# Gödelian arguments against AI

Master's Thesis ARTIFICIAL INTELLIGENCE

Department of Information and Computing Sciences
Faculty of Science
Utrecht University

*Author*:
Jesús Rodríguez Pérez

*Supervisor*:
Dr. Gerard Vreeswijk

*Second reviewer*:
Prof. Dr. Albert Visser

2015

# Acknowledgements

I would like to thank my supervisor Dr. Gerard Vreeswijk for his guidance and support in this project; for pointing to multiple suggestions and possible improvements regarding not only the content of this thesis, and in particular technical inaccuracies, but most valuably its form and discursive structure. His personal guidance on writing argumentative essays have proved to be a valuable source.

I am also indebted to Prof. Dr. Albert Visser for proofreading this text and providing useful feedback in the form of sticky notes. I hope this work reflects the advice that in such respect I received from him.

Last, but not least, a word of gratitude goes to my sisters and parents, for the continuous loving support they demonstrated me along the way. Without their everyday words of encouragement this thesis would not have come about in its current shape.

# Contents

# Introduction

The Dartmouth Summer Research Project on Artificial Intelligence was the name of a 1956 undertaking considered the seminal event for, as stated [65] by its promoter John McCarthy, "the science and engineering of making intelligent machines".[1]

Even though 'intelligence' is too vague a concept, the goal statement of Artificial Intelligence (AI) has sometimes been considered more tractable as understood upon the doctrine of "Computationalism", an umbrella term for a myriad of theses such as 'human mind can be explained as a computing machine', 'thinking can be explained as computing'... etc. Most famously, arguments by philosopher-logicians Kurt Gödel and John Lucas based on Gödel's incompleteness theorems for the idea that the thinking of a mathematician in the case of yes-or-no questions cannot be captured by a Turing machine have been used to target the possibility of an artificial intelligence.

This essay shall give a new spin to the debate on the "Gödelian" arguments that permeates the academic literature. In the first part, I shall present Predicate Logic and the formal axiomatic method as a knowledge representation and reasoning framework emerging from a motivation to reflect upon an early idea of an artificial intelligence and at the same time providing the foundations of Gödel's undecidability results; next, I will highlight the role of Recursion Theory in generalizing the incompleteness theorems and particularly of Turing Machines in providing an interface between formal systems and mechanical procedures. An informal exposition of the proof of the limitative results shall close this section.

In the second part, I will first introduce the "Gödelian" arguments and reveal their weaknesses by building upon popular counter-arguments; next; I shall present a novel refutation of the Gödelian objections from the standpoint of mathematical cognition; finally, I will provide an analysis which to the best of my knowledge is missing in the classical discussions of the Gödelian arguments: the distinction between top-down reasoning systems −such as au-

[1] As a matter of trivia, the word 'engineer' stems from the Latin verb *ingeniare*, meaning 'to design or devise' [28], which is, in turn, derived from the noun *ingenium*, meaning 'clever invention' [16]. On the other hand, the *Oxford English Dictionary* [80] remarks that 'engineer' dates back to a posterior time, when an 'engine'er' ("literally, one who operates an engine") referred to 'a constructor of military engines'.

tomated theorem provers and expert systems− and bottom-up reasoning systems −machine learning systems− as a means to prove that even if the Gödelian objections showed that our mental resources for solving certain number-theoretic problems are in principle beyond a Turing Machine, these arguments are useless as a tool for refuting the possibility of an artificial intelligence.

**Part I**

# Predicate logic, axiomatic systems and Gödel's theorems

# 1 Preliminaries

Some scholars [12] [70] have suggested that Gottfried W. Leibniz (1646-1716) should be regarded as one of the first thinkers to envision something like the idea of AI [53].[2] Leibniz "really believe[d] [...] that a precise analysis of the signification of words would tell us more than anything else about the operations of the understanding" [56] and, in order to overcome the ambiguity of natural languages, he conceived an artificial language of symbols representing unique basic universal concepts which could be (re)combined to yield 'higher' concepts, ultimately an "alphabet of human thought". In fact, Leibniz envisioned a method of symbol analysis and comparison whereby these concepts could be used to calculate truths. Thank to his knowledge representation formalism, Leibniz believed, "when there are disputes among persons" we could "simply say: *calculemus*, without further ado, and see who is right" [55].[3]

Leibniz's dreams would be partly materialized by mathematician George Boole, who in 1854 set out [4] as a research program "to investigate the fundamental laws of those operations of the mind by which reasoning is performed; to give expression to them in the symbolic language of a Calculus, and upon this foundation to establish the science of Logic and construct its method". By means of an *algebra* −a combination of numerical values (0,1) standing for ('*true*','*false*') and operators $\{+, \times, \neg, =\}$− Boole incorporated [3] the classical syllogistic reasoning of Aristotle (deductions of the form 'All men are mortal', 'Socrates is a man'. 'Therefore Socrates is mortal') from the realm of Philosophy to the field of Mathematics, laying the foundations of symbolic logic [63].

In 1879, aware of the limitations of Boole's system and in an attempt to realize Leibniz's conceptual analysis for a language of thought, mathematician Gottlob Frege analyzed [29] Boolean propositions into objects (e.g. 'Socrates') and predicates applying to such objects

[2] In fact, as early as 1666, remarking favourably on materialist philosopher Thomas Hobbes' writings, Leibniz wrote: "Hobbes, everywhere a profound examiner of principles, rightly stated that everything done by our mind is a *computation*" [54].

[3] Note that given a fixed alphabet there is an utterly mechanical procedure for generating all possible 'words' or strings of symbols drawn from that alphabet. In the case, for example, of the lower-case English alphabet, one could first list, in dictionary order, all one-letter words, then, in dictionary order, all two-letter words, then, in dictionary order, all three-letter words, and so on and so forth.

(e.g. 'is a man', 'is mortal'). In the line of Leibniz's combinatorial ideas, symbols from a fixed alphabet could be combined according to the rules of a syntax to yield statements or *formulas* $P(o)$, $R(o,o')$, $\cdots$ standing for '$o$ has the property $P$', '$o,o'$ have the relation $R$', $\cdots$[4] which could themselves be combined together as well as with connectives $\wedge$ ('and'), $\vee$ ('or'), $\neg$ ('not'), $\rightarrow$ ('if...then'), $\leftrightarrow$ ('if and only if') to express more complex formulas. In fact, Frege turned his language into a logic by endowing it with a *deductive apparatus*, namely a set of self-evidently true formulas (*axioms*) together with a set of rules (*inference rules*) allowing to derive true formulas (*theorems*) from the axioms and other true formulas. Frege claimed that any theorem of pure logic is true and then attempted to show that the theorems of certain branches of mathematics such as *arithmetic* −i.e. the theory of natural numbers (i.e. whole, non-negative numbers) and their operations $(+,\times)$− could be derived from pure logic alone.[5]

Frege held that the validity of a rule of inference relied on the meaning of the symbols of the theory −its *semantics*− a view which famously critizied the mathematician David Hilbert; Hilbert defended [46, pp. 1123-4] that "[with] [t]he axioms, formulae, and proofs that make up this formal edifice [...] alone [...] contentual thought takes place−i.e. only with them is actual thought practised", and with such an idea of contentless axioms and "mechanical" rules in mind he optimistically challenged in 1900 the scientific community to find a *proof* of (i.e. a sequence of steps leading to the derivation of) the *consistency* of arithmetic, i.e., a proof of the unprovability of formulas of the form $P(o) \wedge \neg P(o)$, known as contradictions.[6,7]

[4] A predicate statement such as '$x$ is mortal' could be written '$x$ is $P$', but it is more convenient to use the notation '$P(x)$'. '$R(x,y)$' could denote, for example, '$x$ is the son of $y$', with the values of $x, y$ ranging over people.

[5] The theorems of a formal system are already latently present in the system's axioms and rules of inference.

[6] In reference to the axioms of Euclid's geometry (e.g. 'To draw a straight line from any point to any point') he once remarked that instead of points and lines and planes one might just as well talk of tables, chairs and beer mugs [42]. Actually, Boole anticipated Hilbert's view; he stated: "They who are acquainted with the present state of the theory of Symbolic Algebra, are aware, that the validity of the processes of analysis does not depend upon the interpretation of the symbols which are employed, but solely upon the laws of their combination" [3, p. 1].

[7] Later on, mathematician Alfred Tarski −who said to represent "this very crude, naive form of anti-Platonism, one thing which [he] could describe as materialism, or nominalism with some materialistic taint" [62, p. 341] would pave the way for a semantics based on formal notions, in particular that of *interpretation*. To illustrate with an example, an interpretation of the formula $P(x) \wedge Q(y)$ would be a structure composed of: (1) a binding or domain of discourse $D$ to which

From the early 20th century on, the mathematician L.E.J. Brouwer understood [6] that methods of proof (and in particular Hilbert's sought-after consistency proof) had to be constructive, i.e., that in order to prove a statement of the form 'there exists an $x$ with the property $P$' one had to construct a particular object $o$ satisfying the property $P$. In turn, Brouwer would go further to reject [7] the validity of the argument $\neg\neg P(x) \rightarrow P(x)$ for $x$ defined on an infinite domain (e.g. the set $\mathbb{N}$ of *all* natural numbers), which was justified in classical mathematics by the law of excluded middle: 'for any $P$, $P(x)$ is true or $\neg P(x)$ is true'.[8] Brouwer's *intuitionism* would not only naturally oppose Frege's *logicism* −whatever the valid logical processes, intuitionists viewed them all as mental constructs, so properly contained in mathematics− but also be in striking contrast with Hilbert's *formalism*; Hilbert remarked [47] that "to prohibit existence statements and the principle of excluded middle is tantamount to relinquishing the science of mathematics altogether".

In the meantime, a "mechanically precise" [77, p. 159] definition of the Fregean semantic sense of a statement following logically from some other statements was ellaborated by philosopher Bertrand Russell and his former professor Alfred N. Whitehead in the three-volume *Principia Mathematica*, where it was shown how to develop the basis of mathematics (e.g. arithmetic) in a logical calculus. Endowed with Russell-Withehead apparatus, Hilbert and his school optimistically pointed out that mathematics was now only a matter of choosing the right axioms and examining the logical consequences of these axioms. In order for a proof of the consistency of the system of Principia Mathematica −a proof that there is no proof of a contradiction in the system− to be formulable, Hilbert had developed "a sort of proof theory which studies operations with the proofs themselves" −a tool which he called 'meta-mathematics'− while, for the purpose of demonstrating the completeness of of the system, mathematicians Wilhelm F. Ackermann and John von Neumann combined the system

its variables $x, y$ are mapped, (2) a meaning attached to $P, Q$ as properties applicable to objects of $D$, and (3) the truth-values assigned to $P(x), Q(y)$ according to whether or not $P, Q$ are satisfied by the objects instantiating $x, y$, respectively. Now, a formula $F$ is *true* or *universally valid* in a certain theory $T$ if and only if any possible interpretation of the language of $T$ satisfies $F$.

