Minds & Machines (2016) 26:205–226
DOI 10.1007/s11023-015-9382-x

A Theorem about Computationalism and "Absolute" Truth

Arthur Charlesworth[1]

Received: 5 February 2015 / Accepted: 23 September 2015 / Published online: 23 January 2016

Abstract This article focuses on issues related to improving an argument about minds and machines given by Kurt Gödel in 1951, in a prominent lecture. Roughly, Gödel's argument supported the conjecture that either the human mind is not algorithmic, or there is a particular arithmetical truth impossible for the human mind to master, or both. A well-known weakness in his argument is crucial reliance on the assumption that, if the deductive capability of the human mind is equivalent to that of a formal system, then that system must be consistent. Such a consistency assumption is a strong infallibility assumption about human reasoning, since a formal system having even the slightest inconsistency allows deduction of all statements expressible within the formal system, including all falsehoods expressible within the system. We investigate how that weakness and some of the other problematic aspects of Gödel's argument can be eliminated or reduced.

Keywords Consistency · Fallibility · Formal Peano arithmetic · Gödel's Gibbs lecture · Turing machine

Partially funded by a Summer Research Fellowship from the University of Richmond.
The article Charlesworth (2014), which is frequently cited in the current article, is accessible via Open Access from this Web site of the journal Minds and Machines: link.springer.com/journal/11023.

✉ Arthur Charlesworth
ArthurCharlesworth@gmail.com

[1] Department of Mathematics and Computer Science, University of Richmond, Richmond, VA, USA

Introduction

Kurt Gödel investigated implications of his Second Incompleteness Theorem, in giving the American Mathematical Society's 1951 Josiah Willard Gibbs lecture. Gödel presented an argument in support of a conjecture similar to the following, where we use the phrase "perfectly consistent" to emphasize that the consistency is without exception.

Conjecture: *If human mathematical reasoning is perfectly consistent*, then at least one of the following two claims holds:

- Claim I: It is impossible for any computer program to accurately simulate the input–output properties of human mathematical reasoning.
- Claim II: There exists a particular true statement, related to arithmetic, that is impossible for human mathematical reasoning to master.

Claim I contradicts a form of computationalism; i.e., a computational theory of mind. Such computationalism might have significant implications, perhaps implying the lack of free-will, at least as that term is sometimes used, not just in making impulsive decisions Libet et al. (1983), but also in making decisions related to mathematical reasoning that would appear to be based on careful deliberation. There is a variety of well-established opinions on computationalism and free will; for instance, see Kane (2011). We avoid discussing such opinions here since doing so is not essential for obtaining a theorem about computationalism and "absolute" truth, which is the focus of this article.

Claim II might also have significant implications, since it asserts the existence of a particular statement impossible for human reasoning to master and such that the statement is an "absolute" truth; that is, such that the truth of the statement holds—using Gödel's wording here—"in an absolute sense, without any further hypothesis" Gödel (1995), p. 305. Gödel distinguishes such a truth from a truth within "some hypothetical-deductive system such as geometry (where the mathematician can assert only the conditional truth of the theorems)", *ibid*.

This article investigates the above conjecture. As we explain later, we avoid some controversial aspects of the conjecture Gödel considered in his lecture that we show are not essential for obtaining a theorem about computationalism and "absolute" truth. Also, the above Claim II does not mention the particular kind of arithmetical truth Gödel considered (related to polynomial equations with integer coefficients), since the kind of arithmetical truth does not affect the implication of Claim II mentioned in the preceding paragraph. Although the above conjecture differs from the choice of phrasing Gödel used (which is quoted later in this article), the author is indebted to him for the intellectual stimulation such a conjecture can provide.

As emphasized by our use of italics prior to stating the two claims, Gödel's argument required assuming that human mathematical reasoning is perfectly consistent.[1] Such a consistency assumption is a strong infallibility assumption about

[1] Although the "perfectly" in "perfectly consistent" is redundant, we use it occasionally to emphasize that the consistency referred to has no exception. In everyday speech one would not be apt to describe as

human reasoning, because a formal system of the kind Gödel considered that has even the slightest inconsistency allows the deduction of all statements expressible within the formal system, including all falsehoods expressible within that system. It is well-known that many written critiques by scholars, on such a strong infallibility assumption about human reasoning, reject the assumption. Reliance on such a strong infallibility assumption is a serious weakness in Gödel's argument, since without that assumption his argument cannot be carried out, as we explain later.

The current article depends on three assumptions that might themselves be viewed as subjective opinions, but are well-established since they are known to underlie the work of many mathematicians and computer scientists. First is the assumption that Turing machines provide an appropriate mathematical definition for the inherent capabilities of computations. Second is the assumption that each statement in the formal language of Peano arithmetic is either true or false, according to the standard interpretation of the symbols of that formal language. Third is the assumption that each statement provable in Peano arithmetic is true (according to that standard interpretation). The first assumption, known as the Church–Turing Thesis, is accepted by the vast majority of computer scientists. The second and third assumptions are accepted by the vast majority of mathematicians. (A "statement"—sometimes also called a "sentence"—in the formal language of Peano arithmetic is a well-formed expression having no free variable, where a free-variable can cause the truth or falsity of an expression to depend on a later binding of that variable with one of the natural numbers 0, 1, 2,...; to keep the current article concise, we refer the reader to any good logic book—such as Shoenfield (1967) or Hodel (1995)—for an explanation of Peano arithmetic, of the standard interpretation of the symbols in its formal language, and of Zermelo–Fraenkel Set Theory; the explanation on pp. 443–446 of Charlesworth (2014) might suffice.)

In view of the current article's focus on improving Gödel's argument, it is appropriate to mention that Gödel accepted the assumptions mentioned in the preceding paragraph. Gödel used the phrase "mechanical procedure" rather than "computations", and stated:

> We had not perceived the sharp concept of mechanical procedures sharply before Turing, who brought us to the right perspective. And then we do perceive clearly the sharp concept. Wang (2001), p. 205.

Also, Section 8.2 of Wang (2001) explains the reasons Gödel believed that each statement in the formal language of Zermelo–Fraenkel Set Theory is either true or false and that each statement provable in that system is true; it follows that each statement of Peano arithmetic is either true or false. In fact, it is clear from his Gibbs lecture that Gödel believed each statement provable in Peano arithmetic is true.

Given the three assumptions just mentioned, a simple example of a truth that holds "without any further hypothesis" is $0 + 0 = 0$ (whose truth value is that given

Footnote 1 continued
inconsistent a colleague one has known for decades, solely because several decades ago that colleague had asserted "91 is a prime" and then had asserted "91 is not a prime", and it would normally be considered pedantic to point out that it logically follows from those two assertions that "91 is a prime or 4 is a prime" and that "4 is a prime".

according to the standard interpretation), since that statement is a theorem of Peano arithmetic. Of course, one would naturally assert that that particular "absolute" truth is *possible* for human mathematical reasoning to master. In summary: the preceding paragraphs clarify that, *given the three assumptions mentioned*, which are widely-accepted by mathematicians and computer scientists and were accepted and assumed by Gödel, Claim II refers to a truth that holds without further hypothesis.

