

# Is the Halting probability a Dedekind real number?

Bhupinder Singh Anand<sup>1</sup>

*(A .pdf file of this essay before the current update is available at <http://arXiv.org/abs/math/0306023> and at <http://www.mathpreprints.com/math/Preprint/anandb/20030521/1>)*

In a historical overview, Cristian S. Calude, Elena Calude, and Solomon Marcus identify eight stages in the development of the concept of a mathematical proof in support of an ambitious conjecture: we can express classical mathematical concepts adequately only in a mathematical language in which both truth and provability are essentially unverifiable. In this essay we show, first, that the concepts underlying their thesis can, however, be interpreted constructively; and, second, that an implicit thesis in the authors' arguments implies that the Halting problem is solvable, but that, despite this, the probability of a given Turing machine halting on a random input cannot be assumed to define a Dedekind real number.

## Contents

1. Introduction
  - 1.1 What is proof?
2. Interpreting classical mathematical theory
  - 2.1 Standard interpretations of foundational concepts may be ambiguous
  - 2.2 Can classical concepts be defined constructively?
  - 2.3 Standard interpretations may admit ambiguity
  - 2.4 Reducing Tarskian truth and satisfiability to provability

---

<sup>1</sup> The author is an independent scholar. E-mail: [re@alixcomsi.com](mailto:re@alixcomsi.com); [anandb@vsnl.com](mailto:anandb@vsnl.com). Postal address: 32, Agarwal House, D Road, Churchgate, Mumbai - 400 020, INDIA. Tel: +91 (22) 2281 3353. Fax: +91 (22) 2209 5091.

- 2.5 Some consequences of a constructive interpretation
- 2.6 Defining formal, constructive and Platonic concepts
- 3. Mathematical proof and non-algorithmic effective methods
  - 3.1 Non-algorithmic effective methods: Gödel oracles
  - 3.2 Defining Tarskian truth verifiably
- 4. The Halting probability
  - 4.1 CCM's Thesis and the Halting problem
  - 4.2 An effective solution of the Halting problem
  - 4.3 Is the Halting probability a Dedekind real number?
  - 4.4 Standard interpretations of the significance of the Halting probability
- 5. Are mechanistic proofs of mathematical problems logically sound?

## 1. Introduction

In an arXived paper ([Ca01], v2), “Passages of Proof”, Cristian S. Calude, Elena Calude, and Solomon Marcus conjecture that:

*Reason and experiment* are two ways to acquire knowledge. For a long time mathematical proofs required only reason; this might be no longer true.

As the basis for their belief, they identify eight stages in the development of the concept of a mathematical proof:

- (a) The first period was that of pre-Greek mathematics, for instance the Babylonian one, dominated by observation, intuition and experience.
- (b) The second period was started by Greeks such as Pythagoras and is characterised by the discovery of deductive mathematics, based on theorems.
- (c) ... with Galilei, Descartes, Newton and Leibniz, the mathematical language became more and more a mixed language, characterized by a balance between its natural and artificial components. ... This was the third step in the development of mathematical proofs.

- (d) The fourth step is associated with the so-called epsilon rigour, so important in mathematical analysis; it occurred in the XIXth century and it is associated with names such as A. Cauchy and K. Weierstrass.
- (e) The fifth period begun with the end of the 19<sup>th</sup> century, when Aristotle's logic, underlining mathematical proofs for two thousands years, entered a crisis with the challenge of the principle of non-contradiction.
- (f) The sixth period begins with Godel's incompleteness theorem (1931), for many meaning the unavoidable failure of any attempt to formalise the whole mathematics.
- (g) The seventh period belongs to the second half of the 20<sup>th</sup> century, when algorithmic proofs become acceptable only when their complexities were not too high.
- (h) With the eighth stage, proofs are no longer exclusively based on logic and deduction, but also empirical and experimental factors.

## 1.1 What is proof?

The authors then ask, rhetorically, ([Ca01], v2):

What is a mathematical proof ? At a first glance the answer seems obvious: a proof is a series of logical steps based on some axioms and deduction rules which reaches a desired conclusion. Every step in a proof can be checked for correctness by examining it to ensure that it is logically sound. In David Hilbert's words: "The rules should be so clear, that if somebody gives you what they claim is a proof, there is a mechanical procedure that will check whether the proof is correct or not, whether it obeys the rules or not."

They note, however, that:

In 1976, Kenneth Appel and Wolfgang Haken proved the 4CT (*Four Colour Theorem*) ... No human being could ever actually read the entire proof to check its correctness ... “The real question is this: If no human being can ever hope to check a proof, is it really a proof ?”

The authors’ perception of the relation between truth and provability is reflected in their comments:

... Gödel’s incompleteness theorem (GIT) which says that every formal system which is (1) finitely specified, (2) rich enough to include the arithmetic, and (3) consistent, is incomplete<sup>2</sup>. That is, there exists an arithmetical statement which (A) can be expressed in the formal system, (B) is true, but (C) is unprovable within the formal system. ... But what does it mean to be a “true arithmetical statement”? It is a statement about non-negative integers which cannot be invalidated by finding any combination of non-negative integers that contradicts it. ... a true arithmetical statement is a “primordial mathematical reality”. ... The essence of GIT is to distinguish between truth and provability. A closer analogy in real life is the distinction between truths and judicial decisions, between what is true and what can be proved in court. How large is the set of true and unprovable statements? If we fix a formal system satisfying all three conditions in GIT, then the set of true and unprovable statements is topologically “large” (constructively, a set of second Baire category, and in some cases even “larger”).

---

<sup>2</sup> We note that this is an instance of the standard interpretation of Gödel’s seminal 1931 paper [Go31a] that may be arguably definitive; if we apply the standard Deduction Theorem of first order logic to Gödel’s meta-proof of his Theorem VI in the cited paper, then we can, reasonably, argue that his formal system P is omega-inconsistent. Theorem VI would, in such case, hold vacuously, and the incompleteness of P would not, then, be a consequence.

