Does resolving PvNP require a paradigm shift?

An investigation into the philosophical and mathematical significance of Aristotle’s particularisation in the foundations of mathematics, logic and computability

Bhupinder Singh Anand

Draft of June 23, 2010.

Abstract

I shall argue that a resolution of the PvNP problem requires building an iff bridge between the domain of provability and that of computability. The former concerns how a human intelligence decides the truth of number-theoretic relations, and is formalised by the first-order Peano Arithmetic PA following Dedekind’s axiomatisation of Peano’s Postulates. The latter concerns how a human intelligence computes the values of number-theoretic functions, and is formalised by the operations of a Turing Machine following Turing’s analysis of computable functions. I shall show that such a bridge requires objective definitions of both an ‘algorithmic’ interpretation of PA, and an ‘instantiational’ interpretation of PA. I shall show that both interpretations are implicit in the definition of the subjectively defined ‘standard’ interpretation of PA. However the existence of, and distinction between, the two objectively definable interpretations—and the fact that the former is sound whilst the latter is not—is obscured by the extraneous presumption under the ‘standard’ interpretation of PA that Aristotle’s particularisation must hold over the structure \( \mathbb{N} \) of the natural numbers. I shall argue that recognising the falseness of this belief awaits a paradigm shift in our perception of the application of Tarski’s analysis (of the concept of truth in the languages of the deductive sciences) to the ‘standard’ interpretation of PA. I shall then show that an arithmetical formula \([F]\) is PA-provable if, and only if, \([F]\) interprets as true under an algorithmic interpretation of PA. I shall finally show how it then follows from Gödel’s construction of a formally ‘undecidable’ arithmetical proposition that there is a Halting-type PA formula which—by Tarski’s definitions—is algorithmically verifiable as true, but not algorithmically computable as true, under a sound interpretation of PA.

1 Introduction

I define what it means for a number-theoretic function to be:

(i) Instantiationally computable;

(ii) Algorithmically computable.
I argue that P ≠ NP if a number-theoretic function is instantiationally computable but not algorithmically computable. I then show in Lemma 10 below that if Aristotle’s particularisation is presumed valid over the structure \(\mathcal{N}\) of the natural numbers—as is the case under the standard interpretation of PA—then it follows from the instantiational nature of the constructive definition of the Gödel \(\beta\)-function \(1\) that a primitive recursive relation can be instantiationally equivalent to an arithmetical relation where the former is algorithmically computable as always true over \(\mathcal{N}\) whilst the latter is instantiationally computable, but not algorithmically computable, as always true over \(\mathcal{N}\).

I note that this argument cannot be formalised in ZF since functions are defined extensionally as mappings. Hence ZF cannot recognise that a primitive recursive relation may be instantiationally equivalent to, but computationally different from, an arithmetical relation where the former is algorithmically computable as always true over \(\mathcal{N}\) whilst the latter is instantiationally computable, but not algorithmically computable, as always true over \(\mathcal{N}\).

I therefore conclude in Theorem 1 that, if the standard interpretation of PA is presumed sound, then P ≠ NP.

I then consider an algorithmic interpretation of PA that yields P ≠ NP in Theorem 7, but which does not appeal to Aristotle’s particularisation and which is provably sound.

2 The PvNP problem

In a 2009 survey of the status of the P versus NP problem, Lance Fortnow wrote:

“... in the mid-1980’s, many believed that the quickly developing area of circuit complexity would soon settle the P versus NP problem, whether every algorithmic problem with efficiently verifiable solutions have efficiently computable solutions. But circuit complexity and other approaches to the problem have stalled and we have little reason to believe we will see a proof separating P from NP in the near future.

... As we solve larger and more complex problems with greater computational power and cleverer algorithms, the problems we cannot tackle begin to stand out. The theory of NP-completeness helps us understand these limitations and the P versus NP problems begins to loom large not just as an interesting theoretical question in computer science, but as a basic principle that permeates all the sciences.

... None of us truly understand the P versus NP problem, we have only begun to peel the layers around this increasingly complex question.”

1 Introduced by Kurt Gödel in Theorem VII ([Go31], pp.30-31) of his seminal 1931 paper on formally undecidable arithmetical propositions.
2 [Fo09].
2.1 Equivalent definitions of P, NP and P≠NP

The formal definition of the class P by Stephen Cook\(^3\) admits a number-theoretic function \(F\)—viewed set-theoretically as defining (and defined by) a unique subset \(L\) of the set \(\Sigma^*\) of finite strings over some non-empty finite alphabet set \(\Sigma\)—in P if, and only if, some deterministic Turing machine TM accepts \(L\) and runs in polynomial time.

In this investigation I interpret number-theoretic functions and relations over an infinite domain \(D\) as pre-Cantorian computational instructions (which may, or may not, be uniform) that, for any given sequence of allowable values to the variables in the function/relation, determine how the function/relation is to be evaluated—and whether, or not, the result of such evaluation yields a value (or values)—in the domain \(D\). I do not assume—as in Cantorian set theories—that the evaluations always determine a completed infinity (set) that can be referred to as a unique mathematical constant that identifies the function/relation in a mathematical language (or its interpretation) outside of the set theory in which the function/relation is defined.

Fortnow describes the P≠NP problem informally as follows:

“In 1965, Jack Edmonds . . . suggested a formal definition of “efficient computation” (runs in time a fixed polynomial of the input size). The class of problems with efficient solutions would later become known as P for “Polynomial Time”.

. . . But many related problems do not seem to have such an efficient algorithm.

. . . The collection of problems that have efficiently verifiable solutions is known as NP (for “Nondeterministic Polynomial-Time” . . .).

So P=NP means that for every problem that has an efficiently verifiable solution, we can find that solution efficiently as well.

. . . If a formula \(\phi\) is not a tautology, we can give an easy proof of that fact by exhibiting an assignment of the variables that makes \(\phi\) false. But if . . . there are no short proofs of tautology that would imply P≠NP.”

In an earlier paper presented to ICM 2002, Ran Raz explained\(^4\):

“A Boolean formula \(f(x_1,\ldots,x_n)\) is a tautology if \(f(x_1,\ldots,x_n) = 1\) for every \(x_1,\ldots,x_n\). A Boolean formula \(f(x_1,\ldots,x_n)\) is unsatisfiable if \(f(x_1,\ldots,x_n) = 0\) for every \(x_1,\ldots,x_n\). Obviously, \(f\) is a tautology if and only if \(\neg f\) is unsatisfiable.

Given a formula \(f(x_1,\ldots,x_n)\), one can decide whether or not \(f\) is a tautology by checking all the possibilities for assignments to \(x_1,\ldots,x_n\). However, the time needed for this procedure is exponential in the number of variables, and hence may be exponential in the length of the formula \(f\).

. . . P≠NP is the central open problem in complexity theory and one of the most important open problems in mathematics today. The

\(^3\) [Cook].

\(^4\) [Ra02].
problem has thousands of equivalent formulations. One of these formulations is the following:

Is there a polynomial time algorithm \( A \) that gets as input a Boolean formula \( f \) and outputs 1 if and only if \( f \) is a tautology?

\( \text{P} \neq \text{NP} \) states that there is no such algorithm.”

Clearly, the issue of whether, or not, there is a polynomial time algorithm \( A \) that gets as input a Boolean formula \( f \) and outputs 1 if and only if \( f \) is a tautology is meaningful only if we can establish that there is an algorithm \( A \) that gets as input a Boolean formula \( f \) and outputs 1 if and only if \( f \) is a tautology.

