counter-example), since the machine has an infinite number of elements to examine. The problem could only be solved if there existed a machine that could decide about any machine whatever whether it will ever stop or not. Hilbert hoped that the proof procedure using the laws of logic would become such a machine (he did not use the word "machine" itself, but his idea can perfectly be rendered in this way). Indeed, in some cases the demonstration of a mathematician (a living being or an electronic one) performs such a task. Let us mark such a machine with the symbol H, in honour of Hilbert.

Suppose for a moment that the demonstration of Fermat's big theorem, which was published by Andrew Wiles in 1995, has been performed as if a formalized proof (which is not the case, for in highly complicated proofs the attempt to make them formalized would make them unreadable). Then it is really an application of machine H to a specific case, that is, to proving that M_k will never stop due to the lack of counter-examples, and hence that the great theorem of Fermat is true.

The existence of such a machine H, capable of deciding about every machine whether it will stop or not, is equivalent to the positive solution of the problem of decidability, as expected by Hilbert. However, the solution is in the negative, which in the arithmetical case results from Gödel's theorem of the incompletability of arithmetic, and whichTuring proved in a general way in 1936.

The main idea of Turing's argument, despite all its complexity, can be expressed in the following simple way. One gives the machine H the text of any program as input. The machine has to work out whether the program it was given as input will eventually stop, or go on for ever. If the former, it prints 'Halts', otherwise – 'Doesn't halt'.

Turing proved that program H is impossible to devise. It's a logical impossibility, not that imposed by technology. To prove it, Turing showed that if such a machine existed, it would, when given itself as input, both halt and run forever. Since this is impossible, program H must be impossible too (cp. Paine 2000).

This also concerns the universal procedure postulated by Hilbert which was supposed to prove the lack of stop for any general theorem (as Goldbach's or Fermat's formula), when it is true. For this procedure, involving an infinite set of procedures which it diagnoses, must contain an infinite number of steps. Thus, it cannot ascribe the lack of stop to itself, so it cannot answer the question whether it is generally applicable. Hence there exist undecidable problems.

3.3. This should not be understood in the sense that there are problems doomed never to be solved. Maybe for each problem it is possible to find a program (algorithm) to solve the problem in question. Anyway, there is no such single algorithm applicable to all the problems, while this was the task which Hilbert wished to ascribe to formalized reasoning in first-order logic. Hence there may occur a process of reasoning about which Turing machine could not decide whether it is correct. In this sense such a reasoning cannot be mechanized.

As if at the margin of this discussion, there may arise a psychological problem. If the limitative argument (that limiting the scope of the applicability of formalized, or algorithmic, proof procedures) appears in its essence so simple (though technically complex), then why it has not been intuited by Wittgenstein (compare Section 1) and to others, especially to such a master of logic like Hilbert himself?

An analogy with another master of science, namely Einstein, comes to mind. He shared with all the preceding mankind the belief in the immutability of the universe so firmly that in the face of disagreement between that belief and some points of his own relativity theory, he rejected those points, not daring to depart from the sanctified conviction; only after the discovery of the galaxies' escape by Hubble he has restored the theory to its original content.

A similarly strong belief, this time concerning the solvability of every correctly formulated scientific problem, is frequently visible e.g. in the writings of the 17th and 18th century rationalists, as well as in the statements of 19th century physicists. As far as the question of solvability of mathematical problems, which absorbed Hilbert so much, is concerned, there also existed a specific mental blockage consisting in the identification of truth with derivability. It had its origins in a certain mathematical tradition, and was in Hilbert's times a philosophical dogma in influential circles, such as the Vienna Circle. (The role of Hubble, as mentioned above in Einstein's case, here was played by Gödel and Tarski who have successfully attacked that philosophical dogma.)

This conclusion about insolvability is interpreted by some authors (including Gödel and Post) as an argument for the superiority of the human mind over machines. Turing however, although his contribution to this limitative conclusion was so enormous, did not think the limitations of machines to be considerably larger than those of humans. Man, Turing wrote in an essay in 1950, when he accepts the truth of an arithmetical statement, which cannot be demonstrated by a machine, has the feeling of dominating it. But how to gain the certainty that one is not wrong when one accepts the statement as true? And if there is no ground for such certainty, then with regard to the reasoning ability there is no major difference between the machine and the human mind.