Prima facie, under the standard interpretations of classical<sup>3</sup> mathematical theory, which the authors seem to implicitly assume ([Ca01], v2), the above remarks can be taken to imply that the authors accept mathematical truth as being unverifiable effectively. It follows that there could, then, be any number of (equally reasonable) ways of responding to their question:

... what does it mean to be a “true arithmetical statement”?

However, in their attempt to offer an ambitious interpretation of classical theory, the authors do not address the question:

Can such latitude in the perception of fundamental meta-mathematical concepts such as truth and provability reflect a basic ambiguity in our definitions of foundational mathematical concepts?

On the contrary, the authors seem to be comfortable with the, implicitly Platonic, suggestion that classical concepts of mathematical proof, and even truth, might actually lie beyond the ambit of direct intuitive cognition! They conclude that ([Ca01], v2):

If we accept the above assumptions about the biological and physical nature of proofs, then there is little ‘intrinsic’ difference between traditional and ‘unconventional’ types of proofs as i) first and foremost, we have not access to truth, ii) correctness is not absolute, but nearly certain as mathematics advances by making mistakes and correcting and re-correcting them ..., iii) non-deterministic and probabilistic proofs do not allow mistakes in the applications of rules, they are just indirect forms of checking ... which correspond to various degrees of rigour, iv) the explanatory component, the understanding ‘generated’ by proofs, while extremely

---

<sup>3</sup> For the purposes of this essay, we take the expositions by Hardy [Ha47], Landau [La51], Mendelson [Me64], Rudin [Ru53] and Titchmarsh [Ti61] as representative, in the areas that they cover, of classical mathematical reasoning and conclusions.

important from a cognitive point of view, is subjective and has no bearing on formal correctness.

... more research will be performed in large computational environments where we might or might not be able to determine what the system has done or why ... The blend of logical and empirical-experimental arguments are here to stay and develop. ... There are many reasons which support this prediction. They range from economical ones (powerful computers will be more and more accessible to more and more people), social ones (sceptical oldsters are replaced naturally by youngsters born with the new technology, results and success inspire emulation) to pure mathematical (new challenging problems, wider perspective) and philosophical ones (note that incompleteness is based on the analysis of the computer's behaviour).

## **2. Interpreting classical mathematical theory**

### **2.1 Standard interpretations of foundational concepts may be ambiguous**

Now, we note that, as is implicit in Mendelson's [Me90] following remarks (*italicised parenthetical qualifications added*), standard interpretations of classical foundational concepts can, indeed, be argued as being either ambiguous, or non-constructive, or both:

Here is the main conclusion I wish to draw: it is completely unwarranted to say that CT (*Church's Thesis*) is unprovable just because it states an equivalence between a vague, imprecise notion (effectively computable function) and a precise mathematical notion (partial-recursive function). ... The concepts and assumptions that support the notion of partial-recursive function are, in an essential way, no less vague and imprecise (*non-constructive, and intuitionistically objectionable*) than the notion of effectively computable function; the former are just more familiar and are part of a respectable theory with connections to other parts of logic and mathematics. (The

notion of effectively computable function could have been incorporated into an axiomatic presentation of classical mathematics, but the acceptance of CT made this unnecessary.) ... Functions are defined in terms of sets, but the concept of set is no clearer (*not more non-constructive, and intuitionistically objectionable*) than that of function and a foundation of mathematics can be based on a theory using function as primitive notion instead of set. Tarski's definition of truth is formulated in set-theoretic terms, but the notion of set is no clearer (*not more non-constructive, and intuitionistically objectionable*), than that of truth. The model-theoretic definition of logical validity is based ultimately on set theory, the foundations of which are no clearer (*not more non-constructive, and intuitionistically objectionable*) than our intuitive (*non-constructive, and intuitionistically objectionable*) understanding of logical validity. ... The notion of Turing-computable function is no clearer (*not more non-constructive, and intuitionistically objectionable*) than, nor more mathematically useful (foundationally speaking) than, the notion of an effectively computable function.

The questions thus arise: Could the thesis conjectured in ([Ca01], v2) also be founded on ambiguities that are rooted in the standard interpretations of classical foundational concepts such as “mathematical object”, “effective computability”, “truth of a formula under an interpretation”, “set”, “Church’s Thesis” etc.; ambiguities that may, moreover, encourage non-constructive, Platonic, interpretations by default? How would such a thesis fare if we could make these concepts unambiguous, and constructive, in an intuitionistically unobjectionable way?

## 2.2 Can classical concepts be defined constructively?

Now, *prima facie*, we can, indeed, define these concepts constructively in terms of a small number of primitive, formally undefined but intuitively unobjectionable, mathematical terms as below:

(i) **Primitive mathematical object:** A primitive mathematical object is any symbol for an individual constant, predicate letter, or a function letter, which is defined as a primitive symbol of a formal mathematical language.<sup>4</sup>

(ii) **Formal mathematical object:** A formal mathematical object is any symbol for an individual constant, predicate letter, or a function letter that is either a primitive mathematical object, or that can be introduced through definition into a formal mathematical language without inviting inconsistency.

(iii) **Mathematical object:** A mathematical object is any symbol that is either a primitive mathematical object, or a formal mathematical object.

(iv) **Set:** A set is the range of any function whose function letter is a mathematical object.

(v) **Instantiational computability:** A number-theoretic function  $F(x)$  is instantiationally computable if, and only if, given any natural number  $k$ , there is always an effective method (which may depend on the value  $k$ ) to compute  $F(k)$ .

---

<sup>4</sup> We note that, as remarked by Mendelson [Me90], the terms “function” and “function letter” - and, presumably, “individual constant”, “predicate”, and “predicate letter” - can be taken as undefined, primitive foundational concepts.

(vi) **Algorithmic computability:** A number-theoretic function  $F(x)$  is algorithmically computable if, and only if, there is an effective method (necessarily independent of  $x$ ) such that, given any natural number  $k$ , it can compute  $F(k)$ .