In investigating the above conjecture, we seek to avoid using philosophical arguments. The emphasis here, instead, is on using mathematics. We employ a mathematical framework that supported two recent theorems related to mathematical logic found in Charlesworth (2006). The first of the theorems is explicated in Charlesworth (2014), and that first theorem justifies the terminology used in the second theorem. Using that framework we prove a new theorem that avoids infallibility assumptions, in contrast to Gödel's argument for the conjecture in his Gibbs lecture. Also, Gödel's conjecture—like the above conjecture—is nonmathematical; for instance, due to the lack of a mathematical definition of what Gödel called "the human mind" (within the first Gödel quote of our section "The Conjecture in Gödel's Gibbs Lecture") and what the above conjecture calls "human mathematical reasoning". The two recent theorems avoid the need for such a definition, via a more general concept of an "agent", and the mathematical model underlying the two theorems provides a simple mathematical definition of that concept. This article also explains how the above conjecture avoids some other problematic aspects of his argument. The proofs of the two theorems could be carried out in principle in Zermelo–Fraenkel Set Theory, so they satisfy the current standard criterion for rigorous mathematics. In turn, the application of the two theorems is like any other application of pure mathematics to the physical universe.

The rest of the article is organized as follows. The next section discusses Gödel's Second Incompleteness Theorem, which leads to a section presenting Gödel's actual conjecture as well as a discussion of the questionable consistency of human reasoning. That is followed by a section discussing the mathematical framework of the two recent theorems mentioned above, which leads to a section using that framework to present a new theorem that generalizes the conjecture at the beginning of this article, stated in terms of any agent—not just human agents—and *without assuming any strong infallibility property about such an agent*. After a section that considers some of the specific phrases Gödel used within his conjecture, the article concludes with a summary.

Gödel's Second Incompleteness Theorem

Gödel's Incompleteness Theorem roughly states that for each formal axiomatic system S that includes formal Peano arithmetic (which we denote by PA) and whose set of (number codes of) theorems can be checked by a Turing machine, *if S is consistent*, then S is incomplete. [See any good logic book, such as Shoenfield (1967) or Hodel (1995).] The latter means there exists a statement within S such that neither it nor its negation has a formal proof within S.

Gödel's Second Incompleteness Theorem is a stronger result that gives a particular example, Con_S, of such an unprovable statement within S. The meaning of Con_S, according to the standard interpretation of PA, is that S is consistent. (See Hodel 1995.) Thus, *if S is consistent*, then we have the following property: Con_S is true according to the standard interpretation but unprovable within S. It is important to emphasize here (as explained further at the end of this section) that the hypothesis of the just-stated implication is not only sufficient for the implication's conclusion but *necessary for that conclusion*; that is: **if** S **is not consistent, then there is no statement within** S **that is unprovable within** S.

Since Con_S is true but unprovable within S, it is customary to refer to Con_S as a "Gödel statement" of S.

Roughly speaking, if the mathematical deduction of a set H of one or more humans is entirely determined by the kind of Turing machine-based formal axiomatic system S described above, and *if S is consistent*, then H cannot deduce Con_S using mathematical deduction. The preceding sentence, although a nonmathematical statement, helps convey the spirit of attempts to apply the Second Incompleteness Theorem to humans. That is presumably what Alan Turing had in mind in 1947 when he stated:

> In other words, then, if a machine is expected to be infallible, it cannot also be intelligent. There are several theorems which say almost exactly that. But these theorems say nothing about how much intelligence may be displayed if a machine makes no pretense at infallibility. Turing (1986), p. 124.

It is important to notice that consistency is a strong infallibility property and an extremely brittle assumption. As mentioned above, the Second Incompleteness Theorem says that if S is consistent then we have the following property: Con_S is true according to the standard interpretation but unprovable within S. If the mathematical deduction of a set H of one or more humans is entirely determined by the kind of formal axiomatic system S described above and there is even a single statement A, among the infinitely many statements of S, such that H decides (via formal proofs) that both A and its negation hold, then by the definition of "consistency" S is not consistent. It then would follow from a simple well-known argument—for the gist see our footnote 1 and for a formal logic approach see the third paragraph on p. 467 of Charlesworth (2014)—that every statement of S would be provable within S, regardless of whether the statement is true or false; for instance, *the false statement* Con_S *would be provable within* S. The deductive capability of H would then be trivial.

The Conjecture in Gödel's Gibbs Lecture

Here is the phrasing Gödel used, in what we shall call his "Gibbs conjecture".

> *Either mathematics is incompletable in this sense, that its evident axioms can never be comprised in a finite rule, that is to say, the human mind (even within the realm of pure mathematics) infinitely surpasses the powers of any finite*

> *machine, or else there exist absolutely unsolvable diophantine problems of the type specified* ... Gödel (1995), p. 310, emphasis in original.[2]

By a "finite machine" he meant a machine equivalent to a Turing machine. His lecture explained that by "absolutely unsolvable" problems he meant problems that are "undecidable, not just within some particular axiomatic system, but by *any* mathematical proof the human mind can conceive" (*ibid*; emphasis in original). One can view a "diophantine problem" as a problem of determining whether or not certain polynomial equations with integer coefficients have natural number solutions.

Here is the argument Gödel gave for the above Gibbs conjecture, where "*this* theorem" refers to his Second Incompleteness Theorem:

> It is *this* theorem which makes the incompletability of mathematics particularly evident. For, *it makes 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 certainty) that they are consistent. Hence he has a mathematical insight not derivable from his axioms. Gödel (1995), p. 309, emphasis in original.

In the words "... that they are consistent. Hence ..." Gödel makes explicit the dependence of his argument on the assumption that human deduction is consistent. That crucial dependence is because, as explained in our preceding section, the consistency of S is *necessary* for obtaining the conclusion of Gödel's Second Incompleteness Theorem.

Although Gödel's argument supporting the Gibbs conjecture assumed an *idealization* of humans, he recognized the fallibility of actual humans, according to perhaps the leading interpreter of Gödel who knew him personally, Hao Wang:

> There is much more in Gödel's philosophy of mathematics than is commonly believed. For instance, contrary to the general impression, Gödel affirms the fallibility of our mathematical intuition and investigates the different degrees of clarity and certainty that exist within mathematics. Wang (2001), p. 5.

Indeed, the assumption that one can view human reasoning as consistent is highly questionable, as indicated by the 1947 Turing quote in our preceding section, and the following ten quotes.

[2] One might ask: Was Gödel's claim actually stronger, that the human mind surpasses machines? Not in his Gibbs conjecture. As Solomon Feferman put it: "Typically cautious, in his Gibbs lecture he stated his conclusion from the second incompleteness theorem only as a disjunction, despite his personal conviction that mind is not equivalent to a finite machine. Apparently the reason he did that is because he did not feel he had a knock-down proof of the falsity of the mechanist position." Feferman (2006), p. 145.

– Martin Davis, in the journal *Behavioral and Brain Sciences*:

"If *insight* is involved, it must be in convincing oneself that the given axioms are indeed consistent, since otherwise we will have no reason to believe that the Gödel sentence is true. But here things are quite murky. Great logicians (Frege, Curry, Church, Quine, Rosser) have managed to propose quite serious systems of logic which later have turned out to be inconsistent." Davis (1990), emphasis in original.