Accordingly I show in Section 3.1 how it follows from Theorem VII of Kurt Gödel’s seminal 1931 paper—on formally undecidable arithmetical propositions—that every recursive function \( f(x_1, x_2) \) is representable in PA by a formula \([\exists x_3] F(x_1, x_2, x_3)\) such that \([\exists x_3] F(x_1, x_2, x_3)\] is “efficiently verifiable”, but not “efficiently computable”, if the standard interpretation of PA—which presumes that Aristotle’s particularisation holds over the structure \( \mathbb{N} \) of the natural numbers—is sound.

**Aristotle’s particularisation** This holds that from an assertion such as:

‘It is not the case that, for any given \( x \), \( P^*(x) \) does not hold’,

usually denoted symbolically by ‘\( \lnot (\forall x) \lnot P^*(x) \)’, we may always validly infer in the classical, Aristotlean, logic of predicates that:

‘There exists an unspecified \( x \) such that \( P^*(x) \) holds’,

usually denoted symbolically by ‘\( (\exists x) P^*(x) \)’.

However, as L. E. J. Brouwer pointed out in a seminal 1908 paper\(^{11}\), the presumption that Aristotle’s particularisation holds over \( \mathbb{N} \) lies beyond our common intuition. In the rest of the investigation I therefore consider whether the above conclusion would persist under any sound interpretation of PA.

We may express Aristotle’s particularisation in a contemporary context as:

From an assertion such as:

\[ \forall x \neg P^*(x) \]

5cf. [Go31], p.29: Every recursive relation is arithmetical.

6The symbol ‘\( [\exists x] \)’ denotes uniqueness, in the sense that the PA formula \([\exists x_3] F(x_1, x_2, x_3)\) is a short-hand notation for the PA formula \( \neg (\forall x_3) \neg F(x_1, x_2, x_3) \wedge (\forall y)(\forall z)(F(x_1, x_2, y) \wedge F(x_1, x_2, z) \rightarrow y = z)\).

7See Section 5.1. I shall follow Alfred Tarski’s terminology and definitions of the satisfaction and truth of the formulas of a formal language under an interpretation as detailed in Section 5.

8Notation: The asterisk indicates that the expression is to be interpreted semantically with respect to some well-defined interpretation. I shall aim to use this notation consistently in this investigation.

9[HA28], pp.58-59.

10See Appendix A, Section 10 for the meaning and usage of the symbol denoting the existential quantifier in an interpretation.

11[Br08].

4
It is not the case that, for any given \( x \), any witness \( W_D \) of a domain \( D \) can decide that \( P^*(x) \) does not hold in \( D \), usually denoted symbolically by ‘\( \neg(\forall x)\neg P^*(x) \)’, we may always validly infer that:

‘There exists an unspecified \( x \) such that any witness \( W_D \) of \( D \) can decide that \( P^*(x) \) holds in \( D \),’

usually denoted symbolically by ‘\( (\exists x)P^*(x) \)’.

I note that, prima facie, Brouwer’s objection seems valid since Aristotle’s particularisation does not hold if we take \( D \) as the domain of the natural numbers and the witness \( W_N \) as a Turing machine, since \( P^*(x) \) may be a Halting-type number-theoretic relation.

Thus, to ensure that the arguments of this investigation are intuitionistically unobjectionable, any assumption that the ‘standard’ interpretation \( \mathcal{I}_{PA(N, \text{Standard})} \) of \( PA \) is sound shall be explicit.

### 2.2 Defining instantiational computability and algorithmic computability

We introduce the two concepts:\(^{12}\)

**Definition 1 Instantiational computability**: A Boolean number-theoretic function \([F(x_1, \ldots, x_n)]\) is instantiational computable if, and only if, there is a Turing machine \( TM \) that, for any given sequence of numerals \([a_1, \ldots, a_n]\), will accept the natural number input \( m \) if \( m \) is a unique identification number of the formula \([F(a_1, \ldots, a_n)]\), and will always then halt with one of the following as output:

- (i) 0 if \([F(a_1, \ldots, a_n)]\) computes as 0 (or interprets as true) in \( \mathbb{N} \);
- (ii) 1 if \([F(a_1, \ldots, a_n)]\) computes as 1 (or interprets as false) in \( \mathbb{N} \).

**Definition 2 Algorithmic computability**: A Boolean number-theoretic function \([F(x_1, \ldots, x_n)]\) is algorithmically computable if, and only if, there is a Turing machine \( TM_F \) that, for any given sequence of numerals \([a_1, \ldots, a_n]\), will accept the natural number input \( m \) if, and only if, \( m \) is a unique identification number of the formula \([F(a_1, \ldots, a_n)]\), and will always then halt with one of the following as output:

- (i) 0 if \([F(a_1, \ldots, a_n)]\) computes as 0 (or interprets as true) in \( \mathbb{N} \);
- (ii) 1 if \([F(a_1, \ldots, a_n)]\) computes as 1 (or interprets as false) in \( \mathbb{N} \).

The set of identification numbers \( m \) thus corresponds to the set-theoretically defined language \( L \) accepted by \( TM_F \) in Cook’s definition of the class \( P \).

\(^{12}\)The word ‘witness’ is intended to be construed broadly in its usual dictionary sense, and not as a specifically defined technical term. However, see Section \( ^{13} \) for a more specific sense of the term ‘witness’ as used in this investigation.

\(^{13}\)My thanks to Dr. Chaitanya H. Mehta for advising that the focus of this investigation should be the distinction between these two concepts.

\(^{14}\)Strictly speaking, a formula of a formal language that interprets as a Boolean number-theoretic function under a well-defined interpretation of the language.

\(^{15}\)I shall use square brackets to differentiate a formal expression such as \([F(a_1, \ldots, a_n)]\) from its interpretation \( F^*(x_1, \ldots, x_n) \). See Appendix A, Section \( ^{10} \) for the notation and definitions of standard terms as used in this investigation.
It is reasonable to assume that the following thesis will hold when the concepts “efficiently verifiable” and “efficiently computable” are formalised in any formal system of Arithmetic:

**Thesis 1**

(a) A PA formula \([F(x_1, \ldots, x_n)]\) is efficiently verifiable if, and only if, it is instantiationally computable.

(b) If a PA formula \([F(x_1, \ldots, x_n)]\) is efficiently computable then it is algorithmically computable.

It follows that:

**Lemma 1** If Thesis 1 holds, then \(P \neq NP\) if a PA formula \([F(x_1, \ldots, x_n)]\) is instantiationally computable, but not algorithmically computable.

**Proof** By Thesis 1, if \([F(x_1, \ldots, x_n)]\) is instantiationally computable, then it is efficiently verifiable; whereas if \([F(x_1, \ldots, x_n)]\) is not algorithmically computable, then it is not efficiently computable.

Lemma 1 is intended to highlight the fact that the definition of a tautology only requires that a Boolean number-theoretic function \(f(x_1, \ldots, x_n)\) be computable instantiationally as always true; unless we presume the Church-Turing Thesis, it does not require that \(f(x_1, \ldots, x_n)\) be partial recursive, and therefore computable algorithmically as always true.

I shall argue in Section 8 that (as in the case of interpretations of PA in Section 5) it is an implicit belief in the plausibility of—or informal reliance upon—\(P \neq NP\) that obscures the distinction between ‘instantiational’ computability and ‘algorithmic’ computability.

The question thus arises: Is there a Halting-type PA formula \([F(x_1, \ldots, x_n)]\) that is computable instantiationally, but not algorithmically, as always true under a sound interpretation of PA?