(vii) **Effective computability:** A number-theoretic function is effectively computable if, and only if, it is either instantiationally computable, or it is algorithmically computable.<sup>5</sup>

(viii) **Instantiationally true:** A string  $[F(x)]$ <sup>6</sup> of a formal system  $P$  is instantiationally true under an interpretation  $M$  of  $P$  if, and only if, given any value  $k$  in  $M$ , there is an effective method (which may depend on the value  $k$ ) to determine that the interpreted proposition  $F(k)$  is satisfied in  $M$ .<sup>7</sup>

(ix) **Algorithmically true:** A string  $[F(x)]$  of a formal system  $P$  is algorithmically true under an interpretation  $M$  of  $P$  if, and only if, there is an effective method (necessarily independent of  $x$ ) such that, given any value  $k$  in  $M$ , it can determine that the interpreted proposition  $F(k)$  is satisfied in  $M$ .

(x) **Effectively true:** A string  $[F(x)]$  of a formal system  $P$  is effectively true under an interpretation  $M$  of  $P$  if, and only if, it is either instantiationally true in  $M$ , or it is algorithmically true in  $M$ .<sup>8</sup>

---

<sup>5</sup> We note that classical definitions of the effective computability of a function (cf. [Me64], p207) do not distinguish between the two cases.

<sup>6</sup> We use square brackets to distinguish between the uninterpreted string  $[F]$  of a formal system, and the symbolic expression “ $F$ ” that corresponds to it under a given interpretation that unambiguously assigns formal, or intuitive, meanings to each individual symbol of the expression “ $F$ ”.

<sup>7</sup> Under a constructive interpretation of formal Peano Arithmetic, Gödel’s undecidable proposition may, thus, be instantiationally, but not algorithmically, true under the standard interpretation.

<sup>8</sup> We note that, classically, Tarski’s definition of the truth of a formal proposition under an interpretation (cf. [Me64], p49-52) does not distinguish between the two cases.

(xi) **Instantiational Church Thesis:** If, for a given relation  $R(x)$ , and any element  $k$  in some interpretation  $M$  of a formal system  $P$ , there is an effective method such that it will determine whether  $R(k)$  holds in  $M$  or not, then every element of the domain  $D$  of  $M$  is the interpretation of some term of  $P$ , and there is some  $P$ -formula  $[R'(x)]$  such that:

$R(k)$  holds in  $M$  if, and only if,  $[R'(k)]$  is  $P$ -provable.

(In other words, the Instantiational Church Thesis postulates that, if a relation  $R$  is effectively decidable instantiationally (possibly non-algorithmically) in an interpretation  $M$  of some formal system  $P$ , then  $R$  is expressible in  $P$ , and its domain necessarily consists of only mathematical objects, even if the predicate letter  $R$  is not, itself, a mathematical object.)

(xii) **Algorithmic Church Thesis:** If, in some interpretation  $M$  of a formal system  $P$ , there is an effective method such that, for a given relation  $R(x)$ , and any element  $k$  in  $M$ , it will determine whether  $R(k)$  holds in  $M$  or not, then  $R(x)$  is the interpretation in  $M$  of a  $P$ -formula  $[R(x)]$ , and:

$R(k)$  holds in  $M$  if, and only if,  $[R(k)]$  is  $P$ -provable.

(Thus, the Algorithmic Church Thesis postulates that, if a relation  $R$  is effectively decidable algorithmically in an interpretation  $M$  of a formal system  $P$ , then, first,  $R$  is expressible in  $P$ , and, second, the predicate letter  $R$ , and all the elements in the domain of the relation  $R$ , are necessarily mathematical objects.)

### 2.3 Standard interpretations may admit ambiguity

Moreover, the vagueness, and implicitly implied non-constructivity, alluded to in Mendelson's remarks may simply reflect, and result from, the non-specification of an

effective method for determining that the infinity of intuitive assertions, which are implicit in Tarski's definition of the truth of a formula of a formal system under an interpretation, are, indeed, instantiationally verifiable.

Thus, Tarski's definitions may be seen as implicitly implying, first, that relationships may exist only Platonically between the abstract elements of the domain of some model  $M$ , since there is no assurance that every such element is constructively definable, or representable, in every model  $M$  of  $PA^9$ ; and, second, that even when such relationships, in some cases, are asserted as holding in  $M$  intuitively, this may not be in any effectively verifiable manner.

So, what we see here could, arguably, be the thin end of the wedge that keeps the door ajar for the entry of non-constructive, Platonic, elements into the standard interpretations of classical theories; elements that can then be interpreted ambiguously - often creating a mathematical tower of Babel containing frustrated purists, and confused neophytes!

#### **2.4 Reducing Tarskian truth and satisfiability to provability**

If we, therefore, introduce the concept of effective truth, based on effective methods of verification as suggested above, then we may effectively reduce any verifiable truth in the model  $M$  to provability in  $PA$ .

In other words, what such constructive definitions and theses essentially suggest is that, in order to make Tarski's definitions of truth and satisfiability effectively verifiable in any model  $M$  of  $PA$ , we should be able to argue that:

---

<sup>9</sup> Standard first order Peano Arithmetic such as Mendelson's formal system  $S$  ([Me64], p102).

(i) If a string  $[R(x)]$  is PA-provable, then its interpretation  $R(x)$  in  $M$  is *both* instantiationally true *and* algorithmically true; hence, viewed as a Boolean function,  $R(x)$  is Turing-computable.

(ii) If a string  $[R(n)]$  is PA-provable for any given numeral  $[n]$ , then the interpretation  $R(x)$  in  $M$  is *either* instantiationally true *or* algorithmically true; hence, viewed as a Boolean function,  $R(x)$  is not necessarily Turing-computable.

(iii) If  $R(x)$  is instantiationally true in  $M$  (which we may express as  $(\exists x)R(x)$ ), then,  $R(x)$  is expressible (cf. [Me64], p117, §2) in PA; hence every element of the domain of  $M$  is the interpretation of some term of PA.