– Daniel Dennett, in the journal *Behavioral and Brain Sciences*: One

"can perhaps fervently believe, and assert, that the joint or Ideal Mathematician is consistent and capable (in principle) of intuiting every mathematical truth (and no falsehoods), but he cannot hope to persuade those of us who find this an unlikely and unmotivated dogma by offering a mathematical proof, and there seems every empirical reason for simply disbelieving it." Dennett (1990)

– Rudi Lutz, in the journal *Behavioral and Brain Sciences*: The

"argument that mathematicians use an infallible algorithm for determining mathematical truth is unconvincing given several instances in the history of mathematics of 'theorems' that later turned out to be false." Lutz (1990)

– Drew McDermott, in the journal *Behavioral and Brain Sciences*:

"any plausible candidate for an algorithm that duplicates a person would, far from being an infallible procedure, have incomplete and even contradictory beliefs about mathematics. " McDermott (1990)

– Don Perlis, in the journal *Behavioral and Brain Sciences*:

"What is needed to get our hands on an actual instance of the Gödel sort is, typically, knowledge of the consistency of the system in question. Now, this is emphatically not something we can in general see." Perlis (1990)

– Adina Roskies, in the journal *Behavioral and Brain Sciences*:

"Our intuitions about the self-evidence or truth of mathematical statements, however, often turn out to be mistaken. For instance, Hilbert believed that arithmetic was complete until Gödel proved him wrong, and Cantor and Frege thought their formulation of set theory was consistent until Russell advanced his paradox." Roskies (1990)

– Tony Dodd, in the journal *Artificial Intelligence Review*:

  "there is a hazy boundary area, whether near to or far from ordinary mathematics is immaterial, where we would not explicitly assert the consistency of our [mathematical] beliefs." Dodd (1991)

– Marvin Minsky, in the book *The Third Culture*:

  "There's no reason to assume ... that either human minds or computing machines need to be perfectly and flawlessly logical" Minsky (1995).

– John Barrow, in the Oxford University Press book *Impossibility: The Limits of Science and the Science of Limits*:

  "In all these debates, a single assumption is always lurking beneath the surface. It is the assumption that the workings of the brain are infallible, when viewed as logical processors. There is really no reason to believe this (and many reasons not to!)." Barrow (1998), p. 232.

– Stuart Russell and Peter Norvig, in the book *Artificial Intelligence: A Modern Approach*:

  ... if anything, humans are known to be inconsistent. This is certainly true for everyday reasoning, but it is also true for careful mathematical thought. A famous example is the four-color map problem. Alfred Kempe published a proof in 1879 that was widely accepted and contributed to his election as a Fellow of the Royal Society. In 1890, however, Percy Heawood pointed out a flaw and the theorem remained unproved until 1977. Russell and Norvig (2010), p. 1023.

Rather than provide a critique of Gödel's Gibbs lecture, this article's focus is to present as expeditiously as possible a theorem that generalizes the nonmathematical conjecture stated at the beginning of this article. Such a theorem is presented after our next section. Critiques of Gödel's Gibbs lecture have already been published. See for instance the critique by Solomon Feferman (2006), the commentary by George Boolos in the pages immediately preceding Gödel (1995), and an article by Richard Tieszen (2006). Additional information about Gödel's Gibbs lecture appears in the tenth chapter of a biography of Gödel by John Dawson (1997) as well as in a dozen places within Wang (2001).

Mathematical Framework

A standard unifying concept in Artificial Intelligence is the notion of an "agent"; see Russell and Norvig (2010). For the theorem we present in the next section, we use a framework provided by Charlesworth (2006) and explicated in Charlesworth

(2014), in which an **agent** is defined simply to be a function from a set of natural numbers to the set of natural numbers.

One might ask: how could such a mathematical function be relevant to a conjecture about human reasoning? That question is not an issue because it has been shown, using a modification of a well-known approach used by Turing, that such a function can model any single real-world agent in an environment providing appropriately restrictive input, where the agent consists of "one or more ... supercomputers, humans, humans with surgically-implanted silicon memory chips, and so forth" Charlesworth (2014), p. 462. As explained on pp. 461–464 of that article, *the real-world agent modeled by the function can be far from infallible*. Such an agent need not correctly decode its real-world inputs and need not produce real-world outputs that can be decoded to yield the required format for an answer to a question. For a given coded input question, when an agent produces real-world outputs that (when successfully decoded) do yield an answer, it is the first such answer by the agent that is considered as the output of the function. Also, given appropriately coded inputs for the halting problem for a Turing machine that has over a hundred different encodings, it is possible for an agent to give the correct answer to the halting problem for one encoding of that machine and the opposite answer (to the exact same halting problem) for the remaining encodings.[3] Notice that *such an agent need not be correct on even as much as a subset of 1% of its decisions*.

Of course, an *application* of mathematics to any physical agent cannot itself be fully mathematical since there is no fully mathematical definition of any physical object. To illustrate the relevance to the issue of computationalism of a function from natural numbers to natural numbers, we informally observe the following. In chess, any single entire board configuration can be coded as a single input natural number[4] and the chess player's choice of subsequent move can also be coded as a single output natural number. The input–output properties of a player are required to be fully due to the decision making of the player, so access by the player during the game to books, the internet, or assistants is prohibited; compare that with the phrase "an environment providing appropriately restrictive input" in the preceding paragraph. Also, the player cannot retract a move the player has made;[5] compare that with the phrase "first such answer" in the preceding paragraph. Notice that if it is impossible for any computer program to accurately simulate the input–output properties of a specific human grandmaster deciding the next move to make, then it is impossible for any computer program to accurately simulate the decision making (and hence some aspect of the overall reasoning) of that human.[6]

[3] The typical Turing machine has more than one single natural number encoding, since its states and tape symbols can be arbitrarily ordered Hopcroft et al. (2007), p. 379.

[4] The EPD (Extended Position Description) of the board configuration is a finite list of elements from a predetermined finite set. It is well-known that any such list can be coded as a single natural number.

[5] See Article 4 of the Laws of Chess FIDE (2014). Another important fact: a winning chess player can make a single *illegal* move, which is ignored (except for clock issues and for disallowing another illegal move), by Article 7 of those Laws of Chess. Our mathematical framework ignores all incorrectly *formatted* output by an agent, as discussed near the end of the section "The Theorem".

[6] The converse need not hold. For it is conceivable that the input–output properties of a noncomputational agent—perhaps a human, if human reasoning is noncomputational—in playing chess

The same computationalism-related implication also holds for generalized chess in which the chessboard has size n by n, with $n \geq 8$. Generalized chess plays a more important role in theoretical computer science than does usual chess. For a consideration of generalized chess, see Fraenkel and Lichtenstein (1981), which has a purpose different from our purpose of simply illustrating the relevance of a function on the natural numbers to the issue of computationalism. The single natural number code for a board configuration would include information about the value of n for the instance of generalized chess. (Of course, the ten digits suffice to construct the numeral for any n.) Although the halting problem game the two recent theorems in Charlesworth (2006, 2014) are about seems quite different from generalized chess, it is the case that for both kinds of games—unlike for usual chess—there is an infinite number of possible natural number inputs (and outputs).

Modeling a real-word agent, such as an individual human's mathematical reasoning, with a mathematical function is a useful simplification. But, by itself, that simplification falls far short of enabling one to obtain a theorem *whose hypothesis would be widely accepted*, related to the Gibbs conjecture. As explained in our preceding section, obtaining a theorem about human reasoning via Gödel's Second Incompleteness Theorem requires a hypothesis expressing a highly questionable point-of-view: that human mathematical deduction is (perfectly) consistent.

Achieving a widely-acceptable theorem thus requires replacing the role of the Second Incompleteness Theorem by a fundamentally different result, applicable to a real-world agent that need not satisfy strong infallibility properties.[7] The theorem in the next section is such a theorem, and is related to halting problems.