### 2.3 Is there a Halting-type tautology?

To place this query in perspective I note that:

**Lemma 2** If PA has a sound interpretation \(I_{PA(N, \text{Sound})}\) over \(N\), then any PA-provable formula \([F(x_1, \ldots, x_n)]\) is instantiationally computable as always true over \(N\) under \(I_{PA(N, \text{Sound})}\).

**Proof** Gödel has shown how we can algorithmically assign a unique natural (Gödel) number to each PA formula and to each finite sequence of PA formulae. Gödel has also shown how we can construct a primitive recursive relation \(xBy\) that holds if, and only if, \(x\) is the Gödel number of a proof sequence in PA, and \(y\) is the Gödel number of the last formula of the sequence.

Now, if the PA formula \([F(x_1, \ldots, x_n)]\) is PA-provable then, for any given sequence of numerals \([(a_1, \ldots, a_n)]\), the PA formula \([F(a_1, \ldots, a_n)]\) is PA-provable.

---

\(^{16}\)See, for instance, [Rig87], p.21, “Almost all the proofs in this book will use Church’s Thesis to some extent”.

\(^{17}\) [Go31], p.13.

\(^{18}\) [Go31], p.22(45)
Hence $xB[[F(a_1, \ldots, a_n)]]$ always holds for some $x$. Since $xB_B$ is recursive, there is a Turing machine $TM_B$ that will accept $m$ if $m$ is $[[F(a_1, \ldots, a_n)]]$ and halt with output ‘provable’.

Since a PA-provable formula is true under $I_{PA(N, \text{Sound})}$, the lemma follows.

Although the following argument is informal, a formal proof follows immediately from Section 6.2 where I show in Theorem 4 that an algorithmic interpretation $I_{PA(N, \text{Algorithmic})}$ of PA—under which a PA-provable formula is algorithmically computable as always true over $N$—is sound.

**Lemma 3** If PA has a sound interpretation $I_{PA(N, \text{Sound})}$ over $N$, then any PA-provable formula $[F(x_1, \ldots, x_n)]$ is algorithmically computable as always true over $N$.

**Proof** If a PA formula $[F(x_1, \ldots, x_n)]$ is PA-provable, then there is a finite proof sequence in PA whose last member is $[F(x_1, \ldots, x_n)]$. Under any sound interpretation of PA over $N$ this sequence must interpret as an algorithm (program) of fixed size that, for any sequence $[(a_1, \ldots, a_n)]$ of PA numerals, decides $[F(a_1, \ldots, a_n)]$ as true. This algorithm defines a Turing machine $TM_F$ that, for any natural number sequence $(a_1, \ldots, a_n)$, will:

(i) accept the natural number $m$ if, and only if, $m$ is the Gödel number of $[F(a_1, \ldots, a_n)]$;

(ii) halt on any such input $m$ with output ‘true’.

The lemma follows.

**Lemma 4** If PA has a sound interpretation $I_{PA(N, \text{Sound})}$ over $N$, then we may reasonably assume that any PA-provable formula $[F(x_1, \ldots, x_n)]$ is instantiationally computable as always true over $N$ in polynomial time under $I_{PA(N, \text{Sound})}$.

**Proof** If the PA formula $[F(x_1, \ldots, x_n)]$ is PA-provable then there is a sequence $[S_1, S_2, \ldots, S_k]$ of PA formulas each of which is either a PA axiom, or a consequence of the PA axioms and the preceding formulas in the sequence by application of some Rule of Deduction of PA, and where $S_k$ is $[F(x_1, \ldots, x_n)]$.

Further, for any given sequence of numerals $(a_1, \ldots, a_n)$, the PA formula $[F(x_1, \ldots, x_n) \rightarrow F(a_1, \ldots, a_n)]$ is PA-provable. Hence there is a proof sequence $[S_1, S_2, \ldots, S_k, S_{k+1}, \ldots, S_l]$ such that $S_l$ is $[F(a_1, \ldots, a_n)]$.

Now it follows in any system of Gödel numbering such as that defined by Gödel in his seminal 1931 paper on formally undecidable arithmetical propositions$^{21}$ that $[[S_1, S_2, \ldots, S_k, S_{k+1}, \ldots, S_l]] = [\lambda S. \, c(a_1, \ldots, a_n)]$, where $c(a_1, \ldots, a_n)$ is a constant whose value is determined by the sequence $(a_1, \ldots, a_n)$.

Now $xB[[F(x_1, \ldots, x_n)]]$ holds for $x = [S_1, S_2, \ldots, S_k]$, whilst $xB[[F(a_1, \ldots, a_n)]]$ holds for $x = [S_1, S_2, \ldots, S_k, S_{k+1}, \ldots, S_l]$. Hence if the Turing machine $TM_B$ ‘computes’ $[F(x_1, \ldots, x_n)]$ as ‘true’ in time $t$, then it will ‘compute’ $[F(a_1, \ldots, a_n)]$ as ‘true in time $t \ast c(a_1, \ldots, a_n)$ where $c(a_1, \ldots, a_n)$ is a constant whose value is determined by the sequence $(a_1, \ldots, a_n)$.

It is reasonable to presume that $t$ can be treated as a measure that is representative of the length of the program of the Turing machine $TM_F$ defined in Lemma 3. The lemma follows.

---

19 $[[F(a_1, \ldots, a_n)]]$ denotes the Gödel number of $[F(a_1, \ldots, a_n)]$.

20 For a proof of the necessity see Section 7 Theorem 6.

21 Gödel, p.13.
I further note that:

**Lemma 5** If PA has a sound interpretation \( \mathcal{I}_{PA(N, \text{Sound})} \) over \( N \), then there is a PA formula \([F]\) which is instantiationally computable as always true over \( N \) under \( \mathcal{I}_{PA(N, \text{Sound})} \) even though \([F]\) is not PA-provable.

**Proof** Gödel has shown how to construct an arithmetical formula with a single variable—say \([R(x)]\)—such that \([R(x)]\) is not PA-provable, but \([R(n)]\) is instantiationally PA-provable for any given PA numeral \([n]\). Hence, for any given numeral \([n]\), the PA formula \(xB\)[\(R(n)\)] must hold for some \( x \). The lemma follows.

The question arises: Is there a Turing machine \( TM_{R} \) that, for any given numeral \([n]\), accepts the natural number input \( m \) if, and only if, \( m \) is the Gödel number of \([R(n)]\), and halts with output 0 if \([R(n)]\) is true under \( \mathcal{I}_{PA(N, \text{Sound})} \), and with output 1 if \([R(n)]\) is false under \( \mathcal{I}_{PA(N, \text{Sound})} \)?

Obviously there can be no such algorithm if \([R(x)]\) is a Halting-type formula such that there would be some putative Gödel number \([\llbracket R(n) \rrbracket]\) on which any putative Turing machine \( TM_{R} \) defined as above cannot output either 0 or 1.

This could be the case if the definition of the formula in question references—either directly or indirectly—algorithmic computations of some number-theoretic functions over \( N \). Such reference occurs in Gödel’s definition of \([R(x)]\), which involves an explicit—and deliberate—self-reference. However it also occurs—albeit implicitly—in Gödel’s proof that any recursive Boolean function such as \( x_0 = f(x_1, x_2) \) is representable by a PA formula \([F(x_0, x_1, x_2)]\). The proof involves defining \([F(x_0, x_1, x_2)]\) only by its instantiations. Moreover, for any given numerals \([k, m] \), the instantiation \([F(k, m, i)]\) is defined in terms of Gödel’s \( \beta \)-function (see Section 3.1.1)—which is such that \( \beta(u_{(k,m)}, v_{(k,m)}, i) \) represents the first \( m \) terms, i.e. \( f(k,0), f(k,1), \ldots, f(k,m) \) of \( f(k, x_2) \). Thus \([F(x_0, x_1, x_2)]\) implicitly references the values of a putative \( \beta(u_{(x_1,x_2)}, v_{(x_1,x_2)}, i) \) which would represent the sequence \( f(x_1,0), f(x_1,1), \ldots, f(x_1, x_2) \) for unspecified \( x_1, x_2 \) over \( N \).