[8] This is so because, on the basis that the origins of mathematics are in human intuition and not an external reality, it considered the proposition '$P(x)$ is true' meant '$P(x)$ is proved', and that '$\neg P(x)$ is true' meant '$P(x)$ leads to a contradiction', while we could not legitimately claim that in general (i.e. for arbitrary $P$) we have a proof or disproof of a formula $P(x)$ for $x$ ranging on an infinite domain.

of Principia Mathematica with number-theoretic axioms given by mathematician Giussepe Peano in 1889 to develop the system of *Peano Arithmetic* [63].

## 2 The undecidability results

In 1930, the logician-mathematician Kurt Gödel presented results whereby every axiomatic extension $P$ of Peano Arithmetic that is *sound* −i.e. such that every theorem of $P$ follows logically from the axioms of $P$− 1) contains propositions that are formally *undecidable* −i.e. neither provable nor disprovable− in $P$ and 2) cannot prove the formula stating its own consistency.

After the publication of Gödel's undecidability results for "Principia Mathematica and related systems", their scope of applicability to formal mathematical systems in general (with sufficient amount of arithmetic) still remained unsettled, and was sought for understanding better the consequences for Hilbert's program and considering relative conditions under which it could go through. Since Gödel's general characterization of "formal mathematical system" used the notion of a rule's being constructive, i.e., of there existing a "finite procedure" for carrying out a rule, an exact characterization of what constitutes a 'finite procedure' became a prime requisite for a completely adequate development.

A step in the desired direction would be taken by mathematician Alan Turing in 1936; as an essential means to prove the unsolvability of what Hilbert considered the "fundamental problem of mathematical logic" −that of finding a finite and "mechanical" procedure for deciding, for a given set of axioms, whether an arbitrary mathematical formula follows logically from the axioms− Turing proposed [84] a definition of 'finite and mechanical' which featured a transition from the abstract sense of 'mechanical' intended by Hilbert −'performable without the exercise of thought or volition'[9]− to the concrete sense of 'performable by a machine'.[10,11]

[9] Sense coined by the *Oxford English Dictionary* [32].

[10] It seemed clear to Hilbert that with the solution to this problem, the *Entscheidungsproblem*, it should be possible in principle to settle all mathematical questions in a purely mechanical manner. Hence, by contraposition, given unsolvable problems at all, if Hilbert was right then the *Entscheidungsproblem* itself should be unsolvable [17, p. 108].

[11] Turing's machine was a programmable machine, and by using a simple description of its behaviour and possible states when executing an arbitrary procedure $P$, Turing showed that it was possible to express the termination property $T$ of $P$ −i.e. that $P$ halts after a finite number of steps− by a predicate statement $T(P)$. He then proved that if a machine is made for the purpose of deciding whether a *given* $P$ halts it must for some values of $P$ fail to give an answer. That is, even if there is a procedure for enumerating $P_1$, $P_2$, ... such that $T(P_1)$, $T(P_2)$, ... are provable, there is no

It should perhaps be stressed that what Turing actually demonstrated is quite different from Gödel's result; Gödel had shown that in the system of Peano Arithmetic there exist propositions that are undecidable in the system, while Turing showed that there exist propositions of predicate logic that cannot be classified into the categories of "provable", "disprovable" and "undecidable".

In Gödel's view, Turing's methodology provided a "a precise and unquestionably adequate definition of the general concept of formal system" [40, p. 71], while incidentally paved the way, as we shall next review, to a generalization of Gödel's theorems.

## 2.1 Generalization and acceptance

As the title of Gödel's 1931 paper noted, the incompleteness results applied to "Principia Mathematica and related systems"; more precisely, to the formal system $P$ which is "essentially the system obtained when the logic of PM is superposed upon the Peano axioms" [34, p. 151]; and to the sound systems "that result from $P$ when recursively definable classes of axioms are added" [34, p. 185, note 53]. Now, since Gödel's definition of a "formal mathematical system" included the notion of the class of axioms and relations of immediate consequence being "constructive" −i.e., of there existing, for any string $s$ of symbols drawn from the system's alphabet, finite procedures for deciding (1) whether $s$ is a well-formed statement (i.e. correct according to the rules of a pre-specified syntax); (2) whether $s$ is an axiom; and (3) for each rule of inference, whether $s$ is an immediate consequence (by that rule) of given statements $s_1, ..., s_n$ [36]− it was seeked for generalization purposes whether there exist extensions of Peano Arithmetic whose class of axioms is constructive but not 'recursively definable'.

procedure that for an *arbitrary* $P$ decides whether $T(P)$ is provable. Now, since in 1929 Gödel had demonstrated [33] that predicate logic is sound and complete (*Goedel's Completeness Theorem*) it had been proven impossible to effectively determine whether an arbitrary predicate follows logically from a given set of axioms −i.e. Hilbert's *Entscheidungsproblem* is unsolvable. As Gödel once stated: "For the calculus of propositions [...] [y]ou could construct a machine in [the] 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 it is a logical consequence] and if it is not it would not. But one can prove that it is impossible to construct a machine which would do the same thing for the whole calculus of predicates" [8].

Gödel viewed the requirement of there existing a membership indicator function for the set of formulas and proofs of a formal mathematical system that is "(partial) recursive" −with "the important property that, for each given set of values of the arguments, the value of the function for such values can be computed by a finite procedure" [36, p. 348]− as "a precise condition which in practice suffices as a substitute for the imprecise requirement [...] that the class of axioms and the relation of immediate consequence be constructive". However, it was a necessary step towards generalizing the incompleteness theorems to determine whether such condition was necessary, i.e. whether Gödel's 'recursive functions' encompassed all the functions whose values can be computed by a finite procedure.

Even though it was mathematician Emil Post who earliest approached the 'finiteness problem', the first work that saw the public light in the desired direction was due to mathematician Alonzo Church;[12] in 1935, Church proposed [10] the thesis that the class of recursive functions was equivalent to the class of functions whose values are calculable by an 'effective procedure', where a method is formally said "effective" for a class of problems when it satisfies the following criteria [1]:

- It consists of a finite number of exact, finite instructions.
- When it is applied to a problem from its class:
  - It always finishes (terminates) after a finite number of steps.
  - It always produces a correct answer.
- In principle, it can be done by a human without any aids except writing materials.
- Its instructions need only to be followed rigorously to succeed. In other words, it requires no ingenuity to succeed.

Although Gödel did not use the term "effective" himself, he would often vacillate between the

[12] In order to seek algorithms for solving Hilbert's the decision problem for arbitrary combinatorial calculi, Post proposed to abstract from the kind of rules that occur in quantification theory to obtain a class of rules which included them [18]. While Hilbert and his school went on to approach semantically the decision problem for quantification theory after the publication of *Principia Mathematica*, mathematician Emil Post felt that was not a promising direction because the combinatorial intricacies of predicate logic were too great to penetrate in that manner.

terms "mechanical procedure" and "finite procedure" when referring to formal systems.[13] However, Gödel seemed to consider Church's notion of 'effective calculability' −called "lambda-definability"− as highly inadequate; in a letter to Kleene, Church stated [18, p. 8]:

> «In regard to Gödel and the notions of recursiveness and effective calculability, the history is the following. In discussion with him [...] it developed that there was no good definition of effective calculability. My proposal that lambda-definability be taken as a definition of it he regarded as thoroughly unsatisfactory.»

In 1936 and independently of Church, Alan Turing proposed the thesis that the values of the computable functions corresponded to the numbers which are calculable by a particular device manipulating, according to a program or finite "action table", symbols on a tape divided into squares. Turing concluded "that the 'computable' numbers include all numbers which would naturally be regarded as computable" by "a direct appeal to intuition"; considering a human carrying out a calculation of a number and applying a sequence of simplifications to this process he arrived at the conclusion that, without loss of generality, a "human computer" operates under the restrictive conditions below:

1. "The behavior of the computer at any moment is determined by the symbols which he is observing, and his 'state of mind' at that moment".
2. "[T]here is a bound $B$ to the number of symbols or squares which the computer can observe at one moment".
3. "[T]he number of states of mind which need be taken into account is *finite*".
4. "We may suppose that in a simple operation not more than one symbol is altered".
5. "[E]ach of the new observed squares is within $L_Q$ squares of an immediately previously observed square".

Now, Turing moreover proved that a 'recursive function' is exactly one such that an *algorithm* exists for computing its values, i.e. such that a machine constrained by 1-5 exists that when

[13] The property of being mechanical is spelled out in his 1933 address to the Mathematical Association of America entitled "The Present Situation in the Foundations of Mathematics". There he pointed out that the "outstanding feature of the rules of inference [is] that they [...] 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" [35, p. 45]. See the discussion in [79].

supplied on its tape the digit-sequence $n_1, ..., n_r$ of a number and set to run, after a finite number of steps the machine halts and the value of the function for the arguments $n_1, ..., n_r$ is printed on the tape. Turing's proof of equivalence between recursive functions and specific algorithms seems to have been a major influence in Gödel's view on computability, for by 1938 Gödel seemed to be already reconciled with Church's notion [37, p. 166]:

> «When I first published my paper about undecidable propositions the result could not be pronounced in this generality, because for the notions of mechanical procedure and of formal system no mathematically satisfactory definition had been given at that time. This gap has since been filled by Herbrand, Church and Turing.»

If initially Gödel mentions Turing together with Herbrand and Church, two pages down he refers to Turing's work alone as having demonstrated the correctness of the various equivalent mathematical definitions [37, p. 168]:

> «That this really is the correct definition of mechanical computability was established beyond any doubt by Turing.»

Gödel probably found Turing's notion of 'mechanical procedure' correct "beyond any doubt" because Turing characterized it via a clear set of principles; as Church once remarked of Gödel; "his 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" [18, p. 9]. Although Turing's treatment was not "axiomatic" in any formal sense, he did manage to show that generally accepted properties of effective calculability lead inevitably to a definite class of functions equivalent to that of recursive functions. So appeared to Church, who found the identification of Church's "effective calculability" with 'algorithmic calculability' as "evident immediately" [11, p. 43], drawing what is generally known as the Church-Turing thesis.[14]

Turing's proof of equivalence between the notion of 'recursiveness' and 'algorithmic calculability' would establish that there exist no extensions of Peano Arithmetic (and any other formal system) whose class of theorems is effectively but not recursively enumerable, which would turn Gödel's incompleteness theorems applicable to any formal system containing a

[14]For a guide on the use, misuse and abuse of the term '(Church-)Turing thesis', see [15] and [41].

sufficient amount of number theory. This way, Gödel's first theorem was confirmed to undermine the *logicist* view of Russell and Whitehead that any theorem of mathematics could be formulated and proved within one grand formal system whose axioms were just axioms of logic.

On the other hand, Gödel's second theorem became an unarguable refutation of Hilbert's formalist program; while Hilbert was prepared to allow a variety of formal systems rather than just one −one for arithmetic, geometry, set theory, and so on− and the axioms of these formal systems would not be axioms of logic, but axioms appropriate to the subject-matter in hand, he thought it was possible to prove by finitistic methods that these various formal systems were consistent.[15]

Having refuted two major philosophies of mathematics, may Gödel's theorems also refute the Computationalist thesis? Some thinkers (among them Gödel himself) provided arguments for the affirmative by explaining formal systems as procedures performable by Turing Machines. Obviously, in order to understand their arguments it becomes desirable to have some grasp of the proof of the incompleteness theorems, and so, for those who are unfamiliar with them, I provide a brief informal account in the coming section.