Of course, a "halting problem" is any problem that, for a piece of software with some input for that software, asks which of these two statements is correct: the software (eventually) halts, the software does not halt.[8] Questions related to halting problems as well as attempts at formal proofs are coded as natural numbers. We use Turing machines as the standard mathematical abstraction for capturing the notion of software. It is common to use computer software equivalent in power to a Turing machine to construct artificial neural networks. One can also simulate a parallel program, such as IBM's Deep Blue chess-playing AI system and such as IBM's Watson Jeopardy-playing AI system, by a Turing machine program that switches around among the various processes it simulates.

Footnote 6 continued
(alternatively in playing blackjack, or producing square roots) could be accurately simulated by a chess-playing computer program that agent might have learned (alternatively, by a "card counting" program, or by the Babylonian method for finding square roots).

[7] One might ask: Where is the assumption of an infallible agent avoided? Earlier in this section we mentioned several ways that the mathematical model of an agent's reasoning using a function on the natural numbers avoids infallibility assumptions about that agent's reasoning. See pp. 465–466 of Charlesworth (2014) for an explanation of additional ways infallibility assumptions on an agent are avoided.

[8] One might ask: How important are halting problems? Their wide importance is elucidated from p. 442 through the first full paragraph on p. 447 of Charlesworth (2014).

The mathematical framework we use appears in Charlesworth (2006), aimed at a readership of specialists familiar with applying logic to systems like PA and formal Zermelo–Fraenkel Set Theory. An explanation for a readership experienced in Artificial Intelligence and/or Cognitive Science appears in Charlesworth (2014), which—assisted by its Electronic Supplement—addresses over twenty relevant questions. (There is Open Access to that article via the hyperlink just prior to the current article's Introduction.) Although the expressed focus of those questions is on one specific theorem of Charlesworth (2006), many of those answers given are relevant to the theorem in our next section.

As suggested in our Introduction, the mathematical framework we use assumes that all axioms of PA are true; i.e., that PA is "sound". The formal proofs given by an agent $\mathcal{A}$ are given within a **suitable** formal system $\mathbb{F}$. To be suitable[9] a formal system must have a formal language that includes at least the symbols of PA, must have axioms that include at least those of PA, must have Gödel numbering compatible with that of PA, and must satisfy the property that any statement of PA provable in $\mathbb{F}$ is true[10] according to the standard interpretation of PA. There is no requirement that an algorithm must exist for determining which of the statements of $\mathbb{F}$ are axioms[11], or that there must be an algorithm for checking the correctness of proofs in $\mathbb{F}$. Thus the set of theorems of $\mathbb{F}$ need not be computably enumerable.

It follows from the section "Definition of 'Adequate' Formal System" on pp. 458–459 of Charlesworth (2014) that there exists what we call here a "suitable" formal system whose set of axioms is not computably enumerable and there also exists a "suitable" formal system whose set of axioms is computably enumerable. In fact there exist such systems that are natural to consider.

The resulting approach avoids the *petitio principii* fallacy of begging a question.[12] If all proofs by any agent were required to be within a specific $\mathbb{F}$ whose theorems were computably enumerable, then by the mere act of imposing such a requirement a fundamental issue related to computationalism would become trivial. For there would in principle then be a single Turing machine that would be as successful in providing a yes/no decision for any mathematics conjecture (necessarily in $\mathbb{F}$'s formal language, because of the requirement we are considering),

[9] A "suitable" formal system would also be "adequate", according to the definition of the latter term in both Charlesworth (2006, 2014), if its axioms also include the members of a specific computationally-defined set $\mathcal{T}$ of formulas of PA, where the definition of the specific set $\mathcal{T}$ is independent of the agent; the weaker concept of suitable suffices for the new theorem presented in this article.

[10] One might ask: Why is assuming a soundness property of $\mathbb{F}$ not assuming the agent $\mathcal{A}$ is infallible? That is not an issue, because of the distinction between the agent and the system $\mathbb{F}$. See the section on p. 466 of Charlesworth (2014) entitled "How Can One Assume that a System Used by an Agent is Sound Without Assuming Agent Infallibility?".

[11] Axioms can become generally accepted that previously were not, such as the axiom asserting the existence of an infinite set, largely due to investigations of Georg Cantor in the 1800s. Gödel and some others have questioned whether the current standard criterion for rigor in mathematics, provability in principle in Zermelo–Fraenkel Set Theory, must by definition forever be the standard criterion for rigor; see Section 3 of Charlesworth (2006).

[12] In contrast, there is no *petitio principii* fallacy in the important facts that, in principle, there exists an algorithm to check correctness of proofs in PA and there exists an algorithm to check correctness of proofs in Zermelo–Fraenkel Set Theory.

and in proving that such a decision is logically correct, as any agent could be, indeed as all other agents combined could be. That single Turing machine would simply take as input the given conjecture, generate the unending list of theorems of $\mathbb{F}$, and when and if one of the theorems is the given conjecture or its negation, report the decision. Whenever the conjecture could be settled (necessarily within $\mathbb{F}$, because of the requirement we are considering), that single Turing machine could settle the question; otherwise neither that Turing machine nor any other agent or combination of agents could settle the conjecture. Note that this explanation has not conflated "computably enumerable" with the stronger concept of "computable".

When we refer to the meaning of a formula of PA, we shall assume the standard interpretation of the language of PA. As mentioned in footnote 3, a typical Turing machine has more than one encoding as a natural number, which we will simply refer to as a (natural number) **code** for the Turing machine. Our interest will be in statements of PA that relate to individual Turing machines. Since each such statement will be expressed in PA's language via a *specific choice of code* for a specific Turing machine, sometimes it will be important to mention the existence of such a specific code choice.

As mentioned in Charlesworth (2006), using standard techniques of mathematical logic it can be shown that there exists a formula H of PA having exactly one free variable (i.e., having exactly one variable that is not bound within the scope of a universal or existential quantifier) and having the following two properties, where $H[\overline{\#P}]$ denotes the substitution of PA's notation representing the (natural number) code for Turing machine P into the slots of H where the free variable of H is not bound by a quantifier:

- "P halts" if and only if $H[\overline{\#P}]$ is true
- "P halts" if and only if $H[\overline{\#P}]$ is a formal theorem of PA

for every choice $\#P$ of code for P. Here the input for P includes a code for P itself, within a pair $\langle P, Halt?\rangle$ explained later in this section. The meaning of the arithmetical statement $H[\overline{\#P}]$ is "P halts", and the meaning of the arithmetical statement $\neg H[\overline{\#P}]$ is "P does not halt", where $\neg$ denotes negation.

Our goal is a theorem somewhat like our next sentence, where an agent is required to give formal proofs in a "suitable" formal system $\mathbb{F}$ and where $\mathcal{A}$ is an agent that need not be consistent or otherwise infallible. At least one of the following two assertions holds:

- (a) It is impossible for any software to accurately simulate the input–output properties of $\mathcal{A}$.
- (b) There exists a true halting-problem arithmetical statement such that, when asked about the truth value of the statement, $\mathcal{A}$ cannot decide that the statement is true.