The thesis that I shall seek to address formally in this investigation is thus:

**Thesis 2** Under any sound interpretation of PA, Gödel’s \([R(x)]\) is instantiationally computable, but not algorithmically computable, as always true in \( N \).

Moreover, I shall seek to show why—as Fortnow appears to suggest—resolving the PvNP problem may not be the major issue; the harder part may be altering our attitudes and beliefs so that we can see what is obstructing such a resolution.

### 3 Bridging provability and computability

In a 1956 letter to John von Neumann, Gödel raised an issue of computational complexity that is commonly accepted as a precursor of the PvNP problem:

---

22 Gödel refers to this formula only by its Gödel number \( r \) (Go31, p.25(12)).
23 Gödel’s aim in [Go31] was to show that \([\forall x]R(x)\) is not \( P \)-provable; by Generalisation it follows, however, that \([R(x)\]) is also not \( P \)-provable.
24 [Go31], p.29, Theorem VII.
25 See Section 4, Corollary 8.
26 See [Go56] for a translation as provided by Juris Hartmanis.

---

8
One can obviously easily construct a Turing machine, which for every formula $F$ in first order predicate logic and every natural number $n$, allows one to decide if there is a proof of $F$ of length $n$ (length = number of symbols). Let $\Psi(F, n)$ be the number of steps the machine requires for this and let $\phi(n) = \max_F \Psi(F, n)$. The question is how fast $\phi(n)$ grows for an optimal machine. One can show that $\phi(n) \geq K.n$. If there really were a machine with $f(n) \approx K.n$ (or even $\approx K.n^2$), this would have consequences of the greatest importance. Namely, it would obviously mean that in spite of the undecidability of the Entscheidungsproblem, the mental work of a mathematician concerning Yes-or-No questions could be completely replaced by a machine. After all, one would simply have to choose the natural number $n$ so large that when the machine does not deliver a result, it makes no sense to think more about the problem. Now it seems to me, however, to be completely within the realm of possibility that $\phi(n)$ grows that slowly. Since it seems that $\phi(n) = K.n$ is the only estimation which one can obtain by a generalization of the proof of the undecidability of the Entscheidungsproblem and after all $\phi(n) \approx K.n$ (or $\approx K.n^2$) only means that the number of steps as opposed to trial and error can be reduced from $N$ to $\log N$ (or $(\log N)^2$). However, such strong reductions appear in other finite problems, for example in the computation of the quadratic residue symbol using repeated application of the law of reciprocity. It would be interesting to know, for instance, the situation concerning the determination of primality of a number and how strongly in general the number of steps in finite combinatorial problems can be reduced with respect to simple exhaustive search.

Clearly issues of computational complexity—such as those raised by Gödel above—are finitary concerns involving number-theoretic functions and relations containing quantification over $\mathbb{N}$ that lie naturally within the domains of:

(a) First-order Peano Arithmetic PA, which attempts to capture in a formal language the objective essence of how a human intelligence intuitively reasons about number-theoretic predicates, and;

(b) Computability Theory, which attempts to capture in a formal language the objective essence of how a human intelligence intuitively computes number-theoretic functions.

Moreover, since Gödel had already shown in 1931 that every recursive relation can be expressed arithmetically\textsuperscript{27} his formulation of the computational complexity of a number-theoretic problem in terms of formal arithmetical provability suggests that we ought to persist in seeking, conversely, an algorithmic interpretation of first-order PA\textsuperscript{28} in Computability Theory, so that any number-theoretic problem can be expressed—and addressed—formally in PA, and its solution, if any, interpreted algorithmically in Computability Theory. I investigate this in detail in Section\textsuperscript{5}.

3.1 Gödel’s Theorem V and formally unprovable but interpretively true propositions

Now, by Gödel’s Theorem V\textsuperscript{29} every recursive relation $f(x_1, \ldots, x_n)$ can be expressed in PA by a formula $[F(x_1, \ldots, x_n)]$ such that, for any given $n$-tuple of natural numbers $a_1, \ldots, a_n$:

\textsuperscript{27}[Go31], Theorem VII, p.31.

\textsuperscript{28}Part of the finitary consistency proof for PA sought by Hilbert in his ‘program’ ([Hi30], pp.485-494).

\textsuperscript{29}[Go31], p.22.
If \( f(a_1, \ldots, a_n) \) is true, then PA proves \([F(a_1, \ldots, a_n)]\)

If \( \neg f(a_1, \ldots, a_n) \) is true, then PA proves \([\neg F(a_1, \ldots, a_n)]\)

Gödel relies only on the above to conclude—in his Theorem VI—
the existence of an arithmetical proposition that is formally unprovable in a Peano Arithmetic, but true under a sound interpretation of the Arithmetic.

However, I now show that it is Gödel’s Theorem VII—which—for recursive relations of the form \( x_0 = \phi(x_1, \ldots, x_n) \) defined by the Recursion Rule—provides an actual blueprint for the construction of PA formulas that are PA-unprovable, but true under the standard interpretation of PA.

Moreover, I shall show that this Gödelian characteristic is merely a reflection of the fact that, by the instantiational nature of their constructive definition in terms of Gödel’s \( \beta \)-function, such formulas are designed to be instantiationally computable, but not algorithmically computable, under the standard interpretation of PA.

### 3.1.1 Every recursive function is representable in PA

I note some standard definitions and results (which implicitly presume that the standard interpretation of PA is sound, hence quantifiers are interpreted under the assumption that Aristotle’s particularisation is valid over \( \mathbb{N} \)).

Gödel has defined a primitive recursive function—Gödel’s \( \beta \)-function—as:

\[
\beta(x_1, x_2, x_3) = rm(1 + (x_3 + 1) \times x_2, x_1)
\]

where \( rm(x_1, x_2) \) denotes the remainder obtained on dividing \( x_2 \) by \( x_1 \).

Gödel showed that:

**Lemma 6** For any non-terminating sequence of values \( f(x_1, 0), f(x_1, 1), \ldots \), we can construct natural numbers \( b, c \) such that:

(i) \( j = \max(n, f(x_1, 0), f(x_1, 1), \ldots, f(x_1, n)) \);

(ii) \( c = j! \);

(iii) \( \beta(b, c, i) = f(x_1, i) \) for \( 0 \leq i \leq n \).

**Proof** This is a standard result. We reproduce Gödel’s original argument of this critical lemma in an Appendix B, Section 2.

Now we have the standard definition:

**Definition 3** A number-theoretic function \( f(x_1, \ldots, x_n) \) is said to be representable in PA if, and only if, there is a PA formula \([F(x_1, \ldots, x_{n+1})]\) with the free variables \( [x_1, \ldots, x_{n+1}] \), such that, for any given natural numbers \( k_1, \ldots, k_{n+1} \):

---

30 Göd31, p.24.
31 Göd31, p.29.
32 Me64, p.120 & p.132.
33 Such an implicit presumption is seen in Gödel’s reference in the statement of his Theorem IX to the negation of a universally quantified formula of the restricted functional calculus as indicative of “the existence of a counter-example” (Göd31, p.32).
34 cf. Göd31, p.31, Lemma 1; Me64, p.131, Proposition 3.21.
35 cf. Göd31, p.31, Lemma 1; Me64, p.131, Proposition 3.22.
36 Me64, p.118.
(i) if \( f(k_1, \ldots, k_n) = k_{n+1} \) then PA proves: \([F(k_1, \ldots, k_n, k_{n+1})]\);

(ii) PA proves: \([\exists x_n+1)F(k_1, \ldots, k_n, x_{n+1})]\).