[15]Gerhard Gentzen [31] published a consistency proof of PA but, as Gödel had shown unavoidable, Gentzen's proof used methods that could not be formalized in PA itself. Gentzen's work marks the beginning of post-Gödelian proof theory and work on 'Relativized Hilbert Programs'. Proof theory in the tradition of Gentzen has analyzed axiomatic systems according to what extensions of the finitary standpoint are necessary to prove their consistency [95].

# 3 An informal exposition of the proof

In its general form, Gödel's first incompleteness theorem, or rather a later version of the theorem due to mathematician John B. Rosser [76] −and sometimes known as the Gödel-Rosser theorem− reads as follows:

> For any consistent, recursively enumerable axiomatic extension $T$ of a theory in which a sufficiently large amount of arithmetic can be derived, one can find a formula $G_T$ undecidable in $T$; that is to say, a formula of the language of $T$ such that neither $G_T$ nor $\neg G_T$ is a theorem of $T$.

Before going into the formal proof, let us sketch the intuitive idea that led Gödel to the proof of this famous theorem. Gödel [34, p. 149] began by considering the sentence from which the Liar's Paradox arises, then reasoned that if arithmetic truth *were* a property definable within arithmetic then one could find a non-paradoxical version of the Liar's Statement, yielding a contradiction [22, p. 159]. Now, assuming the soundness of number theory, Gödel realized that if one may construct propositions which make statements about themselves then, since provability in a given theory is a meta-theoretical property, [36, p. 64] "the class $\alpha$ of numbers cannot be expressed by a propositional function of our system, whereas the class $\beta$ of provable formulas can. Hence $\alpha \neq \beta$ and if we assume $\beta \subseteq \alpha$ we have $\beta \subset \alpha$, that is, there is a proposition $A$ which is true but not provable. By the soundness of number theory, $\neg A$ is therefore not true and hence not provable either, i.e., $A$ is undecidable".

Now back to the proof sketch, a few comments on the theorem's statement are in order. First, the condition that $T$ should be consistent means that one cannot derive a contradiction in $T$. Since we are assuming that the underlying logic of $T$ is ordinary classical logic, if $T$ is inconsistent, then it can be shown that any proposition whatever can be proved in $T$ −by the 'law of explosion' or 'principle of *ex falso quodlibet*'. In such case both $G_T$ and $\neg G_T$ would be theorems.

Second, the condition that $T$ is recursively enumerable means that there is an algorithm (non-necessarily terminating) for listing the theorems of $T$ −note that this condition is weaker from that of $T$ being recursive, i.e. of there existing a well-defined function that indicates (non-)membership of a formula to the set of theorems of $T$. Note that by dropping this requirement there would be no guarantee of incompleteness, for one could simply set the axioms of $T$ to be "all truths of number theory".

Third, the weakest $T$ which satisfies Gödel's condition of it containing "a sufficiently large amount of arithmetic" is Robinson Arithmetic (hereafter denoted by $Q$) a theory named after the mathematician who first set it out (see [75]). We know that a mathematical theory or formal system consists of a language −an alphabet or finite set of symbols together with a grammar or set of production rules which allows to recursively form terms from these symbols and ultimately statements or well-formed formulas− together with a proof system or deductive apparatus −a set of inference rules. Now, $Q$'s language, $L_Q$, is a first-order language[16] with the following features:[17]

- A logical alphabet: the symbols for variables, the usual logical connectives and quantifiers, the identity (or equality) symbol, and parentheses.
- A non-logical alphabet: the symbol for the constant 0, the one-place function symbol $S$, and the two-place function symbols '+' and '×'.
- The numerals are $0, S0, SS0, SSS0, ...$, which we may abbreviate $0, 1, 2, 3, ...$, respectively. $\underline{n}$ represents the numeral for the integer value that $n$ stands for. So, if $n$ stands for 5, then $\underline{n}$ represents $SSSSS0$.
- Terms are numerical expressions. The set of terms includes 0 and any variable, as well as expressions built up from those using the functions $S$, '+' and '×'. So, $SS0 \times y$ and $S(x + Sn)$ are examples of terms. Closed terms are variable free.
- Well-formed formulas (wff's) are formed from atomic wff's (formulas with the form $\sigma = \tau$, where $\sigma$ and $\tau$ are terms) using the connectives and quantifiers, as usual.[18]

The intended interpretation $I_Q$ of the non-logical symbols of $L_Q$ is the usual one; namely, '0' has the value zero, '$S$' signifies the successor function, and '+' and '×' are interpreted as addition and multiplication.

$Q$'s proof system is some version of first-order logic with identity/equality. The differences between various presentations of first-order logic of course do not make any difference to what

[16] Unlike second-order languages, a first-order language cannot express quantification over properties/predicates, but only over variables

[17] Extracted from [83]

[18] Note that = is the only predicate in $L_Q$

sentences can be proved in $Q$ given the same sentences as premisses, and so we may assume without loss of generality that we are dealing with a Hilbert-style system (see [48]). In a nutshell, a Hilbert-style system defines a class of logical axioms, usually by giving schemata such as $\Phi \rightarrow (\Psi \rightarrow \Phi)$ and then stipulating −perhaps with some restrictions− that any instance of a schema is an axiom.

Having a rich set of axioms, such a deductive system can operate with just one or two rules of inference (such as $\Phi \rightarrow \Psi, \Phi \vdash \Psi$) and a proof in a theory using is then simply represented by a linear sequence of wff's, each one of which is either (i) a logical axiom, or (ii) an axiom belonging to the specific theory, or (iii) follows from previous wffs in the sequence by one of the rules of inference. [19] $Q$'s axioms are the following:

1. $\forall x \neg(0 = Sx)$
2. $\forall x \forall y(Sx = Sy \rightarrow x = y)$
3. $\forall x(x + 0 = x)$
4. $\forall x \forall y(x + Sy = S(x + y))$
5. $\forall x(x \times 0 = 0)$
6. $\forall x \forall y(x \times Sy = (x \times y) + x)$

Peano Arithmetic extends $Q$'s axioms with every sentence that is the universal closure of an instance of the following Induction Schema: $\{\Phi(0) \wedge \forall x(\Phi(x) \rightarrow \Phi(Sx))\} \rightarrow \forall x \Phi(x)$ where $\Phi(x)$ is an open $L_Q$-wff that has '$x$', and perhaps other variables, free [83].

For arriving at Gödel's theorem, it is sufficient to be able to express unary predicates (i.e. formulas with a single unquantified variable) of numbers, and in fact those are the open wff's with which $L_Q$ represents numerical properties. To begin with, the wff $\neg\exists v(S0 \times v = x)$, $\exists v(SS0 \times v = x)$ represent in $L_Q$ the properties of being prime, even, respectively, where −unlike $v$, a quantified variable− $x$ is a free variable, standing in for a prime, even number, resp. More precisely, the statements "$n$ is prime", "$n$ is even" would be represented, resp., by the wffs $\neg\exists v(S0 \times v = \underline{n})$, $\exists v(SS0 \times v = \underline{n})$, where $\underline{n}$ stands for the numeral of $n$.

[19] An alternative proof system is Natural Deduction, which uses few or no axioms and a rich variety of inference rules. Proofs under this system are not linear, but have a tree structure (see [30])

Note that one may assign the symbols of $Q$'s alphabet unique (natural) numbers in an infinity of ways, and so easily encode wff's as number sequences. Now, the key observation of the proof Gödel elaborated is that it is possible to define in terms of the successor, addition and multiplication operators a one-to-one mapping between these sequences and natural numbers −an example of this mapping is given by the so-called $\beta$-function, well detailed in [83]. That is, each $L_Q$-wff may be effectively encoded into a unique natural number $\ulcorner L_Q \urcorner$ (where $\urcorner$, $\ulcorner$ are symbols of the metalanguage of $L_Q$) called its Gödel number, and each Gödel number may be effectively decoded into a unique $L_Q$-wff.

Now, since we assumed w.l.o.g. that $Q$'s deductive apparatus is a Hilbert-style proof system, $L_Q$-proofs are linear sequences of wff's, thus each proof has an associated Gödel number. This way, checking whether a particular number $n$ is the Gödel number of a proof in $Q$ of a formula with Gödel number $m$ reduces to checking whether $n$ maps to some sequence of number such that (1) the last element is $m$ and (2) every earlier number in the sequence is the number of an axiom, or the number of a formula which follows from earlier formulas in the sequence using one of the rules of inference. Since the rules of inference are themselves numbered, checking whether this last possibility holds reduces to checking whether particular numbers are related in a particular fashion. Therefore, "$y$ is a proof of $x$" is thus representable in $L_Q$ by some wff $\psi(y, x)$. Consider now $\phi(x) = \neg\exists y\psi(y, x)$. One may give a bijection $n \mapsto P(v)$ for $n$ a natural number and $P(x)$ an unary $L_Q$-predicate with free variable $x$. Using this fact, one may find a mapping $F : n \mapsto \phi(\underline{n})$, and represent this substitution operator in $L_Q$ by a mapping $G : n \mapsto \ulcorner F(n) \urcorner$. Now way we can show that for any $n$:

1. if the wff with Gödel number $n$ does not have the property Provable then the wff with Gödel number $G(\underline{n})$ is true in $I_Q$.

2. if the wff with Gödel number $n$ has the property Provable then the wff with Gödel number $G(\underline{n})$ is false in $I_Q$.

Now, using the fact that $\psi(y, x)$ is a recursively-defined $L_Q$-wff, one may show that for any $n$:
3. if the formula with Gödel number $n$ has the property Provable then $Q$ proves the formula with Gödel number $G(\underline{n})$.

On the other hand, by means of a "diagonalization procedure" one may find a $\hat{n}$ such that $\hat{n} = \ulcorner \neg\phi(x) \urcorner$. Now call $\gamma$ to the formula with Gödel number $G(\underline{\hat{n}})$. If we assume $Q$'s soundness, that is, that every theorem of $Q$ should be true in $I_Q$, then we arrive at the undecidability of $\gamma$. Now, by changing the form of $\gamma$ to something a little more complicated,

Rosser(1936) [76] was able later to show that Gödel's theorem could be proved by assuming only consistency.

The situation is now as follows. A theory $T := S + \chi$ with $\chi$ a formula not contradicting any theorem of $S$ is a consistent theory with sufficient amount of arithmetic which is again recursively axiomatizable (for by assumption so is $S$) −in other words, for any consistent axiomatic extension of $S$ an undecidable proposition can be constructed. That is, whatever extension of $S$ must contain an undecidable statement.

I shall conclude the present section with a brief sketch of Gödel's second incompleteness theorem:

> For any consistent axiomatic extension $T$ of a recursively enumerable theory in which a sufficiently large amount of arithmetic can be derived, the consistency of $T$ cannot be proved within $T$ itself.