Turing's argumentation could be discussed and questioned on the basis of philosophical assumptions, which would possibly turn out to be more convincing than Turing's philosophy (hidden somewhere behind his arguments). However, it will be more fruitful to think what scientific results could undermine Turing's position. The last part of the essay, referring to Turing's proof of the existence of uncomputable numbers and to its consequences for mechanization of reasoning, is concerned with this issue.

4. Reasonings beyond Turing's barrier

4.1. The discovery of uncomputable numbers is a breakthrough whose import can be compared to the discovery of irrational numbers. It also leads to important consequences for the concept of the mind and of the science; in this respect, it may even surpass the Pythagorean discovery in its weightiness.

But how to discover something, and thus become certain that this something exists when it belongs to a class of objects which are ex definitione incognoscible? It was made possible by George Cantor's method of going beyond what is already known, called diagonal argument.

Here is the application of this argument by Turing (1936). He numbered all the possible machines in a way which was analogous to Gödel's numbering of formulae and proofs. The important difference is that machines are not linguistic objects, but devices used for the calculation of functions. This significant generalization (much appreciated by Gödel when commenting Turing's contribution) was possible as Turing did not employ the notion of proof, which requires a relativisation to axioms and rules, and thus to language; instead, he made use of the notion of calculation procedure, which he defined through the description of the machine's behaviour.

The number of a machine is like its definition, encoding the features of the machine in question. The list of such numbers forms an ordered infinite set of natural numbers. Because it is a denumerable set, no machine is left out the enumeration; this completeness of the list is an important feature in the diagonal argument.

We place that sequence in the first column of the table. In the first row we write the numerical data, which will be transmitted for calculation to every machine; this gives us the same sequence of successive natural numbers as in the first column (when the data form a pair, a tripple etc. of numbers, then the appropriate method of encoding reduces them to one). At the intersection there are the results of the processing (that is, of the calculation) of the given data by the given machine.

Let us now consider all the results located on the diagonal (whence the name of the argument) of our table. Written in one row, they form a certain infinite countable sequence. We then change all the elements of the sequence in a systematic way e.g. by adding one to each of them. A new sequence is formed which differs from all those written in the successive rows of the table. It differs from the sequence in the first row, because there the first position (that is, the first one in the first column) is occupied by a certain number, let's say n, while here it will be $n + 1$. The second position differs from the second number of the second row, the third from the third one etc. We thus have a sequence of numbers, different from all those in the table. Yet we have included in the table all possible calculating machines! Therefore, the new sequence from the diagonal cannot come from any of the machines registered in the table, that is the machines producing countable numbers. So the number represented in the sequence cannot belong to the countable numbers.

In order to use this discovery in the discussion about the possibility to mechanize reasoning, one must introduce the notion of the universal machine. It corresponds to the notion of the computer and is also comparable to the notion of the mind. The machines enumerated on the above-mentioned list are specialized in particular arithmetic operations. They correspond to the formulae recording particular mathematical functions; the simplest one, for instance, can only add one to successive natural numbers, another one can extract roots etc. Of course, machines from this set are also suitable for operations which in their original form do not belong to arithmetic (e.g. syntactic analysis, reasoning), but which are programmed to be represented by arithmetic formulae.

Let us create a machine equipped with a program which is able to imitate any machine from our list. Such an ambitious program is not a phantasy; we have it in every computer through the linking of the operational system and the other elements of the software (translators, applications etc.), which enable us to perform quite a number of tasks on one computer. Such a machine is commonly known as the universal Turing machine.

4.2. After these conceptual preparations we are capable of a more precise statement of the problem put in the title, Now it runs as follows: Is all reasoning which is feasible for the human mind/brain feasible as well for the universal Turing machine?

Before we reached that formulation, we had obtained a negative answer, resulting from the undecidability of logic. But the answer in the context of Turing machine and the computability defined by Turing's method opens new perspectives to the problem, which do not appear in other conceptualizations. However, they are present in the theory of Turing machines, which is suitable as a model of the mind or the brain.

It might not be expected that we shall find the answer at once, whether in informatics, in neurobiology or in physics. However, the very statement of the question is a step forward, as it allows us to make a blueprint of research, in which we would determine which questions must be answered first, before the main issue could be addressed.

A vital question to be included in such research project is the following: do uncomputable numbers appear in nature? More precisely: do certain physical quantities, if they are characterized with absolute accuracy, require uncomputable numbers for their characteristic?

How is this issue related to the question of the mechanization of reasoning? In fact, there is a connection, provided the brain is a physical system. Therefore, if systems characterized by uncomputable numbers are possible in nature, the brain can be claimed to belong to that part of nature (as the mystery of the consciousness phenomenon encourages us to turn to formerly unexplored regions of physics).