It is straightforward to define what it could mean to say that $\mathcal{A}$ is "asked about the truth" of such a statement: $\mathcal{A}$ would be given the Gödel code for the PA statement

and asked to provide as output the code for a list (that could be a formal proof) whose last member is the statement, when $\mathcal{A}$ "decides" the statement is true. (What $\mathcal{A}$ is expected to do when $\mathcal{A}$ decides the statement is false will turn out to be irrelevant to this explanation, so there is no need to suggest what those details might be.) Let us try to prove the above theorem attempt. Suppose both (a) and (b) fail to hold. The failure of (a) to hold implies that there exists a Turing machine that accurately simulates the input–output properties of $\mathcal{A}$; let S denote such a Turing machine. The failure of (b) to hold implies that it is possible for S to correctly decide the truth value of each true halting statement of the form $H[\overline{\#P}]$ or $\neg H[\overline{\#P}]$. Due to the Unsolvability of the Halting Problem, we can successfully reach a contradiction if we can show that S solves the halting problem as follows. When S is given as input the code for any Turing machine P (together with input for P), one of the two halting statements about P—that it halts, alternatively that it fails to halt—will be true, and S will definitely (based upon the logic of the current proof attempt) be able to decide that the true one of those two halting statements is indeed true. Just let (a straightforward modification of) S indicate, as its output, which of those possibilities it decides is true. That contradicts the Unsolvability of the Halting Problem. [End of proof attempt]

Saying that the S in the preceding paragraph can correctly decide that each true arithmetical statement is true does not imply that S cannot also decide that some false arithmetical statements are true; it does not even imply that S cannot decide that every false arithmetical statement is true. That is, the proof attempt in the preceding paragraph assumes that $\mathcal{A}$ is consistent about what it decides is true whereas a very important goal (emphasized throughout the current article, and made clear in the hypothesis of that theorem attempt) is the lack of such a consistency assumption. If $\mathcal{A}$ is inconsistent, then S is inconsistent, and thus the fact that S decides on the truth value of a halting assertion provides no indication of the actual truth value of that halting assertion, so there is no reason to conclude that the output of S (mentioned near the end of the above proof attempt) would be correct. Thus a *fatal flaw* is evident in that attempt. Trying to repair that flaw, by assuming S could always attempt to prove the correctness of a potential decision prior to making the decision and use an algorithm to check the correctness of each such proof, overlooks the importance of the part of the hypothesis of the theorem attempt that avoids making the assumption that $\mathbb{F}$ provides such an algorithmic proof-checking capability. (Recall the discussion earlier in this section about avoiding a *petitio principii* fallacy in defining "suitable" formal system.)

We presented the above proof attempt because some readers might otherwise have considered such an attempt themselves, might have overlooked its flaw, and might wonder why we did not use that approach, which is simpler than our approach below. In fact, no repair of that proof attempt is possible, since there is a counterexample to the above theorem attempt: just take $\mathcal{A}$ to be a function that "decides" that every statement of PA is true, regardless of whether such a decision is correct. For that $\mathcal{A}$, both (a) and (b) are easily seen to fail to hold, because it is easy to define a Turing machine $\mathcal{A}$ that asserts that each statement of PA is true and because there is no true arithmetical statement which that $\mathcal{A}$ would not decide is true.

Thus our approach must be different from the above faulty attempt. The gist of our approach is that we bind the two possibilities—halt, does not halt—for each halting problem into a *single* input to be given to an agent, and we ask the agent to make a choice between them. Such an approach is suggested by Lemma 7.15 of Charlesworth (2006), which is also explained on p. 455 of Charlesworth (2014) and its eight pages leading up to that page.

Before giving additional definitions, we emphasize again that the typical Turing machine has more than one (natural number) code; see footnote 3. As a result, the notation $H[\overline{\#P}]$ to represent a PA statement is ambiguous even when the specific Turing machine P is not ambiguous, since the actual PA statement would depend on which code of P plays the role of $\#P$. Avoiding such an ambiguity is not usually important because the truth value of $H[\overline{\#P}]$ (as well as its provability status in any suitable formal system $\mathbb{F}$) is unaffected by the choice of code for P. But avoiding such an ambiguity in this article is important: we do not assume an agent must be logically consistent, hence an agent's decision about the truth value of $H[\overline{\#P}]$ can differ for different choices of codes for P. Rather than define unnecessarily complicated unambiguous notation to use for such a PA statement that would have the notation for a natural number m within the notation, we use careful explanations in which "specific" will mean specific to a choice of code for the Turing machine.

We let *Halt*? denote the number 1, since doing so helps to clarify the nature of the notation now explained. We let $\langle P, Halt? \rangle$ denote a computational encoding (whose inverse is also computational) of the two-member list containing a specific code for P and also containing the number 1. Let $\mathcal{A}$ be an agent, and let $\mathcal{A}(n)$ denote the output (if there is an output) of $\mathcal{A}$ for input n. Then $\mathcal{A}$ answers halting problem related questions in the following way. If $\mathcal{A}$ produces output for the input $\langle P, Halt? \rangle$ (for a specific choice of code $\#P$ for P), then that output is a **decision about** $\langle P, Halt? \rangle$ iff $\mathcal{A}(\langle P, Halt? \rangle)$ is a number that, first, can be computationally decoded in one way to produce $\mathcal{A}$'s **binary decision**—exactly one of $H[\overline{\#P}]$ and $\neg H[\overline{\#P}]$ (for the specific code choice of P)—and, second, can be computationally decoded in another way to produce a list of codes of formulas, in a situation where $\mathcal{A}$ gives an attempted proof of the correctness of its binary decision. (That list of codes of formulas can be the empty list; $\mathcal{A}$ is permitted to make a binary decision—including an incorrect binary decision—for a halting problem without giving any justification for the binary decision; it is thus easy for $\mathcal{A}$ to give an output that is at least in the acceptable output *form*.) $\mathcal{A}(\langle P, Halt? \rangle)$ is **logically correct** in $\mathbb{F}$ iff $\mathcal{A}$'s attempted proof is a (correct) formal proof in $\mathbb{F}$. One can define such a two-fold computational decoding in a straightforward way; the details here are relatively unimportant, see the footnote on p. 453 of Charlesworth (2014). $\mathcal{A}$ is a **decision system** iff for each Turing machine P (and each code for P) the output (if there is any) that $\mathcal{A}$ produces for the input $\langle P, Halt? \rangle$ is a decision about $\langle P, Halt? \rangle$; a decision system is called a "deduction system" in Charlesworth (2006).

We say software **accurately simulates the input–output properties of** $\mathcal{A}$ to mean the software has the same (natural number) input–output properties as $\mathcal{A}$. By a **specific halting problem** we mean the problem of deciding the output to give for an input $\langle P, Halt? \rangle$ that contains a *specific* code for a Turing machine P. By the

corresponding true arithmetical statement for the specific halting problem $\langle P, Halt? \rangle$ we mean the true one of the pair of PA statements $(H[\overline{\#P}], \neg H[\overline{\#P}])$, where the same specific code for P is used as was used in the specific halting problem.

The Theorem

As mentioned earlier, since there is no known mathematical definition of "human", the conjecture stated at the beginning of the article is nonmathematical. This section presents and discusses a mathematical theorem, whose hypothesis is also much more widely applicable.

Notice that assertion (1) and assertion (2) in the following, like the two assertions within Gödel's Gibbs conjecture, form an Inclusive Or.