I will now sketch the proof of the above theorem. Consider the statement that $T$ is consistent; this is equivalent to saying that there do not exist numbers $m$ and $n$, such that $n$ is the Gödel number of a proof of some formula $\psi$ of $T$, while $m$ is the Gödel number of a proof of $\neg\psi$. Now, using the methods employed in the proof of the first incompleteness theorem, we can translate this into a statement of arithmetic, which can then be expressed as a sentence of $T$, say Con($T$). Further let $\phi$ be the undecidable sentence of $T$ which occurs in the proof of the first incompleteness theorem. It follows from the first incompleteness theorem that, if $T$ is consistent, then $\phi$ is not provable in $T$. But

> $\phi \equiv \phi$ is not provable in $T$.

We can thus translate the statement 'if $T$ is consistent, then $\phi$ is not provable in $T$' into the $S$-statement 'if Con($T$) then $\phi$'. Moreover, by going through the full formal proof of the first incompleteness theorem, and translating it into $T$, which we can do since the proof involves only arithmetical notions, we obtain a proof in $S$ of the statement: 'if Con($T$), then $\phi$'.

Now suppose that we can prove the consistency of $T$ within $T$. We can then obtain a proof of Con($T$) within $T$, but we already have a proof of 'if Con($T$), then $\phi$' within $T$. So we obtain a proof of $\phi$ within $T$. However, by the first incompleteness theorem, if $T$ is consistent, then $\phi$ cannot be proved within $T$. Hence if $T$ is consistent then the consistency of $T$ cannot be proved within $T$, which completes the proof.

**Part II**

# The Gödelian arguments and learning systems

# 4 The arguments against Computationalism

Computationalism is an umbrella term for a myriad of theses such as 'human mind can be explained as a computing machine', 'thinking can be explained as computing'... etc. The Computational theory of Mind is based on the absolute consensus reigning in cognitive science that the human brain is, at least in large part, some sort of information-processing device −that mental states (such as 'believing it will rain') can be *fully* described in terms of its relation to input, output and other mental states− and that thinking is recursive, in the sense that a mental state transition depends on past states. Quite generally, thinking is understood as symbol manipulation, with mind's stimuli from the external environmental being explained as a translation of sensations into symbolic code −in terms of the hardware, a mechanical model of the mind is thought of by the Mechanist as having sensors allowing a continuous stream of new information.

If we are to pave the way to rationally determine the truth-value of Computationalism, one may point out, a mathematical idealization of the concepts of minds and machines is required; now, although an idealization is already possible on the machine side −we may, for example, identify 'computer' with Turing Machine, and 'computation' with Turing Machine computation− and it is clear that we may minimally assume that minds are individual entities which can be quantified over along with machines, the lack of a formalism to apply to minds is arguably fatal to any of our pursuits.

On the other hand, proving that what minds can do is not exhaustible by what machines can do suffices in principle to rationally determine that minds cannot be explained as machines without recurring to a complete idealization; this argument is an instance of the indiscernibility of identicals (or rather, its converse, the difference of discernibles). Now, if we now specifically look at what minds and machines can produce as true and exploit the fact that "due to A. M. Turing's work [...] a formal system can simply be defined to be any mechanical procedure for producing [...] provable formulas" [40, pp. 71-72] then we obtain the logical basis of the Gödelian arguments against Mechanism, which I am about to inspect.

## 4.1 Lucas' argument

In a now famous 1961 article [58] philosopher John Randolph Lucas exploited the first incompleteness theorem (which he refers to simply as Gödel's theorem) to argue against Mechanism; more specifically, he used the fact that given the consistency of a formal system $S$ which contains a sufficient amount of arithmetic, its Gödel formula may be interpreted as being unprovable in $S$ *though true* −as seen in the previous chapter, the statement says that it is unprovable, and so if it were provable it would make $S$ inconsistent. Strangely enough, all Gödel concluded in his 1931 work was "$[P_k(x)]$ is therefore undecidable on the basis of $k$", and not '$[P_k(x)]$ is therefore true and undecidable on the basis of $k$'.[20]: The passage that summarizes Lucas' argument is the following:

> «Gödel's theorem must apply to cybernetical machines, because it is of the essence of being a machine, that it should be a concrete instantiation of a formal system. It follows that given any machine which is consistent and capable of doing simple arithmetic, there is a formula which is incapable of being produced as true −i.e. the formula is unprovable-in-the-system− but which we can see to be true. It follows that no machine can be a complete or adequate model of the mind, that minds are essentially different from machines.»

Let us now scrutinize the premises of the argument. Firstly, Lucas rests upon the assumption that the statements which can be produced as true by a "cybernetical" −by which he means 'programmable'− machine correspond to the theorems of some formal axiomatic system.[21]

[20] Goedel himself would later admit that he avoided the "highly transfinite concept of 'objective mathematical truth' as opposed to that of 'demonstrability'" because he believed that "formalists consider[ed] formal demonstrability to be an *analysis* of the concept of mathematical truth, and, therefore, were of course not in a position to *distinguish* the two" [91]. Note that, in any case, what may be derived from Gödel's first incompleteness theorem is not that the extension of the predicate 'true' is larger than the extension of the predicate 'recognizable as true' −which would be unacceptable from an intuitionist or anti-realist viewpoint− but that the extension of the predicate 'recognizable as true' exceeds the extension of the predicate 'provable in $S$', which is quite a different matter.

[21] The word 'cybernetical' stems from 'cybernetics', which denotes an academic field formalized by Norbert Wiener that focuses on the study of those systems where action by the system generates some change in its environment that triggers a new system change (feedback) [94].

Is this obvious? According to Lucas, the only other option is an "arbitrary and irrational" system, so that "however a machine is designed, it must proceed either at random or according to definite rules".

Such view of a reasoning system is static, in the sense that all of its knowledge is in the axioms and rules. On the contrary, a system's knowledge may be dynamic and so not axiomatic in the usual sense. It may not arbitrary, however, because the knowledge and the inference rules may have justifications. Examples of this type of system include nonmonotonic reasoning systems (see [64]). The distinction made here between static and dynamic systems is, I think, the same as that which Cellucci [9, p. 206] draws between closed and open systems by characterizing closed systems as those whose rules 'are given once for all and allow us to deal only with knowledge not changing in time', and open systems as those whose 'rules can change at any stage and allow us to deal with knowledge changing in time'.

According to Lucas, the axiomatic representability actually applies to "physical deterministic systems, which are sufficiently rich to contain an analogue to simple arithmetic" [60]. At the hardware level, $M$ may be thought of as embodying a behaviour function of the form $M(s_t) = s_{t+1}$, where $s_t$, $s_{t+1}$ correspond to the current and next computational states, respectively −starting from some initial state $s_0$, the computational state is modified by the application of $M$ to generate a next-state. Now, how could we represent these states by a formal system? The obvious way to proceed would be to take a description of the initial state of $M$ as an axiom and to devise rules of inference corresponding to $M$'s program so that, if $\chi$ was inferred from $\psi$, then $\chi$ would be a description of the state of $M$ which immediately followed the state described by $\psi$ according to $M$'s program. In this formal system, say $T_M$, the theorems would describe states reachable by $M$. Now, in this case $T_M$'s axioms do not necessarily correspond to $M$'s beliefs, for our interpretation of these descriptions may have nothing to do with their interpretation in terms of $M$'s knowledge system.

In a later writing, Lucas said that "the operations of any such computer could be represented in terms of a formal logistic calculus with a definite finite number (though enormously large) of possible well-formed formulae and a definite finite number (though presumably smaller) of axioms and rules of inference. The Gödelian formula of such a system would be one that the computer, together with its software, would be unable to prove." [61]. Again, such a system may not be axiomatic in the usual sense, so his conclusion does not hold.

In all the conceptual confusions mentioned above, a central issue is the notion of "level": "But although there is a difference of levels, it does not invalidate the argument. A compiler is

entirely deterministic. Any sequence of operations specified in machine code can be uniquely specified in the programming language, and vice versa" [61]. Indeed, even if $M$ is a "non-deterministic" machine in the sense that for certain $s_t$ one allows the choice of $s_{t+1}$ to be made randomly, the machine's range of allowable "next-states" is still fixed; therefore, if we assume that the $s_t$'s and $s_{t+1}$'s can be specified by strings of symbols then it is not hard to see that $M$'s behaviour can be coded up in some formal system $T_M$ that has a variety of strings of the form "$M(s_t) = s_{t+1}$" as its theorems. What Lucas fails to see, however, is that when a system can be analysed in multiple levels, even when the levels are deterministically related to one another, different levels may still have very different properties, and a conclusion may be correct in a certain level, but not in the level above or below it. For example, we cannot say that a computer cannot process decimal numbers or symbols (because they are coded by bits), or cannot use functional or logical programming language (because they are eventually translated into procedural machine language).

We cannot but assume that the interpretation under which $T_M$ is an axiomatic system whose theorems are the truths producible by $M$ is that of $M$ implementing a model of human knowledge about mathematics. Now one must notice that Lucas assumes that the corresponding $T_M$ must be consistent. This is the assumption which has most famously been objected, the earliest argument coming from philosopher Hilary Putnam [72, p. 366] in his comments on E. Nagel and J.R. Newman (1958) [68]:

> «To be able to assert that the mind surpasses any machine, the human mind $H$ has to be capable of discovering a proposition $U$ which no Turing machine $M$ [an algorithm for producing mathematical theorems] can prove. However, what $H$ can discover is a proposition $U$ such that $H$ can demonstrate:
>
> (') If $M$ is consistent, then $U$ is undecidable (though true).
>
> It is true that $M$ cannot prove $U$, but neither can $H$ prove $U$, unless $H$ can prove that $M$ is consistent. But $H$ cannot prove the consistency of $M$, if $M$ is very complicated. Moreover, $M$ can perfectly well prove (').»

One may endorse Putnam's criticism in a very simple manner. Suppose that we have been given in full detail a complex "formal" system that models the mathematical abilities of a human mind. It is clear that the mind can construct the Gödelian sentence for this system. However, what it can prove is a conditional: that the proposition is true provided the logical system is consistent −indeed, this much is provable inside the very logical system in question,

as seen in the earlier section.

Lucas disregards that it remains possible that "$H$ cannot prove the consistency of $M$, if $M$ is very complicated" [72]. For some relatively simple formal systems of arithmetic, such as Peano Arithmetic, we may know with mathematical certainty, even though this is not provable within the system, that its primitive deductive basis (the axioms and primitive rules of inference) does not generate any contradiction. In these cases, there may be a sense in which it is true that the human mind relevantly "sees" the truth expressed by the Gödelian sentence, since this provably follows from the system's consistency. But there are other formal systems for mathematics with respect to which the system's consistency is anything but obvious.

If the above brings to light an issue of computational complexity, a famous criticism of Lucas' argument due to scholar Paul Benacerraf points to a challenge of effective computability [2]:

> «Lucas often relies on what he takes to be his ability to show of any formal system adequate for arithmetic that its Gödel sentence is true. But he fails to notice that it depends very much on how he is given that system. If given a black box and told not to peek inside, then what reason is there to suppose that Lucas or I can determine its program by watching its output? But I must be able to determine its program (if this makes sense) if I am to carry out Gödel's argument in connection with it. [...] If the machine is not designated in such a way that there is an effective procedure for recovering the machine's program form the designation, one may well know that one is presented with a machine but yet be unable to do anything about finding the Gödel sentence for it.»