(iv) *Algorithmic Turing Thesis*<sup>10</sup>: If  $R(x)$  is algorithmically true in  $M$  (which we may express as  $(\forall x)R(x)$ ), then  $[R(x)]$  is PA-provable; hence, viewed as a Boolean function,  $R(x)$  is Turing-computable.

Now, whilst (i), and (ii), seem, *prima facie*, consistent with standard interpretations of Tarski's definitions, (iii) clearly does not follow from them; however, as the Löwenheim-Skolem theorem ([Me64], p69, Corollary 2.16) suggests, it may not be inconsistent with such interpretations. It is not obvious whether (iv) is independent of, equivalent to, or a consequence of the Instantiation and Algorithmic Church Theses.

## 2.5 Some consequences of a constructive interpretation

The significance of constructively interpreting foundational concepts and assertions of classical mathematics is that:

---

<sup>10</sup> We note that, by reasoning that lies outside the immediate scope of this essay, introduction of an equivalent statement of this thesis, as an independent Quantum Halting Hypothesis, allows us to model a deterministic universe that is essentially unpredictable.

(i) The Algorithmic Church Thesis implies that a formula  $[R]$  is P-provable if, and only if,  $[R]$  is algorithmically true in some interpretation M of P.

(ii) The Algorithmic Church Thesis implies that if a number-theoretic relation  $R(x)$  is algorithmically satisfied in some interpretation M of P, then the predicate letter “ $R$ ” is a formal mathematical object in P (i.e. it can be introduced through definition into P without inviting inconsistency).

(iii) The Algorithmic Church Thesis implies that, if a P-formula  $[R]$  is algorithmically true in some interpretation M of P, then  $[R]$  is algorithmically true in every model of P.

(iv) The Algorithmic Church Thesis implies that if a formula  $[R]$  is not P-provable, but  $[R]$  is classically true under the standard interpretation, then  $[R]$  is instantiationally true, but not algorithmically true, in the standard model of P.

(v) The Algorithmic Church Thesis implies that Gödel’s undecidable sentence GUS is instantiationally true, but not algorithmically true, in the standard model of P.

By defining effective computability, both instantiationally and algorithmically, along similar lines, we can, further, give a constructive definition of uncomputable number-theoretic functions:

(vi) A number-theoretic function  $F(x_1, \dots, x_n)$  in the standard interpretation M of P is uncomputable if, and only if, it is effectively computable instantiationally, but not effectively computable algorithmically.

This, last, removes the mysticism behind the fact that we can constructively define a number-theoretic Halting function that is, paradoxically, Turing-uncomputable.

(vii) If we assume an Algorithmic Church Thesis, then every partial recursive number-theoretic function  $F(x_1, \dots, x_n)$  has a unique constructive extension as a total function.

(viii) If we assume an Algorithmic Church Thesis, then the classical Halting problem is effectively solvable.

(ix) If we assume an Algorithmic Church Thesis, then not every effectively computable function is classically Turing computable (so Turing's Thesis does not, then, hold).

(x) If we assume an Algorithmic Church Thesis, then the class P of polynomial-time languages in the P versus NP problem may not define a formal mathematical object.

Further, since a number-theoretic relation is expressible in P if, and only if, it is recursive ([Me64], p142, Corollary 3.29), it follows, as Mendelson argues, that the classical Church's Thesis can, indeed, be viewed as:

(xi) *Church's Theorem*: The Individual Church Thesis implies that a number-theoretic function is effectively computable if, and only if, it is recursive<sup>11</sup>.

## 2.6 Defining formal, constructive and Platonic concepts

Now, what we have essentially argued above is that a constructive interpretation, of classical mathematical concepts, suggests that not every, effectively well-defined, classical, mathematical concept is necessarily formalisable in its intuitive entirety.

---

<sup>11</sup> We note that the classical Church Thesis is the assertion: "A number-theoretic function is effectively computable if, and only if, it is recursive" (cf. [Me64], p227).

Hence, we do not need to treat every non-formalisable mathematical concept as necessarily Platonic, and so outside the reach of effective methods. Ipso facto, we can distinguish between:

- (i) number-theoretic functions and relations that are instantiationally, but not algorithmically, computable / verifiable;
- (ii) number-theoretic and relations that are always algorithmically computable / verifiable;
- (iii) mathematical concepts that are essentially unverifiable by any effective method, and which implicitly appeal to a non-constructive Platonic oracle for asserting that they are computable / verifiable.

### **3. Mathematical proof and non-algorithmic effective methods**

Accordingly, a more appropriate definition of mathematics and mathematical proof - which can be seen, *prima facie*, as a unifying thread in all the eight stages sought to be identified as distinct by the authors in ([Ca01], v2) - would be:

Mathematics is a language where proof is the yardstick for unambiguous expression and communication.

#### **3.1 Non-algorithmic effective methods: Gödel oracles**

To see the *raison d'être*, and possible significance, of such a paradigm shift - from seeing mathematics as an expression of relations between a universe of abstract objects, to viewing it solely as a language of unambiguous communication - we note that, broadly speaking, Gödel's meta-reasoning (which is constructive and intuitionistically unobjectionable), actually establishes the existence of an undecidable sentence by means of an effective, and not simply theoretical, Turing-oracle.

If we ignore the details of his proof, what Gödel's oracle does is to effectively prove, meta-mathematically, that Gödel's undecidable arithmetical relation  $R(x)$ <sup>12</sup> is such that, for any given natural number  $n$ :

(i) PA proves:  $[R(n)]$ ,<sup>13</sup>

but:

(ii) PA does not prove:  $[(\forall x)R(x)]$ .<sup>14</sup>

Thus Gödel's oracle effectively establishes, by meta-lemma<sup>15</sup> (i), above, that, given any natural number  $n$ , we can always find some proof (which can be converted into an effective method) that the string  $[R(n)]$  is PA-provable (even though it gives no clue as to how we should go about finding such a proof in any, individual, case).

Hence, the arithmetical sentence  $R(n)$  can be effectively asserted as true, for any natural number  $n$ , by Tarski's definition of the “truth” of the arithmetical predicate  $R(x)$  under the standard interpretation M of PA.