The function \( f(x_1, \ldots, x_n) \) is said to be strongly representable in PA if we further have that:

(iii) PA proves: \([\exists x_n+1)F(x_1, \ldots, x_n, x_{n+1})]\)

We then have:

**Lemma 7** \( \beta(x_1, x_2, x_3) \) is strongly represented in PA by \([B(x_1, x_2, x_3, x_4)]\), which is defined as follows:

\([\exists w(x_1 = ((1 + (x_3 + 1) \star x_2) \star w + x_4) \land (x_4 < 1 + (x_3 + 1) \star x_2))].\)

**Proof** This is a standard result.\(^{37}\)

Gödel further showed that:

**Lemma 8** If \( f(x_1, x_2) \) is a recursive function defined by:

(i) \( f(x_1, 0) = g(x_1) \)

(ii) \( f(x_1, (x_2 + 1)) = h(x_1, x_2, f(x_1, x_2)) \)

where \( g(x_1) \) and \( h(x_1, x_2, x_3) \) are recursive functions of lower rank\(^{38}\) that are represented in PA by well-formed formulas \([G(x_1, x_2)]\) and \([H(x_1, x_2, x_3, x_4)]\), then \( f(x_1, x_2) \) is represented in PA by the following well-formed formula, denoted by \([F(x_1, x_2, x_3)]\):

\([\exists u)(\exists v)((\exists w)((B(u, v, 0, w) \land G(w))) \land B(u, v, x_3) \land (\forall w)(w < x_2 \rightarrow (\exists y) (\exists z)((B(u, v, w, y) \land B(u, v, (w + 1), z) \land H(x_1, w, y, z))]).\]

**Proof** This is a standard result.\(^{39}\) In view of the significance of this lemma for the resolution of the PvNP problem offered in Lemma\(^{10}\) below, we reproduce Gödel’s original argument and proof of the lemma in Appendix B, Section 11.

### 3.2 What does “\([\exists x_3)F(k, m, x_3)]\) is provable” assert under the standard interpretation of PA?

Now, if the PA formula \([F(x_1, x_2, x_3)]\) represents in PA the recursive function denoted by \( f(x_1, x_2) \) then by definition, for any given numerals \([k, m]\), the formula \([\exists x_3)F(k, m, x_3)]\) is provable in PA; and true under any sound interpretation of PA. We thus have that:

**Lemma 9** If we assume that the standard interpretation of PA is sound, then:

“\([\exists x_3)F(k, m, x_3)]\) is true under the standard interpretation of PA”

\(^{37}\)cf. [Me64], p.131, proposition 3.21.

\(^{38}\)cf. [Me64], p.132; [Go31], p.30(2).

\(^{39}\)cf. [Go31], p.31(2); [Me64], p.132.
is the assertion that:

Given any natural numbers $k, m$, we can construct natural numbers $t_{(k,m)}$, $u_{(k,m)}$, $v_{(k,m)}$—all functions of $k, m$—such that:

(a) $\beta(u_{(k,m)}, v_{(k,m)}, 0) = g(k);$  
(b) for all $i < m$, $\beta(u_{(k,m)}, v_{(k,m)}, i) = h(k, i, f(k, i));$
(c) $\beta(u_{(k,m)}, v_{(k,m)}, m) = t_{(k,m)};$

where $f(x_1, x_2)$, $g(x_1)$ and $h(x_1, x_2, x_3)$ are any recursive functions that are formally represented in PA by $F(x_1, x_2, x_3)$, $G(x_1, x_2)$ and $H(x_1, x_2, x_3, x_4)$ respectively such that:

(i) $f(k, 0) = g(k)$
(ii) $f(k, (y + 1)) = h(k, y, f(k, y))$ for all $y < m$
(iii) $g(x_1)$ and $h(x_1, x_2, x_3)$ are recursive functions that are assumed to be of lower rank than $f(x_1, x_2)$.

Proof For any given natural numbers $k$ and $m$, if $[F(x_1, x_2, x_3)]$ interprets as a well-defined arithmetical relation under the standard interpretation of PA, then a Turing-machine can construct the sequences $f(k, 0), f(k, 1), \ldots, f(k, m)$ and $\beta(u_{(k,m)}, v_{(k,m)}, 0), \beta(u_{(k,m)}, v_{(k,m)}, 1), \ldots, \beta(u_{(k,m)}, v_{(k,m)}, m)$ and verify the assertion. \hfill $\square$

3.2.1 If the standard interpretation of PA is sound, then $P \neq NP$

We now see that:

Lemma 10 If the standard interpretation of PA is sound, then $[(\exists x_3)F(x_1, x_2, x_3)]$ is computable instantiationally, but not computable algorithmically, as always true over $\mathcal{N}$.

Proof We assume that the standard interpretation of PA is sound (hence we may, for instance, conclude ‘There is some $x$ such that …’ from the assertion ‘It is not the case that for all $x$ it is not the case that …’ in the domain $\mathcal{N}$ of the interpretation.). It then follows from Lemma 9 that:

(1) $[(\exists x_3)F(k, m, x_3)]$ is PA-provable for any given numerals $[k, m]$.

Hence $[(\exists x_3)F(k, m, x_3)]$ is true under the standard interpretation of PA. It then follows from the definition of $[F(x_1, x_2, x_3)]$ in Lemma 8 that, for any given natural numbers $k, m$, we can construct some pair of natural numbers $u_{(k,m)}, v_{(k,m)}$—where $u_{(k,m)}, v_{(k,m)}$ are functions of the given natural numbers $k$ and $m$—such that:

(a) $\beta(u_{(k,m)}, v_{(k,m)}, i) = f(k, i)$ for $0 \leq i \leq m$;
(b) $F^*(k, m, f(k, m))$ holds in $\mathcal{N}$.

Since $\beta(x_1, x_2, x_3)$ is primitive recursive, $\beta(u_{(k,m)}, v_{(k,m)}, i)$ defines a constructible non-terminating sequence $f'(k, 0), f'(k, 1), \ldots$ for any given natural numbers $k$ and $m$ such that:
(c) \( f(k, i) = f'(k, i) \) for \( 0 \leq i \leq m \).

We can thus define a Turing machine \( \text{TM}_{\beta(u(k,m),v(k,m),i)} \) that will accept the natural number input \( g \) if \( g \) is the Gödel number of the PA formula \( [(\exists_1 x_3)F(k,m,x_3)] \), and then halt with output ‘true’. Hence \( [(\exists_1 x_3)F(x_1, x_2, x_3)] \) is computable instantiationally as always true over \( \mathcal{N} \) under the standard interpretation of PA.

(2) Now, the pair of natural numbers \( u(x_1,x_2), v(x_1,x_2) \) are defined such that:

(a) \( \beta(u(x_1,x_2), v(x_1,x_2), i) = f(x_1,i) \) for \( 0 \leq i \leq x_2 \);
(b) \( F^*(x_1, x_2, f(x_1, x_2)) \) holds in \( \mathcal{N} \);

where \( v(x_1,x_2) \) is defined in Lemma 8 as \( j! \) (see Lemma 3.2.1), and:

(c) \( j = \max(n, f(x_1, 0), f(x_1, 1), \ldots, f(x_1, x_2)) \);
(d) \( n \) is the ‘number’ of terms in the sequence \( f(x_1, 0), f(x_1, 1), \ldots, f(x_1, x_2) \).