Benacerraf's point is that if one is given $M$ as an opaque theorem-listing machine −say, printing out a formula whenever suitable action such as pressing a button is performed− then one might never be able to decide the axiomatic implementation of the system, and so legitimately claim that it is consistent. His conclusion therefore is that the most one could infer from Lucas's argument is the disjunction '*either* no formal system encodes all human arithmetical capacity *or* any system which does has no axiomatic specification which human beings can comprehend' Note that Benacerraf's criticism is conceptually prior to Putnam's, for Putnam questions the provability of the consistency of $T_M$ after taking for granted the decidability of $T_M$.

To this objection, Lucas vaguely argues [58] that if we take $h$ to be the integer that codes the program for $M$, (1) after hanging around with $M$ for a while $H$ will feel like asserting

$\mathrm{Tr}(h)$, a sentence which says something like "If I base a machine on the algorithm coded by $h$ I'll get a machine that only produces true sentences about mathematics", and that (2) having perceived the truth of $\mathrm{Tr}(h)$, $H$ will also feel like asserting $\mathrm{Con}(h)$, a sentence which says something like, "If I base a machine on the algorithm coded by $h$ I'll get a machine that does not generate any mathematical contradictions". However, if after $n$ stages $M$ has not outputted the negation of a previously outputted formula $F$, i.e. $\neg F$, it might be because $\neg F$ is not a theorem of $T_M$ or because it shall be outputted at the stage $n+1$.

Lucas argues that in such case $M$ would not qualify for a model of human's mathematical knowledge, for whenever we notice our reasoning is inconsistent we retract one of the halves of the contradiction. However, if we are to give fair play to $M$ then we must allow it to retract some belief once it notices their inconsistency. Granted that, there would be no reason to suppose that the theorems of $T_M$ are recursively enumerable; for even if we could recognize that a statement is granted of retracted effectively when it occurs, there is no effective method to decide whether a given statement will ever be retracted. In the technical jargon, the set of ever-retracted statements is not recursive.

This is a convenient moment to compare Lucas's views on the matter with those expressed by Gödel himself. Gödel's anti-mechanist argument seems to me definitely distinct from Lucas's and in a sense more rigorous, since, as we shall see, is not liable, for example, to Putnam's and Benacerraf's objections.

## 4.2 Goedel's argument

Gödel had a series of meetings with the chronicler of his philosophical thought, Hao Wang, starting in October 1971, in which they discussed a number of issues, including the question of minds and machines.[22] The following excerpt from Wang summarizes Gödel's case for a relevant limitative role of the incompleteness theorem in the mind-machine issue:

[22]"In the summer of 1971, Gödel agreed to hold regular sessions with me [i.e. Hao Wang] to discuss a manuscript of mine, which was subsequently published in 1974 as *From Mathematics to Philosophy*" [92, p. 193]

> «My incompleteness theorem makes it likely that mind is not mechanical [...] [o]n the other hand, on the basis of what has been proved so far, it remains possible that there may exist (an even be empirically discoverable) a theorem-proving machine which in fact is equivalent to the human mind, but cannot be proved to be so, nor even be proved to yield only correct theorems of finitary number theory.»[23]

Gödel derived this as an inevitable conclusion of what he called the incompletability of mathematics:

> «The human mind is incapable of formulating (or mechanizing) all its mathematical intuitions. I.e. : if it has succeeded in formulating some of them, this very fact yields new intuitive knowledge, e.g. the consistency of this formalism. This fact may be called the incompletability of mathematics.»

The source of the incompletability of mathematics was publicly first addressed by Gödel in his 1951 lecture "Some basic theorems on the foundations of mathematics and their implications" [38]. There he started by reformulating the first incompleteness theorem into a result about Diophantine problems −problems of deciding, for arithmetic expressions of the form $P(x_1, ..., x_n) = 0$ (equations) where $P$ is some polynomial over the natural numbers, the domain $D$ of $x_1, ..., x_n$ such that the expression holds.[24]

Subsequently, Gödel expressed the $\omega$-consistency of a formal mathematical system as the stronger condition that the system proves no *false* formulas of the type of the (negation of the) Gödel statement −i.e., that it is *sound* at least in the case of such statements− to produce the first incompleteness theorem: "whatever well-defined system of axioms and rules of inference may be chosen, there always exist diophantine problems of the type described

[23] Actually, Wang [91] replaced "the human mind" by 'mathematical intuition'. However, this is misleading, because the latter is imbued with some kind of philosophical standpoint I claim Gödel did not intend in this context. By mathematical intuition one usually understands something like a (sensorial) perception, if there were one, of the object of some mathematical theory. As for the usage of the term "human mind" Wang [92, p. 189] comments as follows: "There is a terminological complication in Gödel's use of the terms *human mind* and *mathematical intuition* I tend to think in terms of the collective experience of the human species, and so I asked him once about his usage, His reply suggests to me a simplifying idealization. '6.1.23 By mind I mean an individual mind of unlimited life span'". However, no mention is made here to the term "mathematical intuition".

[24] Hilbert asked in 1900 for a "mechanical" procedure for solving an arbitrary Diophantine problem.

which are undecidable by these axioms and rules, provided only that no false propositions of this type are derivable" [91, p. 308].

From that, Gödel's second theorem followed: "for any well-defined system of axioms and rules, in particular the proposition stating their consistency (or rather the equivalent number-theoretical proposition) is unprovable from these axioms and rules, provided these axioms and rules are consistent and suffice to derive a certain portion of finitistic arithmetic of integers" [91, pp. 308-9].

Gödel would then go on to present his distinction between "objective mathematics" (the system of all true mathematical propositions) and "subjective mathematics" (the system of all demonstrable mathematical propositions). As far as objective mathematics is concerned, it follows from Gödel's Second Incompleteness Theorem that no well-defined system $S$ of correct axioms can contain all of objective mathematics, because $\mathrm{Con}(S)$ cannot be proved in $S$, though is true [91, p. 309]:

> «For it impossible that someone should set up a certain well-defined system of axioms and rules and consistently make the following assertion about it: All of these axioms and rules I perceive (with mathematical certitude) to be correct, and moreover, I believe that they contain all of mathematics. If someone makes such a statement he contradicts himself. For if he perceives the axioms under consideration to be correct, he also perceives (with the same mathematical certainty) that they are consistent. Hence he has a mathematical insight not derivable from his axioms.»

But what about subjective mathematics? Can we assert that all of subjective mathematics is contained in the formal system which we believe is sound? In such regard, Gödel would state [91, pp. 309-10]:

> «As to subjective mathematics, it is not precluded that there should exist a finite rule producing all its evident axioms. However, if such a rule exists, we with our human understanding could certainly never know it to be such, that is, we could never know with mathematical certainty that all propositions it produces are correct; or in other terms, we could perceive to be true only one proposition after the other, for any finite number of them. The assertion, however, that they are all true could at most be known with empirical certainty on the basis of a sufficient number of instances or by other inductive inferences. If it were so, this would mean that the human mind (in the realm of pure mathematics) is equivalent to a finite machine that, however,

is unable to understand completely its own functioning.»

Insofar as subjective mathematics is concerned, there is no obstacle to supposing that the human mind is equivalent to a finite machine, which however is unable to understand completely its own functioning. The reason is that to understand completely its own functioning means to know with mathematical certainty that all propositions it produces are correct (in contradistinction to saying that all the consequences from the axioms one after the other are correct) and therefore consistent, contrary to the second incompleteness theorem. Gödel would then go on to derive a disjunction −popularly known as 'Gödel's dichotomy' or 'Gödel's dilemma'− that, as reported by Wang, was considered by Gödel as "the mathematically established fact" on the mind-machine issue which seemed to him "of a great philosophical interest" [91, p. 324]:

> «Either mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule [...] or else there exist absolutely unsolvable diophantine problems of the type specified (where the case that both terms of the disjunction are true is not excluded, so that there are, strictly speaking, three alternatives).»

Gödel probably thought that the first disjunct held, as he believed that the second disjunct had to be false. Wang comments [91, pp. 324-325]:

> «If it [the second alternative of Gödel's disjunction] were true it would mean that human reason is utterly irrational [in] asking questions it cannot answer, while asserting emphatically that only reason can answer them. Human reason would then be very imperfect and, in some sense, even inconsistent, in glaring contradiction to the fact that those parts of mathematics which have been systematically and completely developed (such as, e.g., the theory of 1st and 2nd degree Diophantine equations, the latter with two unknowns) show an amazing degree of beauty and perfection. [...] These facts seem to justify what may be called 'rationalistic optimism'.»

Wang says [91] that "Gödel thinks Hilbert was right in rejecting the second alternative" implying that Gödel did actually reject the second alternative.[25] However, it should be noted

[25]Hilbert stated [45]: "Is the axiom of solvability of every problem a peculiar characteristic of

that Gödel's conclusion is drawn on the assumption that the human mind is equivalent to a finite machine. I will quote the passage in question in full:

> «This inability [of the human mind which equivalent to a finite machine and unable to understand completely its own functioning] to understand itself would then wrongly appear to it as its boundlessness or inexhaustibility. But, please, note that if it were so, this would in no way derogate from the incompletability of objective mathematics. For if the human mind were equivalent to a finite machine, then objective mathematics not only would be incompletable in the sense of not being contained in any well-defined axiomatic system, but moreover there would exist absolutely unsolvable diophantine problems of the type described above, where the epithet "absolutely" means that they would be undecidable, not just within some particular axiomatic system, but by any mathematical proof the human mind can conceive.»

This passage is extremely difficult to understand. On the one hand, "the human mind" that can conceive of mathematical proofs must be the human mind which is supposed to be equivalent to a finite machine, for otherwise the passage would be a *non sequitur*. But if the human mind were equivalent to a finite machine, then the fact that Gödel sides with Hilbert's "rationalistic optimism" would have no point. On the other hand, if there were supposed to exist unsolvable diophantine problems which would be undecidable by any mathematical proof the human mind *per se* (i.e., the human mind as a human mind, in contrast to a finite machine) could conceive, this supposition would be a mere fantasy.

One may well argue that in Wang's interpretation the second alternative of Gödel's dichotomy is a *petitio principii*, since it denies intuitionism from the presupposition of objective mathematics.[26] Another interpretation that suggests itself to me is, contrary to Wang's sug-

mathematical thought alone, or is it possibly a general law inherent in the nature of the mind, that all questions which it asks must be answerable? [...] This conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call: There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no ignorabimus".