The next step in posing questions is the followig. Suppose the brain is a system characterized by some uncomputable numbers, May this result in the possibility of carrying out operations on uncomputable numbers? There is, obviously, a difference between a system which is characterized by certain numbers and a system which performs operations on such numbers. However, it is possible to see a connection, if we consider analog systems. These are devices which map features of some physical phenomena through entering some states which are structurally analogous with those being mapped (that is how the telephone, the phonograph, the photocell, the eye, the ear, etc. work).

Among those operations dealing with uncomputable numbers there may be some reasonings; this is just a conjecture but a serious one, once considered by Turing (1939) himself. There is a relatively little known fact in Turing intellectual quests, commented by Hodges 1997 in a way which he shortly repeated in the following passage (www.turing.org.uk/bio/part3.html).

> The work on 'ordinal logics', probably his most difficult and deepest mathematical work, was an attempt to bring some kind of order to the realm of the uncomputable. This also was connected to the question of the nature of mind, as Turing's interpretation of his ideas suggested that human 'intuition' could correspond to uncomputable steps in an argument.

Obviously, human intuition as mentioned above is that exemplified by asserting an undecidable arithmetical proposition as the result of an informal (i.e. non-algorithmic) reasoning. Hence it is a resoning that cannot be performed by Turing machine.

4.3. However, the very fact that a conclusion cannot be reached by Turing machine does not necessarily entail that this conclusion when obtained by a human being is undoubtedly true. Some authors believe that only algorithms ensure infallibility. According to that view, the subjective feeling that one has certainly reached the truth does not guarantee truthfulness.

Anyway, if the united forces of logicians, computer scientists, physicists and biologists, in a gigantic research project, one day discover structures in the brain which are able to operate on uncomputable numbers, then such processes will prove no more mysterious nor less credible than those which are dealt with by the universal Turing machine, and at the same time they will transgress the limitation of that machine. Among those structures there will probably be a logical reasoning which could not be put in symbols and algorithms, as being crucial in the mechanization process.

Finally, an apology from the author may be necessary for why he is discussing a topic in which still very little is certain, and which requires making one's way among a tangle of hypotheses. The solution, if it is reached, will not depend on philosophical speculation, but rather on the results of particular sciences, whether mathematical or empirical. Wouldn't it be reasonable to refrain from speculation and to wait for those results?

The answer lies in a certain conception of philosophy, which is strongly supported by the success of the ancient atomism. Being once a purely speculative conception, when it was revived in the Renaissance as a philosophical doctrine, it found favourable conditions to inspire and to be tested by physics. It should be observed that stoical philosophy, competitive to atomism, never achieved such mature cooperation with sciences. However, its time seems to be approaching. While the atomists concentrated on what we call hardware, the genius of the stoics anticipated the role of the software. The next wave of philosophy orientated to software came with Leibniz, but it was still too early to use it in the scientific context. Modern times seem to be getting ready for entering that path, although the aim may still be far away.

References and related literature

G. Cantor, Über eine elementare Frage der Mannnigfaltigskeitslehre. *Jahresber. Deutsch. Math. Ver.* 1 (1890/91), 207–246, 282–356.

A. Church, An unsolvable problem of elementary number theory. *Am. J. Math.* 58 (1936), 345–363.

A. Church, A note on the Entscheidungsproblem. *J. of Symbolic Logic* 1 (1936), 40–41, 101–102.

L. Couturat (ed), *Opuscules et fragments inedits de Leibniz.* Paris 1903.

M. Davis (ed), *Solvability, Provability, Definability. The Collected Works of Emil L. Post.* Birkhäuser, Boston etc. 1993.

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

G. Frege, *Begriffsschrift, eine der arithmetischen nachbildete Formelsprache des reinen Denkens.* L. Nebert, Halle 1879.

K. Gödel, Die Vollständigkeit der Axiome des logischen Funktionenkalküls. *Monatshefte für Mathematik und Physik* 37 (1930), 349–360.

K. Gödel, Über formal unentscheibare Sätze der *Principia Mathematica* und verwandter Systeme – I. *Monatshefte für Mathematik und Physik,* 38 (1931), 173–198.

K. Gödel, Über die Länge der Beweisen. *Ergebnisse eines mathematischen Kolloquiums,* Heft 7 [papers read in 1934–35], Franz Deuticke, Leipzig und Wien 1936.