Since \( j \) is not definable for a non-terminating sequence, \( \beta(u(x_1,x_2), v(x_1,x_2), i) \) we cannot construct a non-terminating sequence \( f'(x_1, 0), f'(x_1, 1), \ldots \) such that:

(e) \( f(k, i) = f'(k, i) \) for all \( i \geq 0 \).

We cannot thus define a Turing machine \( \text{TM}_{\beta(u(x_1,x_2), v(x_1,x_2), i)} \) that will accept the natural number input \( g \) if, and only if, \( g \) is the Gödel number of the PA formula \( [(\exists_1 x_3)F(k,m,x_3)] \), and then halt with output ‘true’. Hence \( [(\exists_1 x_3)F(x_1, x_2, x_3)] \) is not computable algorithmically as always true over \( \mathcal{N} \) under the standard interpretation of PA.

The lemma follows. \( \square \)

It follows that:

**Theorem 1** If Thesis \footnote{1} holds, and the standard interpretation of PA is sound, then \( P \neq NP \).

**Proof** By Lemma \footnote{10} \( [(\exists_1 x_3)F(x_1, x_2, x_3)] \) is computable instantiationally, but not computable algorithmically, as always true over \( \mathcal{N} \). By Lemma \footnote{1} \( P \neq NP \). \( \square \)

A critical issue that I do not address in this investigation is whether the PA formula \( \Phi(x_1, x_2, x_3) \) can be considered to interpret under a sound interpretation of PA as a well-defined predicate since the denumerable sequences \( f'(k, 0), f'(k, 1), \ldots, f'(k, m), m_p \) where \( p > 0 \), and \( m_p \) is not equal to \( m_q \) if \( p \) is not equal to \( q \) are represented by denumerable, distinctly different, functions \( \beta(x_{p1}, x_{p2}, i) \) respectively. There are thus denumerable pairs \( (x_{p1}, x_{p2}) \) for which \( \beta(x_{p1}, x_{p2}, i) \) yields any given sequence \( f'(k, 0), f'(k, 1), \ldots, f'(k, m) \).
4 Is Gödel’s undecidable arithmetical proposition a one-off anomaly?

It also follows from the preceding section that:

**Corollary 1** If the standard interpretation of PA is sound, then \((\exists_1 x_3)F(x_1, x_2, x_3)\) is not PA-provable.

**Proof** If \((\exists_1 x_3)F(x_1, x_2, x_3)\) were PA-provable then, by Lemma 3, there would be an algorithm that decides \((\exists_1 x_3)F(x_1, x_2, x_3)\) as always true under the standard interpretation of PA. By Lemma 10, this is not the case. The corollary follows.

Now, the counter-intuitive element in Gödel’s conclusions in his 1931 paper has occasionally given rise to the perception that ‘undecidable’ Gödelian propositions are artificially constructed anomalies which are not likely to be encountered in, or have any appreciable significance for, mainstream mathematics. However, this may not be a realistic perception since the first part of Gödel’s Theorem VI is merely a special case of the following theorem:

**Theorem 2** If PA is consistent and the PA formula \([F(x_1, x_2, x_3)]\) represents the recursive function \(f(x_1, x_2)\), then:

(a) \((\exists_1 x_3)F(k, m, x_3)\) is PA-provable for any given numerals \([k], [m]\);

(b) \((\exists_1 x_3)F(x_1, x_2, x_3)\) is not PA-provable.

**Proof** (a) By definition \((\exists_1 x_3)F(k, m, x_3)\) is PA-provable for any given numerals \([k], [m]\) if the PA formula \([F(x_1, x_2, x_3)]\) represents the recursive function \(f(x_1, x_2)\).

(b) If \((\exists_1 x_3)F(x_1, x_2, x_3)\) were PA-provable, then it would be algorithmically computable as true under the standard interpretation of PA. By Lemma 10, this is not the case. The theorem follows.

If, further, we assume that the standard interpretation of PA is sound, then it follows from (a) that the second part of Gödel’s Theorem V \([41]\) is a special case of the following:

**Lemma 11** If the standard interpretation of PA is sound, then \([\neg(\exists_1 x_3)F(x_1, x_2, x_3)]\) is not PA-provable.

**Proof** \([\neg(\exists_1 x_3)F(x_1, x_2, x_3)]\) is an abbreviation of the PA formula:

\[\neg(F(x_1, x_2, x_3) \land (\forall y)(\forall z)(F(x_1, x_2, y) \land F(x_1, x_2, z) \rightarrow y = z))\]

Under any sound interpretation of PA over \(\mathcal{N}\), the latter formula interprets as the arithmetical relation denoted by:

\[(\forall x_3)\neg F^a(x_1, x_2, x_3) \land (\forall y)(\forall z)(F^a(x_1, x_2, y) \land F^a(x_1, x_2, z) \rightarrow y = z)\].

If the standard interpretation of PA is sound then Aristotle’s particularisation holds over \(\mathcal{N}\), and this relation can be equivalently denoted by:

\[\neg F^a(x_1, x_2, x_3) \land (\forall y)(\forall z)(F^a(x_1, x_2, y) \land F^a(x_1, x_2, z) \rightarrow y = z)\].

---

40 [Go31], p. 25(1).
41 [Go31], p. 25(2).
\[
\neg(\exists x_3)F^*(x_1, x_2, x_3) \land (\forall y)(\forall z)(F^*(x_1, x_2, y) \land F^*(x_1, x_2, z) \rightarrow y = z).
\]

It follows that if \(\neg(\exists x_3)F(x_1, x_2, x_3)\) is PA-provable, then \(\neg(\exists x_3)F^*(x_1, x_2, x_3)\) is always true over \(\mathcal{N}\).

However, this is false since \((\exists x_3)F^*(x_1, x_2, x_3)\) is always true over \(\mathcal{N}\) by Definition 3. The lemma follows.

We thus have:

**Corollary 2** If the standard interpretation of PA is sound, then \([\exists x_3]F(k, m, x_3)\] is undecidable in PA.

**Proof** The corollary follows from Theorem 2(b) and Lemma 11.

4.1 The significance of omega-consistency and Hilbert’s program

The significance of Corollary 2 is that, in order to avoid intuitionistic objections to his reasoning in [Go31], Gödel did not assume that the standard interpretation of PA is sound. Instead, Gödel introduced the syntactic property of \(\omega\)-consistency as an explicit assumption in his formal reasoning.

Gödel explained at some length that his reasons for introducing \(\omega\)-consistency as an explicit assumption in his formal reasoning was to avoid appealing to the stronger, semantic, concept of classical arithmetical truth—a concept which is implicitly based on an intuitionistically objectionable logic that assumes Aristotle’s particularisation is valid over \(\mathcal{N}\).

However, I now show that if we assume the standard interpretation of PA is sound, then PA is consistent if, and only if, it is \(\omega\)-consistent.

4.1.1 Hilbert’s \(\omega\)-Rule

Assuming that PA has a sound interpretation over \(\mathcal{N}\), is it true that:

**Algorithmic \(\omega\)-Rule:** If it is proved that the PA formula \([F(x)]\) interprets as an arithmetical relation \(F^*(x)\) that is algorithmically decidable as true for any given natural number \(n\), then the PA formula \([\forall x]F(x)\] can be admitted as an initial formula (axiom) in PA?