[26] Note that if one could reformulate Gödel's disjunction as "either subjective mathematics surpasses the capability of all computers, or else objective mathematics surpasses subjective mathematics, or both alternatives may be true", then the second alternative should be a direct consequence of the fact

gestion, that the human mind which is unable to understand itself is supposed to be equivalent to an absolutely unsolvable number theoretic question. One difficulty with this interpretation is that it makes no distinction between objective and subjective mathematics. Another difficulty is how a finite machine (or the human mind equivalent to it) could ask that absolutely unsolvable question. If it could not ask the question, then the second alternative would not be rejected for the sake of a "rationalistic optimism". In any way, the following extract from Wang [92, pp. 186-7] is an evidence for this interpretation:

> «My incompleteness theorem makes it likely that mind is not mechanical, or else mind cannot understand its own mechanism. If my result is taken together with the rationalistic attitude which Hilbert had and which was not refuted by my results, then [we can infer] the sharp result that mind is not mechanical. This is so, because, if the mind were a machine, there would, contrary to this rationalistic attitude, exist number-theoretic questions undecidable for the human mind.»

Gödel's argument seems to me definitely distinct from Lucas' argument, specially in the following aspect; counter to Lucas, Gödel does not argue that the incompleteness results by themselves show that the mathematical mind is not equivalent to a Turing Machine. An essential extra ingredient that must be added to the incompleteness results to arrive at such conclusion is that the human mind can decide *any* number-theoretic question, including the consistency of its mathematical thought.

However, even granted the truth of rationalistic optimism, does the mind's non-mechanicity hold? Turing did not believe so. In a letter to Max Newman, Turing stated [86]:

> «I think you take a much more radically Hilbertian attitude about mathematics than I do. You say 'If all this whole formal outfit is not about finding proofs which can be checked on a machine it's difficult to know what it is about'. [D]o you have in mind that there is (or should be or could be, but has not been actually described anywhere) some fixed machine [...] and that the formal outfit is, as it were about

---

that objective mathematics is not equivalent to subjective mathematics and that no false proposition can be demonstrated. However, to those who believe that there could be no mathematical truth outside of demonstrability, the distinction of objective and subjective mathematics would be in itself a Platonist presupposition.

this machine. If you take this attitude [...] there is little more to be said: we simply have to get used to the technique of this machine and resign ourselves to the fact that there are some problems to which we can never get the answer. [...] However I don't think you really hold quite this attitude because you admit that in the case of the Gödel example [...] there is a fairly definite idea of a true formula which is quite different from the idea of a provable one.»

On the other hand, Turing argued:

«If you think of various machines I don't see your difficulty. One imagines different machines allowing different sets of proofs, and by choosing a suitable machine one can approximate 'truth' by 'provability' better than with a less suitable machine, and can in a sense approximate it as well as you please.»

Turing's crucial move was to abandon the doctrine that decidability must be achieved by a uniform proof system, accepting that an open-ended rule was needed. He first explored [85] the possibility of achieving arithmetical completeness not by a logical system more powerful than that of Principia Mathematica, which he knew to be impossible, but by an infinite non-constructive sequence of logical systems $L_1$, $L_2$,.... such that, even though for each $L_i$ $\mathrm{Con}(L_i)$ is unprovable by means of $L_i$, one could extend $L_i$ by adding $\mathrm{Con}(L_i)$ to its sets of axioms to yield $L_{i+1}$, which could itself be extended with $\mathrm{Con}(L_{i+1})$, and so on and so forth.

Turing then showed that extension by such axioms may be effectively iterated into the transfinite so that the sequence of logical systems becomes complete, being non-constructive only in the sense that there is no *uniform* method that could be used to generate the whole sequence. These kinds of systematic extensions of a given formal system were called by Turing 'ordinal logics', being later renamed by the scholar Solomon Feferman [20][23] as "(transfinite) recursive progressions of axiomatic theories".

The motivation for the recursive progressions of axiomatic theories in the mind-machine issue was clearly expressed by Turing; the "unsolvability or incompleteness results about systems of logic", he said, amount to the fact that "[o]ne cannot expect that a system will cover all possible methods of proof", so that "when one takes [such limitation] into account one has to admit that not one but many methods of checking up are needed. In writing about ordinal logics I had this kind of idea in mind" [88]. At the same time, Turing had declared [85, p. 209]:

«In pre-Gödel times it was thought by some that it would probably be possible to

> carry this program [of formalizing mathematical reasoning] 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. In consequence of the impossibility of finding a formal logic which wholly eliminates the necessity of using intuition, we naturally turn to "non-constructive" systems of logic with which not all the steps in a proof are mechanical, some being intuitive.»

Turing was suggesting that a human mathematician working according to the rules of a fixed logical system might indeed be a proof-producing Turing machine that through the activity of intuition becomes a different machine, capable of a larger set of proofs.

But how could the function from from mind-stages to proof-finding Turing machines fail to be computable? Turing conjectured that intuitive steps −such as that involved in recognizing the truth of the Gödel formula− may be modelled as "choices" (partially non-deterministic actions) by a Turing machine upon 'advice' by an external operator or 'oracle' (an entity capable of providing an answer to certain number theoretic yes-no questions) an idea which Newman considered [69] resembles a mathematician 'having an idea', as opposed to using a mechanical method. This reflection, however, does not provide a definite answer as to what the source of uncomputability may be.

Had Gödel commented specifically on the multi-machine picture of the human mind, he might have said the following (he was actually commenting on the idea of a race of theorem-proving machines, analogous to the mind stages of the multi-machine picture): "Such a state of affairs would show that there is something non-mechanical in the sense that the overall plan for the historical development of machines is not mechanical. If the general plan is mechanical, then the whole race can be summarised in one machine" [91]. However, Gödel does not clarify this notion of an 'overall plan', and although his remark gets the issues into sharp focus, it does neither help very much with the specific question of where the uncomputability (if any) might come from.

I can only guess that the source of the hypothetical uncomputabilty of the function on the non-negative integers whose value at $i$ is the $i$-th Turing machine on the trajectory is randomness, for if a sequence of integers $T_1, T_2, ...T_i, ...$ is random then there is no function $f(i) = T_i$ that is calculable by a Turing Machine. The point to raise, in either case, is that human arithmetical capacity may at every stage be identical to some Turing machine, even if there is no *single* machine capable of "sumarising" a mathematician's proving-abilities.

## 4.3 The counter-argument from the standpoint of mathematical cognition

The Gödelian arguments may be expressed as the following *reductio*:

> For any machine $M$ and any (mathematically sophisticated) human mind $H$:
> (1) There is a mathematical statement $s$ that $M$ cannot produce as true.
> (2) $H$ can produce $s$ as true.
> Hence $H$ cannot be explained as a machine.

The point I wish to stress in this chapter is that, even if (1)-(2) is true for arbitrary $M$ and $H$, there is a possibility that $s$ can be produced as true by a machine $M' \neq M$ of which $H$ is completely unaware. But how could it possibly be? To suppose that the two things are in contradiction arises from a failure to distinguish two different levels. If we assume materialism, these two different levels are (1) the conscious stream of thoughts and ideas in $H$ and (2) the activity going on in $H$'s brain, which in some mysterious way gives rise to the conscious thoughts and ideas. Now the correctness of the argument's being 'true' has to be a well-defined event in $H$'s consciousness; however, the brain processes which, on the materialist assumption, give rise to this conscious experience are, of course, completely unknown to $H$.[27]

Neuroscientist Warren S. McCulloch and logician Walter Pitts [66] tried to provide a parallelism between mental operations and interconnected "neurons" computing boolean functions (and, or, not...) by means of two simple operations (summing their input signals and comparing the result to a fixed threshold). However, biological neurons are known to operate analogically –"a small error in the information about the size of a nervous impulse impinging on a neuron, may make a large difference to the size of the outgoing impulse." [89, p. 451]– and the admissible input-output values of a logical function are (no matter how fuzzy the logic it implements) discrete. One may counter that, since the observable limits of precision are finite, an analog behaviour of neurons may not suppose and obstacle for it being captured by McCulloch-Pitts model. Even granted that, we face the fact that the physical or biological substrate of neurons radically influences their functional properties. This view is

[27] One may, of course, argue that $H$'s mechanism could nonetheless be understood by *some* $H' \neq H$, however in such case Lucas' argument would be ill-defined.

opposed to the functionalist philosophy, most famously advocated by the early advocator of Computationalist Putnam [73, pp. 299-300]: "Machines forced us to distinguish between an abstract structure and its concrete realization. Not that this distinction came into the world for the first time with machines. But in the case of computing machines, we could not avoid rubbing our noses against the fact that we had to count as to all intents and purposes the same structure could be realized in a bewildering variety of different ways; that the important properties were not physical-chemical".

All in all, McCulloch and Pitts proved that his model described many psycho-neural functions, so let us suppose mind's behaviour can be explained as neural information-processing. If it were so, then a a many-to-one or many-to-many correlation between neural and mental phenomena would exist. However, it remains a mystery whether psycho-neural parallelism holds true. Gödel once stated that "there aren't enough nerve cells to perform the observable operations of the mind" [92, p.190], and in such respect he rightfully argued that "the generally accepted view that there is no mind separated from matter" is "a false prejudice of our time" [91, p. 326].

Although my appeal to possible limitations of self-knowledge may appear gratuitous to the reader, we may have evidence that when we describe our knowledge in the language of mathematics, we may be discussing a secondary language, and so the outward, intelligible forms of our mathematics may not be absolutely relevant from the point of view of evaluating what our true knowledge is. I wish to refer here to a case of numerical cognition. Elaborated logical systems whose axioms and syntactic rules attempted to capture our intuition of what numbers are have been designed throughout history. The most intuitive formalization of arithmetic is first-order Peano arithmetic, which as we pointed out in an earlier chapter extends the following axioms:

1. $\forall x \neg(0 = Sx)$
2. $\forall x \forall y(Sx = Sy \rightarrow x = y)$
3. $\forall x(x + 0 = x)$
4. $\forall x \forall y(x + Sy = S(x + y))$
5. $\forall x(x \times 0 = 0)$
6. $\forall x \forall y(x \times Sy = (x \times y) + x)$

Where '0' has the value zero, '$S$' signifies the successor function, and '$+$' and '$\times$' are interpreted as addition and multiplication.

These axioms formalize the very concrete notion of an endless chain of integers 0,1,2,3,4,...; any number can always be followed by another number that differs from all the preceding ones. But this formalism has one major problem. While Peano's axioms provide a good description of the intuitive properties of natural numbers, they also allow for other numbers, called "non-standard models of arithmetic", which raise considerable difficulties for the formalistic theories of cognition.

It is difficult, in only a few lines, to explain what a non-standard model looks like, but for present purposes a simplified metaphor should suffice. Let us start with the domain $D$ of usual integers 1, 2, 3,... and let us add other elements $D^*$ that we can picture as being "larger than all other numbers"; namely, to the numerical half line formed by the numbers 1, 2, 3,... let us for instance add a second line spreading toward infinity on both sides, e.g. ...-3*, -2*, -1*, 0*, 1*, 2*, 3*... Now let us form the reunion of standard integers and these new elements, and call it the set of "artificial integers":

$$A = \{0,\ 1,\ 2,\ 3,...\text{-}3^*,\ \text{-}2^*,\ \text{-}1^*,\ 0^*,\ 1^*,\ 2^*,\ 3^*...\}$$

The set $A$ verifies all of Peano's axioms. Indeed, as Axiom 1 states there is an artificial number 0 that is not the successor of any other artificial number, and every artificial number has a unique and distinct successor in $A$ (Axiom 2). The successor of 1 is 1+1=2, that of 2 is 2+1=3, and so on; and likewise the successor of -2* is -1*, that of -1* is 0*, that of 0* is 1*, and so on (Axiom 4). From a purely formal point of view, then, $A$ provides a fully adequate representation of the set of integers as defined by Peano's axioms however it is an artificial model of arithmetic, because the elements in $D*$ have no successor.