However, because of meta-lemma (ii), there is no effective method for determining a PA-proof of the string  $[(\forall x)R(x)]$ , since such a proof does not exist. Hence, there is no guarantee of an algorithm such that, given any natural number  $n$ , it will determine that the sentence  $R(n)$  is true under the standard interpretation M. Such an algorithm would, of course, be guaranteed if  $[(\forall x)R(x)]$  were PA-provable.

---

<sup>12</sup> For “relation  $R(x)$ ”, read “proposition  $(\forall x)R(x)$ ”.

<sup>13</sup> This is actually an implicit meta-lemma in Theorem VI of Gödel's seminal 1931 paper [Go31a].

<sup>14</sup> This, too, is an implicit meta-lemma in Theorem VI of Gödel's seminal 1931 paper [Go31a].

<sup>15</sup> For “meta-theorem”, read “meta-lemma” in this section.

The intended thesis, here, is, thus, that meta-lemma (ii), in fact, implies that there is no algorithm such that, given any natural number  $n$ , it will determine that the sentence  $R(n)$  is true under the standard interpretation.

In other words, we may replace the *classical Turing Thesis*:

(iii) A number-theoretic predicate  $F(x)$ , viewed as a Boolean function, is effectively computable if, and only if, it is classically Turing-computable,

with, for instance, an *Arithmetic Provability Thesis*:

(iv) A PA-string  $[(\forall x)F(x)]$  is provable if, and only if, in its standard interpretation  $M$ , the predicate  $F(x)$ , viewed as a Boolean function, is classically Turing-computable as true for every input.

Now, like the classical Turing Thesis, the Arithmetic Provability Thesis cannot be proven. However, unlike the classical Turing Thesis, this thesis can never be disproved.

The reason: The unprovability of the PA-string  $[(\forall x)R(x)]$  means that we could never formally prove that a Turing machine, (algorithm)  $T(R)$ , that computes the arithmetical predicate  $R(x)$ , when seen as a Boolean function, will halt and return true on every input. In fact, in view of meta-lemmas (i) and (ii), the Arithmetic Provability Thesis implies that any such  $T(R)$  will loop for some natural number  $k$ , even though  $R(k)$  is true.

The significance of this is that an Arithmetic Provability Thesis also implies that the standard Turing Thesis does not hold, since Gödel's oracle is, then, an effective (non-algorithmic) method that is not classically Turing-computable.

By Occam's dictum, and since there is no loss of generality in replacing the classical Turing Thesis with an Arithmetic Provability Thesis, the replacement is to be preferred.

The above is, essentially, the argument for introducing constructive Gödel oracles, so that we may extend the scope of effective methods to include non-algorithmic effective methods by means of an Algorithmic Turing Thesis.

### 3.2 Defining Tarskian truth verifiably

The significance of replacing the classical Church Thesis by Instantiational and Algorithmic Church Theses lies in the fact that, without these, there is no explicit convention for asserting Tarskian truth unambiguously.

For instance, one reason why Turing oracles may, so far, be viewed as offering theoretical, rather than effective, solutions to constructive problems of Theoretical Computer Science (such as, say, the Halting problem), could be that they are deeply wedded to standard interpretations of classical theory that, in turn, are rooted in Tarski's definitions, such as:

(i) The PA-string  $[(Ax)F(x)]$  is true under an interpretation  $M$  if, and only if,  $F(x)$  is satisfied by every element  $s$  of  $M$  (i.e.  $F(s)$  holds in  $M$ ).

Now, the three significant points to note here are that Tarski implicitly assumes:

(ii) there is some language, accessible to us, in which  $F(s)$  is expressible for any  $s$  of  $M$ ;

(iii) this is a language in which we can verify that  $F(s)$  holds for any  $s$  of  $M$ ;

(iv) there is some effective method to verify that  $F(s)$  holds for any element of  $M$ .

Hence, if we seriously intend to make our mathematical language precise and unambiguous, then:

(v) we should introduce these assumptions explicitly as premises in Tarski's definitions;

(vi) we should specify what we mean by an effective method that can verify that  $F(s)$  holds for any element  $s$  of  $M$ .

Moreover, if we are not uncomfortable with accepting the Church-Turing Thesis (which is widely assumed to hold), then it is reasonable to assume that:

(vii) any effective method must be an instantiationally effective method;

(viii) we can only verify that  $F(s)$  holds in  $M$  by an effective method if every  $s$  in  $M$  is the interpretation of some symbol  $[s]$  of PA.

This is, essentially, the reasoning behind the introduction of constructive definitions of classical concepts such as “mathematical object”, “instantiationally / algorithmic effective methods”, “instantiationally true/computable predicates/functions”, “Instantiationally / Algorithmic Church Thesis”, etc.

Prima facie, it seems that the Algorithmic Turing Thesis, as expressed earlier, is actually a theorem that follows from the introduction of these definitions. However, it may, in fact, be an additional thesis, and it is not immediately obvious whether the Instantiationally / Algorithmic Church Theses should be treated as definitions or theses.

Now, we note that, if we leave (ii), (iii), and (iv) as implicit, hence undecided and ambiguous, then we are immediately prevented from making constructive assertions that would invalidate non-constructive interpretations of PA. We could, then, develop non-constructive interpretations that are not necessarily claimed as valid or meaningful, but, by default, simply as valid, and possibly meaningful, till shown otherwise.

However, if we tolerate such ambiguities, then we are in danger of shortchanging scientific disciplines for whom mathematics is, essentially, a language of reliable, and verifiable, external expression and communication. Such a language should, clearly, be based on notions of formal truth that offer a maximum of precision in, and verifiability of, its assertions, with a minimum of ambiguity.

For most scientific disciplines, the authority of the standard interpretations of classical mathematics is seen, and accepted - perhaps with some element of reluctance, since such acceptance occasionally flies against the grain of observation and experience - not only as absolute, but also as implicitly promising sufficiency, when needed, to help bridge the seemingly unbridgeable chasm between a Platonic world of abstract objects, and the real world of sensory perceptions, that sometimes confronts such disciplines!