[Gentzen 1934] – G. Gentzen, Untersuchungen über das logische Schliessen, *Mathematische Zeitschrift* 39 (1934), 176-210, 405–431.

R. Herken (ed), *The Universal Turing Machine. A Half-Century Survey.* Oxford Univ. Press, Oxford 1988.

D. Hilbert, Mathematische Probleme. Vortrag, gehalten auf dem internationalen Mathematiker-Kongress zu Paris 1900. *Archiv der Mathematik und Physik,* 3rd series, 1 (1901), 41–63, 213–237.

[Referred to as HA-1928] – D. Hilbert und W. Ackermann, *Grundzüge der theoretischen Logik.* Julius Springer, Berlin 1928. English version *Principles of Mathematical Logic.* Chelsea, New York 1950, ed. by R. E. Luce, is based on 2nd German edition, 1938.

A. Hodges, *Alan Turing: the Enigma*, Walker and Company, New York 2000.

W. Kneale and M. Kneale, *The Development of Logic.* Clarendon Press, Oxford 1962.

L. Kreiser (ed), *Gottlob Frege: Schriften zur Logik. Aus dem Nachlass.* Akademie-Verlag, Berlin 1973.

Leibniz – see Couturat 1903.

W. Marciszewski and R. Murawski, *Mechanization of Reasoning in a Historical Perspective.* Editions Rodopi, Amsterdam 1995.

W. S. McCulloch and W. Pitts, A logical calculus of the ideas immanent in nervous activity. *B. Math. Biophy.* 5 (1943), 115–133.

R. Murawski, Hilbert's Program: Incompleteness theorems vs. partial realizations. In [Woleński (ed) 1994].

Jocelyn Paine, Home Page 2000
www.j-paine.org/students/lectures/lect7/node14.html

R. Penrose, *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics.* Oxford Univ. Press, Oxford etc. 1989.

R. Penrose, *Shadows of the Mind*. Oxford University Press, New York 1994.

R. Penrose, *The Large, the Small and the Human Mind.* Cambridge Univ. Press, Cambridge 1997.

R. Penrose, Beyond the doubting of a shadow: A reply to commentaries on *Shadows of the Mind. Psyche* 2 (23), January 1996. http://psyche.cs.monash.edu.au/v2/psyche-2-23-penrose.html

E. L. Post, Finite combinatory process – Formulation I. *J. of Symbolic Logic* 1 (1936), 103–105. Reprinted in [Davis (ed) 1994].

E. L. Post, Absolutely unsolvable problems and relatively undecidable propositions – account of an anticipation. In [Davis (ed) 1994].

A. N. Whitehead and B. Russell, *Principia Mathematica.* Cambridge Univ. Press, Cambridge 1913.

T. Skolem, Über die Mathematische Logik, *Norsk Matematisk Tidsskrift,* 10 (1928), 125–142. English translation in: [van Heijenoort (ed.) 1967].

A. Tarski, The concept of truth in formalized languages. In [Tarski 1956], 152–278. Polish original 1933.

A. Tarski, On the concept of logical consequence. In [Tarski 1956], 152–278. Polish original 1936.

A. Tarski, *Logic, Semantics, Metamathematics,* Clarendon Press, Oxford 1956.

A. Turing, On computable numbers, with an application to the Entscheidungsproblem. *Proc. of the London Math. Society,* Series 2, 42 (1936), 230–265.

A. Turing, Systems of logic based on ordinals. *Proc. of the London Math. Society,* Series 2, 45 (1939), 161–228.

A. Turing, Intelligent machinery: Raport, National Physics Laboratory. In: B. Meltzer and D. Michie (eds), *Machine Intelligence* 5, Edinburgh Univ. Press, Edinburgh 1969. Also in [Turing 1992].

A. Turing, Computing machinery and intelligence. *Mind* 59 (1950), 433–460.

Collected Works of A. M. Turing. Volume *Mechanical Intelligence.* ed. by D. C. Ince. North-Holland, Amsterdam etc. 1992.

J. van Heijenoort (ed), *From Frege to Gödel. A Source Book in Mathematical Logic, 1879–1931.* Harvard Univ. Press, Cambridge (Mass.) etc. 1967.

J. von Neumann, *The Computer and the Brain,* Yale Univ. Press, New Haven 1958.

A. Wiles, Modular elliptic curves and Fermat's last theorem, *Annals of Mathematics* 142 (1955), 443–551.

L. Wittgenstein, *Tractatus Logico-Philosophicus.* Routledge and Kegan Paul, London 1922.