The significance of this query is that, as part of his program for giving mathematical reasoning a finitary foundation, Hilbert proposed an \(\omega\)-Rule as a finitary means of extending a Peano Arithmetic to a possible completion (i.e. to logically showing that, given any arithmetical proposition, either the proposition, or its negation, is formally provable from the axioms and rules of inference of the extended Arithmetic):

**Hilbert’s \(\omega\)-Rule:** If it is proved that the PA formula \([F(x)]\) interprets as an arithmetical relation \(F^*(x)\) that is true for any given natural number \(n\), then the PA formula \([\forall x]F(x)\] can be admitted as an initial formula (axiom) in PA.

---

42 [Go31], p.23 and p.28.
43 In his introduction on p.9 of [Go31].
44 cf. [Hi30], pp.485–494.
Now, in his 1931 paper—which can, not unreasonably, be seen as the outcome of a presumed attempt to validate Hilbert’s \( \omega \)-rule—Gödel introduced the concept of \( \omega \)-consistency \(^{45} \), from which it follows that:

**Lemma 12** If we meta-assume Hilbert’s \( \omega \)-rule for PA, then a consistent PA is necessarily \( \omega \)-consistent \(^{46} \).

**Proof** If the PA formula \([F(x)]\) interprets as an arithmetical relation \( F^*(x) \) that is true for any given natural number \( n \), and the PA formula \([\forall x]F(x)\) can be admitted as an initial formula (axiom) in PA, \( \neg[\forall x]F(x) \) cannot be PA-provable if PA is consistent. The lemma follows. \( \Box \)

Moreover, it follows from Gödel’s 1931 paper that one consequence of assuming Hilbert’s \( \omega \)-Rule is that there must, then, be an undecidable arithmetical proposition \(^{47} \); a further consequence of which is that PA is essentially incomplete.

However, since Gödel’s argument in this paper—from which he concludes the existence of an undecidable arithmetical proposition—is based on the weaker (i.e., weaker than assuming Hilbert’s \( \omega \)-rule) premise that a consistent PA can be \( \omega \)-consistent, the question arises whether an even weaker Algorithmic \( \omega \)-Rule (which, prima facie, does not imply that a consistent PA is necessarily \( \omega \)-consistent) can yield a finitary completion for PA as sought by Hilbert, albeit for an \( \omega \)-inconsistent PA.

### 4.1.2 Aristotle’s particularisation and \( \omega \)-consistency

I shall now argue that these issues are related, and that placing them in an appropriate perspective requires questioning not only the persisting belief that Aristotle’s 2000-year old logic of predicates—a critical component of which is Aristotle’s particularisation—remains valid even when applied over an infinite domain such as \( \mathbb{N} \), but also the basis of Brouwer’s denial of the Law of the Excluded Middle following his challenge of the belief in 1908 \(^{48} \).

Now, we have that:

**Lemma 13** If PA is consistent but not \( \omega \)-consistent, then there is some PA formula \([F(x)]\) such that, under any sound interpretation—say \( I_{PA(N, \text{Sound})} \)—of PA over \( \mathbb{N} \):

(i) for any given numeral \([n]\), the PA formula \([F(n)]\) interprets as true under \( I_{PA(N, \text{Sound})} \);

(ii) the PA formula \([\neg(\forall x)F(x)]\) interprets as true under \( I_{PA(N, \text{Sound})} \).

\(^{45}\) Gödel, p.23.

\(^{46}\) However, we cannot similarly conclude from the the Algorithmic \( \omega \)-Rule that a consistent PA is necessarily \( \omega \)-consistent.

\(^{47}\) Gödel constructed an arithmetical proposition \([R(x)]\) and showed that, if a Peano Arithmetic is \( \omega \)-consistent, then both \([\forall x]R(x)\) and \([\neg(\forall x)R(x)]\) are unprovable in the Arithmetic (Gödel, p.25(1), p.26(2)).

\(^{48}\) Brouwer.
Proof The lemma follows from the definition of $\omega$-consistency and from Tarski’s standard definitions\textsuperscript{49} of the satisfaction, and truth, of the formulas of a formal system such as PA under an interpretation as detailed in Section\textsuperscript{5}.

Further:

Lemma 14 If the PA formula $\neg(\forall x)F(x)$ interprets as true under $\mathcal{I}_{PA(N)}$, then it is not the case that, for any given PA numeral $[n]$, the PA formula $[F(n)]$ interprets as true under $\mathcal{I}_{PA(N)}$.

Proof The lemma follows from Tarski’s standard definitions of the satisfaction, and truth, of the formulas of a formal system such as PA under an interpretation.

It follows that:

Lemma 15 If the interpretation $\mathcal{I}_{PA(N)}$ admits Aristotle’s particularisation over $\mathbb{N}$, and the PA formula $\neg(\forall x)F(x)$ interprets as true under $\mathcal{I}_{PA(N)}$, then there is some unspecified PA numeral $[m]$ such that the PA formula $[F(m)]$ interprets as false under $\mathcal{I}_{PA(N)}$.

Proof The lemma follows from Aristotle’s particularisation and Tarski’s standard definitions of the satisfaction, and truth, of the formulas of a formal system such as PA under an interpretation.

Hence:

Lemma 16 If PA is consistent and Aristotle’s particularisation holds over $\mathbb{N}$, there can be no PA formula $[F(x)]$ such that, under any sound interpretation $\mathcal{I}_{PA(N, \text{Sound})}$ of PA over $\mathbb{N}$:

(i) for any given numeral $[n]$, the PA formula $[F(n)]$ interprets as true under $\mathcal{I}_{PA(N, \text{Sound})}$;

(ii) the PA formula $\neg(\forall x)F(x)$ interprets as true under $\mathcal{I}_{PA(N, \text{Sound})}$.

Proof The lemma follows from the previous two lemma.

In other words\textsuperscript{51}:

Corollary 3 If PA is consistent and Aristotle’s particularisation holds over $\mathbb{N}$, then PA is $\omega$-consistent.

It follows that:

Lemma 17 If Aristotle’s particularisation holds over $\mathbb{N}$, then PA is consistent if, and only if, it is $\omega$-consistent.

\textsuperscript{49} [Ta33]; see also [Ho01] for an explanatory exposition. However, for standardisation and convenience of expression, I follow the formal exposition of Tarski’s definitions given in [Me64], p.50.

\textsuperscript{50} As, for instance, in [Me64], pp.51-52 V(ii).

\textsuperscript{51} The above argument is made explicit in view of Martin Davis’ remark in [Da82], p.129, that such a proof of $\omega$-consistency may be “...open to the objection of circularity”.

17
Proof If PA is $\omega$-consistent then, since $[n = n]$ is PA-provable for any given PA numeral $[n]$, we cannot have that $[\neg(\forall x)(x = x)]$ is PA-provable. Since an inconsistent PA proves $[\neg(\forall x)(x = x)]$, an $\omega$-consistent PA cannot be inconsistent.

The arguments of this section and of the preceding Section 4 thus show that J. Barkley Rosser’s ‘extension’ of Gödel’s argument succeeds in avoiding an explicit assumption of $\omega$-consistency only by implicitly appealing to Aristotle’s particularisation.

4.2 Is PA $\omega$-inconsistent?

Now, it follows from the preceding section that:

**Corollary 4** If PA is consistent but not $\omega$-consistent, then Aristotle’s particularisation does not hold over $\mathcal{N}$.

As the classical, ‘standard’, interpretation of PA—say $\mathcal{I}_{PA(N, \text{Standard})}$—appeals to Aristotle’s particularisation, it follows that:

**Corollary 5** If PA is consistent but not $\omega$-consistent, then the standard interpretation $\mathcal{I}_{PA(N, \text{Standard})}$ of PA is not sound, and does not yield a model of PA.