## **4. The Halting probability**

### **4.1 CCM's Thesis and the Halting problem**

The significance of a constructive interpretation of classical concepts, as outlined above, for the arguments offered by ([Ca01], v2), emerges if we note that the:

*CCM Thesis:* For arithmetical functions and relations, Turing-computability is equivalent to PA-provability.

is implicit in the following remarks ([Ca01], v2):

Classically, there are two equivalent ways to look at the mathematical notion of proof: logical, as a finite sequence of sentences strictly obeying some axioms and inference rules, and computational, as a specific type of computation. Indeed, from a proof given as a sequence of sentences one can easily construct a Turing machine producing that sequence as the result of some finite computation and, conversely,

given a machine computing a proof we can just print all sentences produced during the computation and arrange them into a sequence.

Now, *prima facie*, the *CCM Thesis* seems equivalent to the Algorithmic Church Thesis, since it can be interpreted as implying that if an arithmetic function  $f(x)$  is algorithmically (Turing) computable, then, first,  $f$  is a mathematical object in PA and, second, the formula  $[(\exists!y)f(x) = y]$  is PA-provable.

It would then follow, by formal arguments that lie outside the immediate scope of this essay, that the Halting problem is effectively solvable; for, what we essentially argue there is that, given any Turing machine T, if there is an effective method to recognise whether or not a given arithmetical string P is a valid input of T, then a *CCM Thesis* should imply that there is an instantiational, non-algorithmic, effective method that will determine whether or not T will halt on input P.

#### 4.2 An effective solution of the Halting problem

We reproduce this argument below.

**Theorem 1:** The *CCM Thesis* implies that the Halting problem is effectively solvable.

**Proof:** Given a Turing machine that computes a number-theoretic function  $F(x)$ , we note that, by ([Me64], p233, Corollary 5.13),  $F(x)$  is partial recursive. We may thus assume that such an  $F$  is obtained from a recursive function  $G$  by means of the unrestricted  $\mu$ -operator; in other words, that (cf. [Me64], p214):

$$F(x) = \mu y(G(x, y) = 0).$$

If  $[H(x, y)]$  expresses  $\sim(G(x, y) = 0)$  in PA, we consider the PA-provability, and Tarskian-truth (cf. [Me64], p49), in the standard interpretation M of PA, of the formula  $[H(a, y)]$  for a given numeral  $[a]$  of PA.

Thus:

(a) Let  $Q_1$  be the meta-assertion that  $[H(a, y)]$  is not classically true in  $M$ . Since  $G(a, y)$  is recursive, it follows that there is some finite  $k$  such that any Turing machine  $T_1(y)$  that computes  $G(a, y)$  will halt and return the value 0 for  $y = k$ .

(b) Next, let  $Q_2$  be the meta-assertion that  $[H(a, y)]$  is classically true in  $M$ , but that there is no Turing machine  $T$  such that, for any given  $y$  in  $M$ ,  $T$  will halt if, and only if,  $y$  satisfies  $[H(a, y)]$  classically.

Since  $G(a, y)$  is recursive, it follows that there is some finite  $k$  such that any Turing machine  $T_2(y)$  that computes the arithmetical function  $H(a, y)$  will halt, and return the symbol for self-termination (looping), for  $y = k$ .

(c) Finally, let  $Q_3$  be the meta-assertion that  $[H(a, y)]$  is classically true in  $M$ , and that there is some Turing machine  $T$  such that, for any given  $y$  in  $M$ ,  $T$  will halt if, and only if,  $y$  satisfies  $[H(a, y)]$  classically.

Now, if we assume the *CCM Thesis*, then it follows that  $[H(a, y)]$  is PA-provable. Let  $h$  be the Gödel-number of  $[H(a, y)]$ . We consider, then, Gödel's primitive recursive number-theoretic relation  $xBy$  ([Go31a], p22, def. 45), which holds in  $M$  if, and only if,  $x$  is the Gödel-number of a proof sequence in PA for the PA-formula whose Gödel-number is  $y$ . It follows that there is some finite  $k$  such that any Turing machine  $T_3(y)$ , which computes the characteristic function<sup>16</sup> of  $xBh$ , will halt and return the value 0 for  $x = k$ <sup>17</sup>.

---

<sup>16</sup> "If  $R(x_1, \dots, x_n)$  is a relation, then its characteristic function,  $C_n(x_1, \dots, x_n)$ , is defined as follows:

$C_n(x_1, \dots, x_n) = 0$  if  $R(x_1, \dots, x_n)$  is true, and  $C_n(x_1, \dots, x_n) = 1$  if  $R(x_1, \dots, x_n)$  is false." ([Me64], p119).

<sup>17</sup> We assume that such a machine can be effectively meta-programmed to proceed to the next instantaneous tape description whenever it encounters a loop.

Since  $Q_1$ ,  $Q_2$ , and  $Q_3$  are mutually exclusive and exhaustive<sup>18</sup>, it follows that, when run simultaneously<sup>19</sup> over the sequence 1, 2, 3, ... of values for  $y$ , one of the parallel trio  $\{T_1(y) // T_2(y) // T_3(y)\}$  of Turing machines will always halt for some finite value of  $y$  for any given  $a$ .

Thus, the Halting problem is effectively solvable if we assume a *CCM Thesis*.

### 4.3 Is the Halting probability a Dedekind real number?

Now, there are  $2^k - 1$  possible digital strings of length  $k$ , which start with 1. If  $f(k)$  of these, when input to a given Universal Turing machine  $U$ , yield prefix-free Halting programs<sup>20</sup>, the classical probability that any randomly generated string of length  $k$  is prefix-free, and that it will halt, is  $HP_{U, k} = f(k)/(2^k - 1)$ .

---

<sup>18</sup> They correspond to the instances where a classical Turing machine that computes the recursive function  $G(a, y)$  will halt for some  $y$ , loop for some  $y$ , or not halt for any  $y$ , respectively.