Theorem (COAT, Computationalism-impossible Or "Absolute" Truth)
Let $\mathbb{F}$ be any suitable formal system, in which formal proofs are to be given, and let $\mathcal{A}$ be any agent, which need not be consistent or otherwise infallible. At least one of the following two assertions holds:

1. *It is impossible for any software to accurately simulate the input–output properties of $\mathcal{A}$.*
2. *There exists a specific halting problem such that $\mathcal{A}$ cannot decide to choose its corresponding true arithmetical statement.*

Moreover, there is a computational function that maps any decision system that is a counterexample for (1), if such a counterexample exists, to a **particular** *example of (2) that demonstrates $\mathcal{A}$'s lack of mastery of arithmetical truth.*

Proof We prove the inclusive or within the theorem by supposing both assertion (1) and assertion (2) fail to hold, and reaching a contradiction. The failure of (1) to hold implies that there exists a (deterministic) Turing machine that accurately simulates the input–output properties of $\mathcal{A}$; let S denote such a Turing machine. The failure of (2) to hold then implies that for each specific halting problem, S's binary decision is the corresponding true arithmetical statement. Then a Turing machine M could solve the halting problem, by doing the following: Given any input code for a Turing machine P (together with input for P), let M first construct the two-member list $\langle P, Halt? \rangle$, using the specific given code for P. Then let M provide the code of that list as input to S, thereby asking S which of the PA statements in the pair $(H[\overline{\#P}], \neg H[\overline{\#P}])$ is true, where the same specific code for P is used. Finally, let M output S's binary decision, which is guaranteed (by the *reductio ad absurdum* assumption within this proof) to be the corresponding true statement. That capability of Turing machine M contradicts the Unsolvability of the Halting Problem.

We now prove the theorem's final sentence. By Lemma 7.15 of Charlesworth (2006) (which uses "deduction system" for what we call "decision system"), there is a Turing machine that maps the code for any decision system Turing machine P to the code for a Turing machine P' such that P cannot produce a correct binary decision for what we here call the specific halting problem $\langle P', Halt? \rangle$, where the same specific code for P' is used within it. (That Turing machine P' might seem analogous to a Gödel statement, but an important difference is the lack of an assumption of *infallibility* of P in the hypothesis of that lemma.) It is thus sufficient to see that there is a Turing machine that maps the code for any such P' to the code for the corresponding $\langle P', Halt? \rangle$, which is straightforward. □

Since it can be applied to any agent without requiring that the agent satisfy an infallibility property, the COAT Theorem is much more widely applicable than the Gibbs conjecture. Another aspect of its wide applicability is the fact that the set of theorems in the $\mathbb{F}$ chosen for the application need not be computably enumerable (but can be, if such an $\mathbb{F}$ is desired). The statement and proof of the COAT Theorem also avoid using the mathematically-undefined concept of human reasoning.

Suppose the mathematical function $\mathcal{A}$ models some real-world agent so that the input–output of $\mathcal{A}$ is fully due to that real-world agent. Informally speaking, if assertion (1) holds then it is impossible for any software to accurately simulate that real-world agent; for consider the contrapositive of that implication. Compare this with the illustration of the human chess grandmaster near the beginning of the preceding section.

Assertion (2) implies—and is considerably stronger than saying that—it is impossible for $\mathcal{A}$ to produce an output related to the specific halting problem in (2) such the output contains a formal proof and satisfies our definition of being "logically correct" output. In the rest of this paragraph, we discuss the situation when assertion (2) is true, letting s denote the corresponding true arithmetical statement mentioned in that assertion. The statement s is a true statement of PA that $\mathcal{A}$ cannot always decide is true. That is, $\mathcal{A}$ is limited in its mastery of a particular arithmetical truth in the sense that $\mathcal{A}$ *cannot decide s is true when given input that is the specific halting problem mentioned in (2) and thus is input related in a straightforward computational way to the pair consisting of s and its negation.* [As illustrated by the proof attempt in our preceding section, unusual care is needed when proving a "particular" limitation to a not-necessarily consistent agent, and our approach has been to bind into a single input (what we call a "specific halting problem") two opposite halting assertions. There could be other situations in which the $\mathcal{A}$ mentioned earlier in this paragraph could be reasonably said to "decide" that the corresponding true statement s of assertion (2) is true. For instance, suppose we define a 0 ("false") or 1 ("true") output of an agent to be a "decision" by that agent of the truth value of an input that is the code for a PA statement having the form $H[\overline{\#P}]$ or $\neg H[\overline{\#P}]$, rather than an input that is the code for what we call a "specific halting problem". Then it is possible that agent $\mathcal{A}$ could give output 1 for input that is the code for s. Nonetheless, we have the italicized definitive limitation on the ability of $\mathcal{A}$ to master a particular arithmetical truth.]

Assertion (2) implies that the particular true statement s of PA that agent $\mathcal{A}$ cannot master is an "absolute" truth, in the sense of "absolute" mentioned in our Introduction; that is, the truth of that statement holds—using Gödel's wording here—"in an absolute sense, without any further hypothesis", given the three widely-accepted assumptions mentioned in the Introduction. (Also, those three assumptions could be weakened somewhat and one still could obtain a result similar to the COAT Theorem. For instance, the assumption that the Church–Turing Thesis holds can be avoided by expressing "software" and "computational" in the COAT Theorem in terms of deterministic Turing machine concepts, as we have done in the proof of that theorem.)

We now discuss the final sentence of the theorem. As clarified near the beginning of the preceding section, our mathematical framework does not assume agents are infallible; for instance, it is possible within our framework that all of an agent's decisions could be incorrect, or that all but one of the agent's decisions could be incorrect, or that any number of an agent's decisions could be incorrect and any number of the agent's decisions could be correct. Recall from our footnote 5 that the official chess rules accommodate fallible chess players: a player who makes one illegal move in a chess match can still win the chess match, with the illegal move ignored (except there is a time penalty, and just one such illegal move is permitted). Our mathematical framework accommodates fallible agents even more than does chess, by ignoring any number of an agent's "illegal" outputs (i.e., outputs that are not in the defined acceptable *form* for an output). Thus, for any specific halting problem $\langle P, Halt? \rangle$ for which $\mathcal{A}$ actually gives output but simply does not format that output correctly, that output of $\mathcal{A}$ is ignored according to our mathematical framework. That is, our mathematical framework considers an agent to be what the preceding section defines as a "decision system". Thus, by the final statement of the theorem, if assertion (1) fails to hold, then a specific halting problem that is a particular example of the lack of mastery of $\mathcal{A}$ for arithmetical statements can be computationally determined from the code of the resulting simulation software.

Our intent has been to use the word "particular" in a special way in this article. That word appears in our brief explanation of Gödel's Second Incompleteness Theorem (which is why it appears in the conjecture at the beginning of the article, which is based indirectly on that theorem of Gödel), and that word also appears in the statement of the COAT Theorem. We now explain our intended meaning of that word. First, consider its use in our description of Gödel's Second Incompleteness Theorem, a theorem that does not merely say that there exists a true unprovable statement within a certain kind of formal system S (a fact already implied by Gödel's Incompleteness Theorem), it says there is a special such statement, Con_S. Based on mathematical logic, one can construct a computable function such that, when the function is given as its input a full mathematical description of any appropriate S satisfying the hypothesis of the Second Incompleteness Theorem, the function would produce as its output a statement Con_S for that system S. That is why we consider that formal statement to be "particular", rather than an arbitrarily chosen entity (once a function as described in the preceding sentence has been constructed) having the desired property of being true and unprovable. Next consider the use of "particular" in the COAT Theorem. Although Charlesworth

(2006) does not consider anything analogous to the COAT Theorem, it follows from the proof of its Lemma 7.15 that one can construct a computable function having the properties described in the final sentence of the COAT Theorem: a function that takes the code of any decision system counterexample of assertion (1)—if there is such a counterexample—as its input and produces as its output the code of an example of a specific halting problem that satisfies the description in assertion (2). That is why we consider such an example of assertion (2) to be "particular", rather than an arbitrarily chosen entity (once a function as described in the preceding sentence has been constructed) having the desired property of satisfying assertion (2). In addition to the fact that each of Gödel's Gibbs conjecture and the COAT Theorem provides an inclusive or related to computationalism and "absolute" truth, each identifies a particular limitation related to the mastery of "absolute" arithmetical truth.