But could not we definitively fill in this hole adding $\forall x(\neq (x = 0) \rightarrow \exists y(x = Sy)))$ as an axiom? We are now reaching the heart of the paradox. A powerful theorem in mathematical logic, first proved by Skolem [82] and deeply related to Gödel's theorems, shows that no extension of Peano's axioms can abolish non-standard models; this effectively means that an axiomatization of our basic intuitions of number theory is doomed to fail.

This fact constitutes a proof of the plausibility (not mere possibility) of our ignorance of the true human thinking processes (at least in the realm of pure mathematics) and so a fundamental rejoinder to the Gödelian arguments.

# 5 (Non-)implications for AI

In the current chapter I shall focus on arguing that the Gödelian arguments may target certain *approaches* to Artificial Intelligence −assuming that the fatal flaws in their premises which have been repeatedly exposed in the earlier section can be repaired− *but* not the *possibility* of Artificial Intelligence.

The Gödelian arguments strictly allow to conclude that the thinking capabilities of a mathematician in the case of yes-no question cannot be summarized in a fixed formalism such as a Turing machine or algorithm. But how far-reaching is the truth of this conclusion in AI?

Note that a thinking process in the case of yes-no questions is effectively equivalent to a decision problem solving process. Now, while in mathematics the apparatus for solving a problem such as deciding whether a particular statement is true is well known in advance (axioms and rules of inferences) and a definite procedure exists for deciding whether a given solution is correct (checking whether the negation of the solution leads to inconsistency) outside mathematics the situation is completely different. In the real world there is usually no guaranteed sufficiency of knowledge and resources to solve a problem, and there is often no well-defined distinction between solutions and non-solutions −the human mind can solve them in the sense it gives satisfying (not optimal) solutions. An example of these problems is the recognition of varied and distorted patterns, such as classifying the mood revealed by an human face. We shall call the first family of problems Type 1 problems and the second family Type 2 problems.

The question therefore is: why cannot we expect Type 2 problem solving from a computer? while classical AI systems such as automated theorem-provers and expert systems can only solve Type 1 problems, because they have a static inference engine and provide a clear-cut solution to the problem at hand −they are *top-down* reasoning systems− modern AI systems such as machine learning systems have inference rules subject to continual modification according to experience, and provide solutions to decision problems whose correctness is a matter of degree −they are *bottom-up* reasoning systems− so are potential Type 2 problem solvers. The common answer to this question is: because Type 2 problems, by their very nature, have no algorithmic solution and a computer works by following algorithms.

Sure, but algorithms for what? Suppose a machine learning system $S$ designed to solve the problem $P$ of deciding whether a human face's image reveals a mood of a given type. Initially

$S$ does not solve $P$ well. However, $S$ is equipped with a learning algorithm $L_Q$, and after some training, when $P$ appeared again, it got the right answer. During this learning process, does $S$ have an algorithm with respect to $P$? According to the definition of "algorithm" −which involves a computational procedure defined with respect to "determined" input-output pairs− the answer is "No", simply because the system's solution to the problem changes over time. $S$ does follow an algorithm, $L$, but it is not an algorithm *for the solving of* $P$, but an algorithm *for the learning of solving* $P$, which is quite a different matter.

One may well argue that as far as a "changing algorithm" is governed by a meta-algorithm, the two together can be seen as a single algorithm, because the *way* in which our system $S$ modifies its procedures must itself be provided by something entirely computational which is specified ahead of time: the chosen type of decision procedure and a set of input-correct output pairs. The problem with this interpretation is that we can no longer argue that the human mind does not follow algorithms in this sense, because all the examples listed before only shows that the human behaviors change from situation to situation.

If a system can learn and work in a context-sensitive manner, then its decision problem-solving process is not equivalent to a Turing machine, also because there is no fixed initial and final state. For the same problem, in different time and context the system may provide different answers. In this sense, the system can solve problems that are not algorithmic or computable. Similar conclusions have been proposed as "something can be computational at one level, but not at another level" [50], "cognitive processes that, although they involve more than computing, can still be modelled on the machines we call computers" [52]. This conclusion does not contradict the conclusions about the undecidability of certain mathematical problems considered in this essay, because here "problem" and "solution" get different meanings.

But why does the notion of algorithmization work excellently in mathematics but run into trouble in AI? Simply speaking, this is because mathematical reasoning and empirical reasoning follow different "logics"− here "logic" is used in its broad sense, meaning regularity, principle, rationale, and so on. One key issue that distinguishes mathematical reasoning from empirical reasoning is that the former assumes the sufficiency of its knowledge and resources with respect to the problems to be solved. In mathematical reasoning a system does not attempt to change itself to increase its capability. If it fails to solve a problem that is beyond its capability, the fault is not its, but belongs to whoever gave it the problem. On the contrary, in empirical reasoning the system is always open to new knowledge and problems. Because of

this fundamental difference, traditional theories about mathematical reasoning cannot be used in empirical reasoning, and minor revisions and extensions such as non-monotonic reasoning systems are not enough.

In the discussions of the current chapter about the lack of implications of the Gödelian arguments in AI a central point to be stressed is that Type 2 problem solving by machine learning systems provides a way for AI to go beyond the limitations associated with the notion of algorithm. The claim that the problem-solving process of an AI system is equivalent to a Turing machine is a misconception because it treats a certain way to see or use a computer system as the only way to see or use it. The differences among these "ways" often correspond to the difference levels or scales of description; a notion that correctly describes a system at a certain level or scale may be improper or wrong at another one (for a literary exposition of these concepts see [49]).

What makes things even more complicated is the involvement of concepts like "model" or "emulation". To study the nature of a target system $X$, a mathematical tool or platform $T$ is often used to build a model $M$, which shares certain properties with $X$, so can be used to emulate $X$ for certain purposes. In this situation, it is crucial to clearly distinguish the properties of $X$ and $M$ from those of $T$ in general. In the current discussion, the $X$ to be studied is an "intelligent system" as exemplified by the human mind, the $T$ is an ordinary computer system, and the $M$ is the AI system built in $T$ according to the designer's understanding of $X$. A point stressed in this chapter is that neither $X$ nor $M$ should be seen as algorithmic. However, this conclusion does not directly apply to the tools that is used to build $M$. For example, if $M$ is a virtual machine, it is possible to emulate it in a host virtual machine, which can be described abstractly as a Turing Machine. In this case, we should not confuse a virtual machine with its host machine, or a reasoning system with its meta-theory. This discussion is about what properties $M$ (and therefore $X$) has, but not about what properties $T$ has.

To sum up, many properties of intelligence can be explained as the capability of adaptation with insufficient knowledge and resources. The theories about mathematical reasoning cannot be directly applied to AI, because they do not assume these insufficiencies. The development of machine learning systems shows the possibility of building a reasoning system on a new theoretical foundation. To say that the notions in mathematical logic and computability theory −and so the results derived from the undecidability/incompleteness results examined in this essay− do not set limitations for AI, does not mean that AI has no limitation at all.

The problem solving capability of a machine learning system, like that of the human mind, is limited by the system's available resources, so its knowledge shall be incomplete. However, this kind of limitation shows nothing that can be done by the mind but not by the machine.

# References


[1] Audi, R. (ed.) (1999). *The Cambridge dictionary of philosophy*. Cambridge University Press, 1999. ↑11

[2] Benacerraf, P. (1967). *God, the devil, and Gödel*. The Monist, pp. 9-32. ↑26

[3] Boole, G. (1847). *The mathematical analysis of logic*. Philosophical Library. ↑5, ↑6

[4] Boole, G. (1854). *An investigation into the laws of thought*. Walton and Maberly, London. ↑5

[5] Boolos, G. S. (1995). *Introductory note to: Some basic theorems on the foundations of mathematics and their implications*. In [24], pp. 290-304.

[6] Brouwer, L. E. J. (1907). *Over de grondslagen der wiskunde*. ↑7

[7] Brouwer, L. E. J. (1908). *Die onbetrouwbaarheid der logische principes*. English translation in [44], pp. 107-111. ↑7

[8] Cassou-Nogues, P. (2009). *Goedel's introduction to logic in 1939*. History and Philosophy of Logic 30, pp. 69-90. ↑10

[9] Cellucci, C. (1993). *From closed to open systems*. Philosophie der Mathematik. Akten des, 15, pp. 206-220. ↑24

[10] Church, A. (1936). *An unsolvable problem of elementary number theory*. In [17]. ↑11

[11] Church , A. (1937). *Review of Turing (1936)*. Journal of Symbolic Logic 2, pp. 42-43. ↑13

[12] Churchland, P. (1984). *Matter and Consciousness*. Cambridge: MIT Press. ↑5

[13] Copeland, B. J. (ed.) (2004). *The Essential Turing*. Oxford University Press, p. 22. ↑47, ↑48

[14] Copeland, B. J., Carl J. P., & Shagrir, O. (2013). *Computability: Turing, Gödel, Church, and Beyond*. MIT Press, 2013. ↑47

[15] Copeland, B. J. (2015). "The Church-Turing Thesis: Misunderstandings of the Thesis". The Stanford Encyclopedia of Philosophy, Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/sum2015/entries/church-turing/#Bloopers`. ↑13

[16] Council, T. A. (2009). *Engineering in K-12 Education: Understanding the Status and Improving the Prospects*. National Academies Press. ↑1

[17] Davis, M. (ed.) (1965). *The Undecidable: Basic Papers on Undecidable Propositions. Unsolvable Problems and Computable Functions*. Raven Press, New York. ↑9, ↑42, ↑44

[18] Davis, M. (1982). *Why Gödel didn't have Church's thesis*. Information and control 54(1), pp. 3-24. ↑11, ↑12, ↑13

[19] Davis, M. (2001). *Engines of Logic: Mathematicians and the Origin of the Computer*. W. W. Norton & Company, New York.