<sup>19</sup> This concept is essentially that of parallel computing, where the action of one machine can influence the action of another unpredictably, without human intervention. Since classical Turing machines are necessarily sequential, such a procedure cannot be defined as a classical Turing machine. Hodges remarks [HA00] that the possibility of parallel machines being essentially different from his Logical Computing Machines does not (arguably) appear to have been considered by Turing:

“... Another source may lie in Turing's definition of an ‘oracle-machine’ which is a Turing machine allowed at certain points to ‘consult the oracle’. Such a machine is not purely mechanical: it is like the ‘choice-machine’ defined in (Turing 1936-7) which at certain points allows for human choices to be made. Turing used the word ‘machine’ for entities which are only partially mechanical in operation, reserving the term ‘automatic machine’ for those which are purely mechanical. Copeland appears to imagine that when Turing describes the oracle-machine definition as giving a ‘new type of machine’, he is defining a new type of automatic machine. On the contrary, Turing is defining something only partially mechanical.

To take this point further, it is worth noting that the expression ‘purely mechanical process’ enters into Turing's definitive statement of the Church-Turing thesis, which comes as an opening section to (Turing 1939), and that Turing goes on: ‘understanding by a purely mechanical process one which could be carried out by a machine’. In the subsequent discussion the word ‘machine’ is used to mean ‘Turing machine’. There is no evidence that Turing had any concept of a purely mechanical ‘machine’ of any kind other than encapsulated by the Turing machine definition.”

<sup>20</sup> We define a prefix-free Halting program as a digital string that does not start with a string, smaller than itself, that is, itself, a Halting program.

The classical probability that any randomly generated string of length less than, or equal to,  $k$  is prefix-free, and that it will halt, is, thus  $HP_{U, i=<k} = (\text{Sum}_{i=<k} f(i)) / ((\text{Sum}_{i=<k} 2^i) - k)$ .

Since, by Theorem 1 above, we can determine  $f(i)$  for any given  $i$ , we can also determine  $HP_{U, i=<k}$  as a rational number for any given  $k$ .

The question arises: Does the non-terminating sequence  $HP_U$ , expressed by  $\langle HP_{U, i=<1}, HP_{U, i=<2}, HP_{U, i=<3}, \dots \rangle$  define a Cauchy sequence, and, hence, a Dedekind real number?

Now, since  $f(k)$  is determined by a parallel duo of Turing machines that is not, itself, a Turing machine, but which can be viewed as a deterministic Turing oracle, it follows that there is no algorithmic method of determining  $f(k)$ ; in other words, although the number-theoretic function  $f(x)$  is a well-defined mathematical concept,  $f$  is not a mathematical object. Hence  $f(x)$  may be considered as determinate, but uncomputable; its values are essentially unpredictable, and so, by definition, truly random.

It follows that the rational function  $HP_{U, i=<k}$  (viewed as a number-theoretic function over ordered integer pairs), too, is, essentially, (Turing) uncomputable. Hence  $HP_U$  is also not a mathematical object, and so its range does not define a set that can be assumed to determine a set-theoretically defined Dedekind real number.

However, since  $HP_{U, i=<k}$  is, indeed, effectively computable individually for any given natural number  $k$ , we may look upon  $HP_U$  as a random mathematical function that, nevertheless, contains an infinitude of non-algorithmically computable information.

#### 4.4 Standard interpretations of the significance of the Halting probability

We contrast the above with the standard interpretation of the relation of classical theory to the Halting problem, and of the significance of the Halting probability, as expressed by the authors in ([Ca01], v2):

In modern times a penetrating insight into the incompleteness<sup>21</sup> phenomenon has been obtained by Chaitin’s information–theoretic analysis ... The simplest way to state one of Chaitin’s main results is the following ... : “If you have  $N$  bits of axioms, then you can never prove that a program is the smallest possible if it is more than  $N$  bits long.” The most striking results have been obtained by studying the Chaitin’s Omega Number, *Omega*, the halting probability of a self-delimiting universal Turing machine<sup>22</sup>. This number is not only uncomputable, but also (algorithmically) random. Chaitin has proven the following important theorem: If ZFC is arithmetically sound<sup>23</sup>, then, ZFC can determine the value of only finitely many bits of *Omega*, and one can give a bound on the number of bits of *Omega* which ZFC can determine.<sup>24</sup> Robert Solovay ... has constructed a self-delimiting universal Turing machine such that ZFC,

---

<sup>21</sup> The significance of this may need to be viewed, however, in the light of earlier remarks in footnote 1.

<sup>22</sup> Chaitin defines *Omega*, for a given Universal Turing machine  $U$ , as:

$Omega = \sum 2^{-|p|}$  over all prefix-free, binary, strings  $p$  on which  $U$  halts, where  $|p|$  is the length of the string  $p$ .

However, when compared to the classical definition of probability considered earlier, and the value of *Omega* if we eliminate the prefix-free stipulation, it is not clear in which sense this sum can be termed as a probability that  $U$  will halt when its binary, prefix-free input is chosen randomly, e.g., by flipping a coin.

<sup>23</sup> This theorem would, then, hold vacuously; it would, then, follow that, constructively, ZFC may not be arithmetically sound, in the sense that it may not be a model for standard PA.

<sup>24</sup> This result may also need to be viewed against the arguments of the previous section; if the Halting probability is not a set-theoretically defined Dedekind real number, then it is not clear what arithmetical interpretation, or significance, is to be given to a routine that calculates “finitely many bits of *Omega*” or one that “can give a bound on the number of bits of *Omega* which ZFC can determine”.

if arithmetically sound, cannot determine any single bit of its halting probability ...  
 Rephrased, the most powerful formal axiomatic system is powerless when dealing with the questions of the form “is the  $m$ 'th bit of *Omega* 0?” or “is the  $m$ 'th bit of *Omega* 1?”.