Discussion of Specific Phrases in the Gibbs Conjecture

This section considers the use of the phrases "evident axioms", "infinitely surpasses", and "diophantine problem" in the statement of Gödel's Gibbs conjecture.

Gödel's use of the phrase "evident axioms"—that is, axioms that are evidently true—is problematic since logicians recognize the phrase lacks clarity. As J. R. Shoenfield pointed out in his book *Mathematical Logic*:

> An interesting question is whether every mathematical truth (or at least every mathematical truth expressible in $L(N)$) can be proved from axioms which are evidently true. However, we cannot hope to make much progress with this question until we understand more clearly what is meant by being evidently true. Shoenfield (1967), p. 133.

(Here $L(N)$ means the language of a specific system of formal arithmetic Shoenfield used in his book). The statement of the COAT Theorem avoids that problematic issue by avoiding the use of the phrase "evident axioms" or a similar word or phrase.

Gödel's phrase "infinitely surpasses" is problematic because that phrase can suggest a human or set of humans can carry out a process of reasoning about an infinite number of mathematical truths. [In another context Gödel has been quoted as saying "By *mind* I mean an individual mind of unlimited life span" Wang (2001), p. 189.] The statement of the COAT Theorem also avoids that problematic issue, by not requiring the agent it mentions to make decisions about infinitely many halting problems. It is possible that the total number of decisions such an agent makes could be infinite, but it is also possible that it could be any finite number, including zero.

Gödel's use of the phrase "diophantine problem" has the advantage of selecting a kind of arithmetical problem whose nature seems simple, at least on the surface. It is more difficult to explain the nature of the true arithmetical statement mentioned in assertion (2) of the COAT Theorem, to a person unfamiliar with the representation

within PA of assertions about Turing machines. Citing a kind of arithmetical problem whose nature seems simple to a broad audience is not essential if one's goal is a theorem about computationalism and "absolute" truth.

Summary

This article's single purpose is to seek to use mathematics to improve an argument about minds and machines given by Kurt Gödel, rather than to prove theorems sharply different from those of Charlesworth (2006, 2014), or to include a philosophical discussion of the extensive literature of commentary related to Gödel's Gibbs conjecture, including somewhat-related well-known arguments—that had a different purpose than Gödel's stated purpose—given by J. R. Lucas and Roger Penrose. (We would not suggest comparing this article with any of Gödel's articles, except to say that most of his articles also had strictly mathematical purposes. Although he mentioned the Richard Paradox and Liar Paradox [Gödel (1986), p. 149] in the groundbreaking article that stated his Second Incompleteness Theorem as its Theorem XI, that article included no discussion of the philosophical literature related to those paradoxes.)

The mathematical framework underlying the COAT Theorem avoids the standard simplification often made by logicians (when considering issues relevant to a topic like that of the present article) that Turing machines and formal logical systems are generally logically interchangeable with one another. Here is part of that standard simplification: "given any formal language L, any Turing machine M can be made to correspond to a formal system S in L, by extracting from the numbers it enumerates those that are the Gödel numbers of L, and taking their deductive closure to be the theorems of S." Feferman (2006), p. 138. It follows from that simplified way of associating a Turing machine with a formal system that "To say that the human mind—in its capacity as a producer of mathematical truths—is equivalent to a finite machine amounts ... to the same thing as saying that the set of humanly demonstrable theorems can be axiomatized by an effectively given formal system S." *ibid*, p. 7. Notice that the "deductive closure" requirement within that simplification implies an extremely strong infallibility assumption about human reasoning: humans would be assumed to have the capability of producing any of the theorems within such a formal system *S*; when *S* is PA there is no upper bound[13] on the length[14] of a required proof and humans would also be assumed never to produce a result that fails to be a (correct) theorem of the formal system *S*. In contrast, in the framework underlying the COAT Theorem, less than 1% of the decisions by an agent modeled by a Turing machine need to be correct decisions, and the agent can make blatantly inconsistent decisions; see the second paragraph of our section "Mathematical Framework". Yet, since there is no "deductive closure"

[13] Such an upper bound would imply a decision method for PA, contradicting Church's Theorem; see Shoenfield (1967) for a statement of Church's Theorem.

[14] A reminder: numbers themselves can be extremely large even when denoted using concise notation. Donald Knuth has suggested that "the magnitude of this number $10 \uparrow\uparrow\uparrow\uparrow 3$ is so large as to be beyond human comprehension"; see Knuth (1996), p. 1236.

requirement on such an agent, the agent need not make any additional incorrect decisions.

Avoiding the simplification discussed in the preceding paragraph is one key part of the path to the COAT Theorem. Our section "Mathematical Framework" begins by describing several ways the framework avoids infallibility assumptions about an agent's reasoning. The hypothesis of the COAT Theorem itself avoids infallibility assumptions. In contrast, as explained in our section "The Conjecture in Gödel's Gibbs Lecture", Gödel's argument in support of the Gibbs conjecture—based on his Second Incompleteness Theorem—crucially depends on the assumption that human reasoning is perfectly consistent, where we use the modifier "perfectly" to emphasize that the consistency must be without exception. Thus, by avoiding assumptions about the infallibility of the agent mentioned in its statement, the COAT Theorem is a new result that differs sharply from the Gibbs conjecture.

Gödel's Gibbs conjecture uses the phrase "absolutely unsolvable", yet his argument in support of his Gibbs conjecture assumes that the only way humans make decisions about mathematical and logical questions is to use deductive reasoning. That restrictive assumption is not made in the context of the COAT Theorem, which permits but does not require a formal proof to accompany an agent's decision. Cognitive scientists study a variety of ways that humans reason. The book *How We Reason* summarizes that variety, including deduction, induction, and abduction Johnson-Laird (2006). The part of that book entitled *How We Make Deductions* summarizes scientific evidence from that author's experiments that suggest humans often employ what are termed "mental models" when given questions related to predicate logic, rather than employing what logicians would term "deductions".

In addition to the above distinctions between the COAT Theorem and the Gibbs conjecture, the Gibbs conjecture does not qualify for being the statement of a theorem. The critique of Gödel's Gibbs lecture in Feferman (2006) points out several ways the Gibbs conjecture fails to be a mathematical theorem, largely due to the lack of mathematical definitions for several of its concepts: "the human mind", "evident axioms", and "infinitely surpasses". Although Gödel judged the Gibbs conjecture sufficiently important to become a main focus of his 1951 highly-visible invited lecture to the American Mathematical Society, and although he published a few articles between then and his death in 1978, he never published that conjecture. A possible explanation is his well-known preference for publishing theorems rather than imprecise conjectures,[15] and that he did not find a mathematical way to formulate and prove something more like his Gibbs conjecture than his Second Incompleteness Theorem. Another possibility is that he came to realize, and was unable to overcome, what many others would view as the highly questionable use of an idealization of humans within his argument. His Gibbs lecture was only published posthumously, in the part of his Collected Works containing his unpublished papers Gödel (1995).