[20] Feferman, S. (1962). *Transfinite recursive progressions of axiomatic theories*. The Journal of Symbolic Logic 27, pp. 259-316. ↑33

[21] Feferman, S., Dawson Jr, J. W., Kleene, S. C., Moore, G. H., Solovay, R. M., & van Heijenoort, J. (eds.). (1986). *Kurt Gödel. Collected works. Vol. I: Publications 1929-1936*. New York: Oxford University Press. ↑44

[22] Feferman, S. (1988). *Kurt Godel: conviction and caution*. In [25], pp. 150-164. ↑15

[23] Feferman, S. (1988). *Turing in the land of O(z)*. In [25], pp. 150-164. In [43], pp. 113-147. ↑33

[24] Feferman, S. et al. (eds.) (1995). *Kurt Gödel. Collected Works, Vol. III: Unpublished Essays and Lectures*. New York: Oxford University Press. ↑42, ↑44

[25] Feferman, S. (1998). *In the Light of Logic*. Logic and Computation in Philosophy. ↑43

[26] Feferman, S. et al. (eds.) (2003). *Kurt Gödel. Collected Works, Vol. V: Correspondence H-Z*. New York: Oxford University Press. ↑44

[27] Fenstad, J. E. (ed.) (1970). *Selected Works in Logic*. Universitetsforlaget, Oslo. ↑47

[28] Flexner, S. B. (1987). *Random House dictionary of the English language*. Random House. ↑1

[29] Frege, G. (1879). *Begriffsschrift, a formula language, modeled upon that of arithmetic, for pure thought*. In Heijenoort, J. v. (1967), pp. 1-82. ↑5

[30] Gentzen, G. (1935). *Untersuchungen über das logische Schliessen. II*. Mathematische Zeitschrift, 39(1), 405-431. ↑17

[31] Gentzen, G. (1936). *Die widerspruchsfreiheit der reinen zahlentheorie*. Mathematische Annalen, 112(1), pp. 493-565. ↑14

[32] George, A. (1994). *Mathematics and mind*. Oxford University Press. ↑9

[33] Gödel, K. (1929). *On the completeness of the calculus of logic*. Doctoral Thesis, University of Vienna. In [21], pp. 60-101. ↑10

[34] Gödel, K. (1931). *On Formally Undecidable Propositions of Principia Mathematica and Related Systems I*. In [17], pp. 4-38. ↑10, ↑15

[35] Gödel, K. (1933). *The present situation in the foundations of mathematics*. In [24], pp. 45-53. ↑12

[36] Gödel, K. (1934). *On undecidable propositions of formal mathematical systems* (Princeton Lectures). In [17], pp. 39-74. ↑10, ↑11, ↑15

[37] Gödel, K. (193?). *Undecidable Diophantine Propositions*. Lecture Delivered in 193?. In Feferman, S. et al. (eds.) (1995). ↑13

[38] Gödel, K. (1951). *Some basic theorems on the foundations of mathematics and their implications*. In [24], pp. 304-323. ↑28

[39] Gödel, K. (1956). *Letter to John von Neumann, March 20, 1956*. In [26], pp. 372-377.

[40] Gödel, K. (1964). *Postscriptum to Gödel 1934*. In [17], pp. 71-73. ↑10, ↑22

[41] Goldin, D., & Wegner, P. (2004). *The Origins of the Turing Thesis myth*. ↑13

[42] Greenberg, Marvin Jay (1974). *Euclidean and Non-Euclidean Geometries/Development and History*. San Francisco: W.H. Freeman. ↑6

[43] Herken, R. (1992). *The Universal Turing Machine. A Half-Century Survey*. Oxford University Press. ↑43

[44] Heyting, A. (e.d.) (1975). *Collected Works: Vol.: 1.: Philosophy and Foundations of Mathematics*. North-Holland Publishing Company, American Elsevier Publishing Company, Incorporated. ↑42

[45] Hilbert, D. (1900). *Mathematical problems*. Bulletin of the American Mathematical Society 8.10, pp. 437-479. ↑30

[46] Hilbert, D. (1922). *The New Grounding of Mathematics. First Report*. In Ewald(2005), pp. 1115-1133. ↑6

[47] Hilbert, D. (1927). *The foundations of mathematics*. ↑7

[48] Hilbert, D. and Ackermann, W. (1928). *Grundzüge der theoretischen Logik*. Berlin: Springer. 2nd edn. 1938, translated as Principles of Mathematical Logic, New York: Chesea Publishing Co., 1950. ↑17

[49] Hofstadter, D. R. (1979). Gödel, Escher, Bach: an Eternal Golden Braid. Basic Books, New York. ↑40

[50] Hofstadter, D. R. (1985). *Waking up from the Boolean dream, or, subcognition as computation*. In Metamagical Themas: Questing for the Essence of Mind and Pattern, chapter 26. Basic Books, New York. ↑39

[51] Kleene, S. (1952). *Introduction to metamathematics*. Bibliotheca Mathematica (1991), pp. 46-55.

[52] Kugel, P. (1986). *Thinking may be more than computing*. Cognition 22, pp. 137-198. ↑39

[53] Kulstad, Mark and Carlin, Laurence (2013). *Leibniz's Philosophy of Mind*. The Stanford Encyclopedia of Philosophy. Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/win2013/entries/leibniz-mind/`. ↑5

[54] Leibniz, G. W. (1666). *Dissertatio de arte combinatoria*. Sämtliche Schriften und Briefe (Berlin: Akademie Verlag, 1923), A VI 1, p. 163. ↑5

[55] Leibniz, G. W. (1685). *The Art of Discovery*. In [57]. ↑5

[56] Leibniz, G. W. (1704). *New Essays on Human Understanding*. Translated and edited by Peter Remnant and Jonathon Bennett. Cambridge UP, 1982. ↑5

[57] Leibniz, G. W. (1951). *Leibniz selections*. P. P. Wiener (Ed.). Charles Scribner's Sons. ↑45

[58] Lucas, J. (1961). *Minds, machines and Gödel*. Philosophy 36, pp. 112-127, `http://users.oc.ax.uk/~jrlucas/index.html`. ↑23, ↑26

[59] Lucas, J. (1968). *Satan Stultified: A Rejoinder to Paul Benacerraf*. Monist 52, pp. 145-58.

[60] Lucas, J. R. (1970). *The Freedom of the Will*. Oxford University Press. ↑24

[61] Lucas, J. R. (1996). *Minds, machines and Gödel: A retrospect*. In Millican, P. and Clark, A. (eds.), *Machines and Thought: The Legacy of Alan Turing*, pages 103-124. Oxford University Press, Oxford. ↑24, ↑25

[62] Mancosu, P. (2005). *Harvard 1940-41: Tarski, Carnap and Quine on a finitistic language of mathematics for science*. History and Philosophy of Logic 26, pp. 327-357. ↑6

[63] Marí, R. P. (2006). *De Euclides a Java. Historia de los algoritmos y de los lenguajes de programación*. ↑5, ↑8

[64] McCarthy, J. (1989). *Artificial intelligence, logic and formalizing common sense*. In Thomason, R. H., editor, Philosophical Logic and Artificial Intelligence, pp. 161-190. Kluwer, Dordrecht. ↑24

[65] McCarthy, J. (2007). *What is artificial intelligence*. `http://www-formal.stanford.edu/jmc/whatisai` ↑1

[66] McCulloch, W. S. & Pitts, W. (1943). *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics 5(4), pp. 115-133. ↑35

[67] Mendelson, E. (1964). *Introduction to Mathematical Logic*. Van Nostrand.

[68] Nagel, E., & Newman, J. R. (1958). *Goedel's Proof*. New York University Press. ↑25

[69] Newman, M. H. A. (1955). *Alan Mathison Turing*. In Biographical memoirs of the Royal Society (1955), pp. 253-263. ↑34

[70] Pratt, V. (1987). *Thinking Machines: The Evolution of Artificial Intelligence*. Oxford: Basil Blackwell. ↑5

[71] Presburger, M. (1929). *Ueber die Vollstaendigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt.* In Comptes Rendus du I congres de Mathematiciens des Pays Slaves. Warsaw, Poland.

[72] Putnam, H. (1960). *Minds and machines.* In [73], pp. 362-385. ↑25, ↑26

[73] Putnam, H. (1975). *Mind, Language and Reality: Philosophical Papers, Vol. 2.* Cambridge University Press. ↑36, ↑47

[74] Raatikainen, P. *GÃűdel's Incompleteness Theorems.* The Stanford Encyclopedia of Philosophy (Spring 2015 Edition), Edward N. Zalta (ed.), URL = `http://plato.stanford.edu/archives/spr2015/entries/G{ö}del-incompleteness/sup2.html`.

[75] Robinson, R. M. (1950). *An essentially undecidable axiom system.* In Proceedings of the international Congress of Mathematics, Vol. 1, pp. 729-730. ↑16

[76] Rosser, B. (1936). *Extensions of some theorems of Gödel and Church.* The Journal of Symbolic Logic 1(03), pp. 87-91. ↑15, ↑19

[77] Rucker, R. (1982). *Infinity and the Mind: The Science and Philosophy of the Infinite.* ↑7

[78] Russell, S. & Norveig., P. (1994). *Articial Intelligence: A Modern Approach.* Addison-Wesley.

[79] Shagrir, O. (2006). Gödel on Turing on computability. In [14]. ↑12

[80] Simpson J.A. et al. (eds.). *The Oxford english dictionary. Vol. 2.* Oxford Clarendon Press, 1989. ↑1

[81] Skolem, T. (1930). *Über einige Satzfunktionen in der Arithmetik. Skrifter utgitt av Det Norske Videnskaps-Akademi i Oslo, I.* In [27], pp. 281-306.

[82] Skolem, T. (1934). *Über die Nicht-charakterisierbarkeit der Zahlenreihe mittels endlich oder abzählbar unendlich vieler Aussagen mit ausschliesslich Zahlenvariablen.* Fundam. Math. 23, pp. 150-161. ↑37

[83] Smith, P. (2013). *An introduction to Gödel's theorems.* Cambridge University Press. ↑16, ↑17, ↑18

[84] Turing, A. M. (1936). *On computable numbers, with an application to the Entscheidungsproblem.* In [13], pp. 58-90. ↑9

[85] Turing, A. M. (1939). *Systems of logic based on ordinals*. Proceedings of the London Mathematical Society 2(1), pp. 161-228. ↑33

[86] Turing, A.M. (1940). *Letter to Newman, dated "early 1940?" by R. O. Gandy*. Contemporary Scientific Archives Centre, King's College Library, Cambridge. ↑32

[87] Turing, A. M. (1946). *Letter to W. Ross Ashby*. `http://www.rossashby.info/letters/turing.html`.

[88] Turing , A. M. (1947). *Lecture to the London Mathematical Society on February 20, 1947*. In [13], pp. 378-394. ↑33

[89] Turing, A. M. (1950). *Computing machinery and intelligence*. Mind, pp. 433-460. ↑35

[90] Turing, A.M. (ca. 1951). *Intelligent machinery, a heretical theory*. In [13], pp. 472-475.

[91] Wang, H. (1974). *From mathematics to philosophy*. ↑23, ↑28, ↑29, ↑30, ↑34, ↑36

[92] Wang, H. (1996). *A logical journey: from Gödel to philosophy*. MIT Press. ↑27, ↑28, ↑32, ↑36

[93] Wang, P. (2007). *Three fundamental misconceptions of artificial intelligence*. Journal of Experimental & Theoretical Artificial Intelligence 19.3 (2007): 249-268.

[94] Wiener, N. (1948). *Cybernetics, or Communication and Control in the Animal and the Machine*. New York: ffiley. ↑23

[95] Zach, R. (2015). *Hilbert's Program*. The Stanford Encyclopedia of Philosophy (Summer 2015 Edition), Edward N. Zalta (ed.), `http://plato.stanford.edu/archives/sum2015/entries/hilbert-program/#1.4`. ↑14