Now, a point of some significance is that, since standard interpretations of classical theory do not explicitly distinguish between well-defined mathematical concepts and well-defined mathematical objects, the distinction between a non-constructively defined Chaitin real number (*Omega*), and a Dedekind real number (which provides the foundation for classical mathematical theory) remains implicit; it is thus not clear whether the significance ascribed to mathematical assertions about the former can be meaningfully, and significantly, extended to the latter<sup>25</sup>.

## **5. Are mechanistic proofs of mathematical problems logically sound?**

In conclusion, we note that the pedantic point made by Robertson, Sanders, Seymour and Thomas, as quoted in ([Ca01], v2), about the reliability of their 1996 computer-generated proof of 4CT, can be seen as besieged sophistry:

However, an argument can be made that our “proof” is not a proof in the traditional sense, because it contains steps that can never be verified by humans. In particular, we have not proved the correctness of the compiler we compiled our programs on, nor have we proved the infallibility of the hardware we ran our programs on. These have to be taken on faith, and are conceivably a source of error.

---

<sup>25</sup> Standard interpretations of classical mathematical theory ignore the possibility of such a distinction between Cantorian real numbers and Dedekind real numbers.

Although one may doubt whether their program is logically sound, there is no essential reason why such soundness cannot be established theoretically<sup>26</sup>; thus, there is no sensible reason to doubt the output, even though the relation between a logically effective method and a mechanistic computation is, indeed, one of faith.

It is conceivable that an appropriately designed, and maintained, machine continuously calculating the digits of *Pi* may start outputting an unending series of zero's, perhaps for as long as a zillion years. Our belief that it will eventually output a digit other than zero comes about because of our faith that the machine is faithfully translating our theoretical calculations concerning the digits of *Pi* into a physical language of mechanical I/O devices; hence we believe, as an article of faith, and despite the physical evidence, that the series of zeros will not be unending.

Similarly, the effective solution of the Halting problem consists of a duo of parallel Turing machines, where an assumption of a Uniform Church (or Turing / CCM) Thesis implies that one of the two will halt either because its program is a halting program, or because it recognises and halts due to an impending looping situation. Again, physically, this could take a zillion years in some particular case, and may call upon all our reserves of patience and faith.

So, finally, what we should place our faith in is our ability to intuitively “see” the soundness of our axiomatic assertions, and theses, at the lowest, and not, as some believe, at a sufficiently sophisticated, level of understanding. This, of course, is the distinguishing feature of, for instance, Dedekind's formulation of the Peano axioms, or -

---

<sup>26</sup> Since every step of a formal proof sequence is either an axiom, or an immediate consequence of any two preceding elements of the sequence, each step can be effectively verified mechanistically by identifying the concerned axiom, or two preceding elements of the sequence. So long as each step is verified as logically sound, such a procedure need not be time bound, nor limited to the conceptual ability of any one individual to grasp, or even verify, the correctness of the entire proof.

as any high-school student can testify - Euclid's axioms for geometry; except the parallel postulate, these axioms are obvious to all, even if their consequences are not.

## References

[Br93] Bringsjord, S. 1993. *The Narrational Case Against Church's Thesis*. Easter APA meetings, Atlanta.

<Web page: <http://www.rpi.edu/~brings/SELPAP/CT/ct/ct.html>>

[Ca01] Calude, Cristian S., Calude, Elena, and Marcus, Solomon. 2001. *Passages of Proof*. Workshop, Annual Conference of the Australasian Association of Philosophy (New Zealand Division), Auckland.

<PDF file: <http://arXiv.org/abs/math.HO/0305213>>

[Da95] Davis, M. 1995. *Is mathematical insight algorithmic?* Behavioral and Brain Sciences, 13 (4), 659--60.

<PDF file: <http://citeseer.nj.nec.com/davis90is.html>>

[Go31a] Gödel, Kurt. 1931. *On formally undecidable propositions of Principia Mathematica and related systems I*. Translated by Elliott Mendelson. In M. Davis (ed.). 1965. *The Undecidable*. Raven Press, New York.

[Go31b] Gödel, Kurt. 1931. *On formally undecidable propositions of Principia Mathematica and related systems I*. Translated by B. Meltzer.

<Web page: <http://home.ddc.net/ygg/etext/godel/index.htm>>

[Ha47] Hardy, G.H. 1947, 9<sup>th</sup> ed. *Pure Mathematics*. Cambridge, New York.

- [Ho00] Hodges, A. 2000. *Uncomputability in the work of Alan Turing and Roger Penrose*.  
<Unpublished lecture: <http://www.turing.org.uk/philosophy/lecture1.html>>
- [Ka59] Kalmár, L. 1959. *An Argument Against the Plausibility of Church's Thesis*. In Heyting, A. (ed.) *Constructivity in Mathematics*. North-Holland, Amsterdam.
- [Kl36] Kleene, S.C. 1936. *General Recursive Functions of Natural Numbers*. *Math. Annalen* **112**.
- [La51] Landau, E.G.H. 1951. *Foundations of Analysis*. Chelsea Publishing Co., New York.
- [Me64] Mendelson, Elliott. 1964. *Introduction to Mathematical Logic*. Van Norstrand, Princeton.
- [Me90] Mendelson, E. 1990. *Second Thoughts About Church's Thesis and Mathematical Proofs*. *Journal of Philosophy* **87.5**.
- [Ru53] Rudin, Walter. 1953. *Principles of Mathematical Analysis*. McGraw Hill, New York.
- [Ti61] Titchmarsh, E. C. 1961. *The Theory of Functions*. Oxford University Press.
- [Tu36] Turing, Alan. 1936. *On computable numbers, with an application to the Entscheidungsproblem*. *Proceedings of the London Mathematical Society*, ser. 2. vol. 42 (1936-7), pp.230-265; corrections, *Ibid*, vol 43 (1937) pp. 544-546.  
<Web version: <http://www.abelard.org/turpap2/tp2-ie.asp#index>>

(Created: Friday 15<sup>th</sup> July 2005 10:14:45 AM IST. Updated: Monday 10<sup>th</sup> July 2006 9:20:44 PM IST by [re@alixcomsi.com](mailto:re@alixcomsi.com))