Among other things, the COAT Theorem answers the following questions: Can a mathematical *theorem* related to the Gibbs conjecture be obtained? Can the hypothesis of such a theorem avoid *assuming* an agent must have strong infallibility

[15] Recall the Wang quote mentioning Gödel's concern for "clarity and certainty", in our section "The Conjecture in Gödel's Gibbs Lecture".

properties? Can the hypothesis of such a theorem avoid *assuming* that the reasoning by the agent mentioned in the theorem must be deductive? When an agent does use deductive reasoning, can the hypothesis of such a theorem avoid *assuming* that the formal theorems within the formal system the agent uses must be computably enumerable (which would imply there is a single Turing machine that is at least as successful making and correctly justifying mathematical decisions as all other possible agents, including humans)? Are phrases like "human mind" and "evident axioms" and "infinitely surpasses" necessary in such a result? The answers to these questions are yes, yes, yes, yes, and no, respectively.

The COAT Theorem also raises questions. Many questions about the appropriateness of the mathematical framework we use are answered in Charlesworth (2014). For example, those explanations indicate how one can view each "rigorous" (according to the current standard criterion for rigor in mathematics) conjecture as a halting problem without confusing the formalization of a problem (within a Kleene-like hierarchy of problems) with the formalization of provability of a problem in Zermelo–Fraenkel Set Theory.

The degree of belief in each of the two assertions in the inclusive-or COAT Theorem might vary from person to person, when the theorem is applied to any specific agent $\mathcal{A}$. The COAT Theorem essentially implies there should be a rational constraint on those two degrees of belief for any single person.[16]

Many who might be interested in the relevant issues may not have taken an opinion position related to the assertions in Gödel's Gibbs conjecture since they gave that conjecture little thought because that conjecture was clearly not a theorem or because Gödel's argument in support of it crucially depends on an infallibility assumption or because his argument crucially assumes that reasoning about arithmetical truths must be purely deductive. Will that change, now that there is a mathematical theorem that avoids those assumptions? Will others be stimulated to provide fresh reasons to justify such opinions?

Finally, will the COAT Theorem—and its demonstration of how a mathematical model in terms of Turing machines has an advantage over a mathematical model in terms of formal logical systems when seeking such a theorem and its illustration of the relevance a simple mathematical function can have to computationalism—stimulate others to seek ways to improve upon it?

Acknowledgments The author thanks Editor-in-Chief Gregory Wheeler for helpful advice. The author is also grateful for questions asked by the anonymous referees, which improved the clarity of the article.

[16] A metaphor might help. First, recall that according to one established interpretation, a probability represents a "degree of belief" in an assertion, Russell and Norvig (2010), p. 504. The alternative terminology "plausibility" is used in Jaynes (2003). The COAT Theorem implies that one's degree of belief in its first assertion should be constrained by one's degree of belief in its second assertion and vice versa, since the sum of the two degrees of belief (each of which is a real number from 0.0 to 1.0 inclusive) should be from 1.0 to 2.0 inclusive. If considering its first assertion highly plausible corresponds to shining a bright red light upon a white lab coat and its second assertion highly plausible as shining a bright green light, then both assertions highly plausible corresponds to shining a very bright golden-yellow light, due to the additive nature of two colored light beams. That gives a metaphorical "coat of many colors", in the sense that people having differing opinions about the plausibility of one or both of the two assertions would have differing views of the color of the coat.

References

Barrow, J. D. (1998). *Impossibility: The limits of science and the science of limits*. New York: Oxford University Press.

Charlesworth, A. (2006). Comprehending software correctness implies comprehending an intelligence-related limitation. *ACM Transactions on Computational Logic*, *7*, 590–612.

Charlesworth, A. (2014). The comprehensibility theorem and the foundations of artificial intelligence. *Minds and Machines*, *24*, 439–476.

Davis, M. (1990). Is mathematical insight algorithmic? *Behavioral and Brain Sciences*, *13*(4), 659–660.

Dawson, J. (1997). *Logical dilemmas: The life and work of Kurt Gödel*. Wellesley, MA: A. K. Peters.

Dennett, D. (1990). Betting your life on an algorithm. *Behavioral and Brain Sciences*, *13*(4), 660–661.

Dodd, T. (1991). Gödel, Penrose and the possibility of AI. *Artificial Intelligence Review, 5*, 187–199.

Feferman, S. (2006). Are there absolutely unsolvable problems? Gödel's dichotomy. *Philosophia Mathematica*, *14*(2), 134–152.

FIDE 2014. World Chess Federation Handbook, www.fide.com/fide/handbook.html.

Fraenkel, A. S., & Lichtenstein, D. (1981). Computing a perfect strategy for $n \times n$ chess requires time exponential in n. *Journal of Combinatorial Theory, Series A*, *31*, 199–214.

Gödel, K. (1986). On formally undecidable propositions of *Principia Mathematica* and related systems. In S. Feferman (Ed.), *Kurt Gödel's collected works* (Vol. I, pp. 145–195). New York: Oxford University Press.

Gödel, K. (1995). Some basic theorems on the foundations of mathematics and their implications. In S. Feferman (Ed.), *Kurt Gödel's collected works* (Vol. III, pp. 304–323). New York: Oxford University Press.

Hodel, R. E. (1995). *An introduction to mathematical logic*. Boston: PWS Publishing Company.

Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2007). *Introduction to automata theory, languages, and computation* (3rd ed.). Boston: Addison Wesley.

Jaynes, E. T. (2003). *Probability theory: The logic of science*. New York: Cambridge University Press.

Johnson-Laird, P. N. (2006). *How we reason*. New York: Oxford University Press.

Kane, R. H. (2011). *The Oxford handbook of free will* (2nd ed.). New York: Oxford University Press.

Knuth, D. E. (1996). *Selected papers on computer science*. New York: Cambridge University Press.

Libet, B., Gleason, C. A., Wright, E. W., & Pearl, D. K. (1983). Time of conscious intention to act in relation to onset of cerebral activity (readiness potential): The unconscious initiation of a freely voluntary act. *Brain*, *106*, 623–642.

Lutz, R. (1990). Quantum AI. *Behavioral and Brain Sciences*, *13*(4), 672–673.

McDermott, D. (1990). Computation and consciousness. *Behavioral and Brain Sciences*, *13*(4), 676–677.

Minsky, M. (1995). Reply to Penrose's "Consciousness involves noncomputable ingredients". In J. Brockman (Ed.), *The third culture* (pp. 256–257). New York: Simon and Schuster.

Perlis, D. (1990). The emperor's old hat. *Behavioral and Brain Sciences*, *13*(4), 680–681.

Roskies, A. (1990). Seeing truth or just seeming true? *Behavioral and Brain Sciences*, *13*(4), 682–683.

Russell, S. J., & Norvig, P. (2010). *Artificial intelligence: A modern approach* (3rd ed.). New Jersey, Englewood Cliffs: Prentice-Hall.

Shoenfield, J. R. (1967). *Mathematical logic*. Reading, MA: Addison-Wesley.

Tieszen, R. (2006). After Gödel: Mechanism, reason, and realism in the philosophy of mathematics. *Philosophia Mathematica*, *14*(2), 229–254.

Turing, A. M. (1986). Lecture to the London mathematical society on 20 February 1947. In A. M. Turing's ACE report of 1946 and other papers, (pp. 106–124). Cambridge, MA: MIT Press.

Wang, H. (2001). *A logical journey*. Cambridge: MIT Press.