Now, formal quantification in computational theory is currently interpreted—as in classical logic—so as to admit Aristotle’s particularisation over $\mathcal{N}$ as axiomatic.

However, if Aristotle’s particularisation does not hold over $\mathcal{N}$, it would explain to some extent why efforts to resolve the P versus NP problem by arguments that appeal to classical Aristotelian logic cannot prevail.

5 The implicit Satisfaction Condition in Tarski’s inductive assignment of truth-values under an Interpretation

I shall now show that a sound algorithmic interpretation—say $\mathcal{I}_{PA(N, \text{Algorithmic})}$—of PA under which Aristotle’s particularisation does not hold over $\mathcal{N}$ is implicit in the ‘standard’ interpretation of PA.

Moreover, the interpretation emerges naturally once we make explicit the precise role of the implicit Satisfaction Condition in Tarski’s definitive 1933 paper on the concept of truth in the languages of the deductive sciences.

Now, Tarski essentially defines:

---

52 See also [An09b].
53 See [Ro36].
54 See, for instance, [Me64], p.107 and p.52(V)(ii).
55 See [Hi25], p.382; [HA28], p.48; [Be59], pp.178 & 218.
56 In the sense of being intuitively obvious. See, for instance, [Ja62], p.xxv; [Re87], p.308 (1)-(4); [EC89], p.174 (4); [BBe03], p.102.
57 [Tr51].
58 cf. [Me64], p.51.
Definition 4 If $[A]$ is an atomic formula $[A(x_1, x_2, \ldots, x_n)]$ of a formal
language $S$, then a sequence $(a_1, a_2, \ldots, a_n)$ in the domain $D$ of an interpretation
$\mathcal{I}_{S(D)}$ of $S$ satisfies $[A]$ if, and only if:

(i) $[A(x_1, x_2, \ldots, x_n)]$ interprets under $\mathcal{I}_{S(D)}$ as a relation $A^*(x_1, x_2, \ldots, x_n)$
in $D$ for a witness $W_D$ of $D$;

(ii) $A^*(a_1, a_2, \ldots, a_n)$ holds in $D$ if, and only if, $\|\text{SATCON}(\mathcal{I}_{S(D)})\|$ holds for a witness $W_D$ of $D$;

where $\|\text{SATCON}(\mathcal{I}_{S(D)})\|$ is some Satisfaction Condition that is always decid-
able by a witness $W_D$ of $D$.

Further, Tarski’s analysis shows how we can inductively assign truth values of
’satisfaction’, ‘truth’, and ‘falsity’ as follows to the compound formulas of a first-
order theory $S$ under the interpretation $\mathcal{I}_{S(D)}$ in terms of only the satisfiability
of the atomic formulas of $S$ over $D$.

Definition 5 A sequence $s$ of $D$ satisfies $[\neg A]$ under $\mathcal{I}_{S(D)}$ if, and only if, $s$
does not satisfy $[A]$;

Definition 6 A sequence $s$ of $D$ satisfies $[A \rightarrow B]$ under $\mathcal{I}_{S(D)}$ if, and only if,
either it is not the case that $s$ satisfies $[A]$, or $s$ satisfies $[B]$;

Definition 7 A sequence $s$ of $D$ satisfies $[(\forall x_i)A]$ under $\mathcal{I}_{S(D)}$ if, and only if,
given any denumerable sequence $t$ of $D$ which differs from $s$ in at most the $i$'th component, $t$
satisfies $[A]$;

Definition 8 A well-formed formula $[A]$ of $PA$ is true under $\mathcal{I}_{S(D)}$ if, and only if,
given any denumerable sequence $t$ of $D$, $t$ satisfies $[A]$;

Definition 9 A well-formed formula $[A]$ of $PA$ is false under $\mathcal{I}_{S(D)}$ if, and only if, it is not the case that, given any denumerable sequence $t$ of $D$, $t$
satisfies $[A]$.

It then follows that

Theorem 3 (Satisfaction Theorem) If, for any interpretation $\mathcal{I}_{S(D)}$ of a first-
order theory $S$, there is a Satisfaction Condition $\|\text{SATCON}(\mathcal{I}_{S(D)})\|$ which holds
for a witness $W_D$ of $D$, then:

(i) The $\Delta_0$ formulas of $S$ are decidable as either true or false over $D$ under
$\mathcal{I}_{S(D)}$;

(ii) If the $\Delta_n$ formulas of $S$ are decidable as either true or as false over $D$
under $\mathcal{I}_{S(D)}$, then so are the $\Delta(n + 1)$ formulas of $S$.

Proof It follows from the above definitions that:

(a) If, for any given atomic formula $[A(x_1, x_2, \ldots, x_n)]$ of $S$, it is decidable by
$W_D$ whether or not a sequence $(a_1, a_2, \ldots, a_n)$ of $D$ satisfies $[A(x_1, x_2, \ldots, x_n)]$
in $D$ under $\mathcal{I}_{S(D)}$ then, for any given compound formula $[A^1(x_1, x_2, \ldots, x_n)]$ of

\text{cf.} [\text{Med.}], p.51.

\text{cf.} [\text{Med.}], pp.51-53.
S containing any one of the logical constants ¬, →, ∀, it is decidable by \( \mathcal{W}_D \) whether or not the sequence \( (a_1, a_2, \ldots, a_n) \) of \( \mathcal{D} \) satisfies \( A^1(x_1, x_2, \ldots, x_n) \) in \( \mathcal{D} \) under \( I_{\mathcal{S}(D)} \);

(b) If, for any given compound formula \([B^n(x_1, x_2, \ldots, x_n)]\) of \( S \) containing \( n \) of the logical constants ¬, →, ∀, it is decidable by \( \mathcal{W}_D \) whether or not a sequence \( (a_1, a_2, \ldots, a_n) \) of \( \mathcal{D} \) satisfies \([B^n(x_1, x_2, \ldots, x_n)]\) in \( \mathcal{D} \) under \( I_{\mathcal{S}(D)} \) then, for any given compound formula \([B^{(n+1)}(x_1, x_2, \ldots, x_n)]\) of \( S \) containing \( n+1 \) of the logical constants ¬, →, ∀, it is decidable by \( \mathcal{W}_D \) whether or not the sequence \( (a_1, a_2, \ldots, a_n) \) of \( \mathcal{D} \) satisfies \([B^{(n+1)}(x_1, x_2, \ldots, x_n)]\) in \( \mathcal{D} \) under \( I_{\mathcal{S}(D)} \);

We thus have that:

(c) The \( \Delta_0 \) formulas of \( S \) are decidable by \( \mathcal{W}_D \) as either true or false over \( \mathcal{D} \) under \( I_{\mathcal{S}(D)} \);

(d) If the \( \Delta_n \) formulas of \( S \) are decidable by \( \mathcal{W}_D \) as either true or as false over \( \mathcal{D} \) under \( I_{\mathcal{S}(D)} \), then so are the \( \Delta(n+1) \) formulas of \( S \). □

In other words, if the atomic formulas of \( S \) interpret under \( I_{\mathcal{S}(D)} \) as decidable with respect to the Satisfaction Condition \( \| \text{SATCON}(I_{\mathcal{S}(D)}) \| \) by a witness \( \mathcal{W}_D \) over some domain \( \mathcal{D} \), then the propositions of \( S \) (i.e., the \( \Pi_n \) and \( \Sigma_n \) formulas of \( S \)) also interpret as decidable with respect to \( \| \text{SATCON}(I_{\mathcal{S}(D)}) \| \) by the witness \( \mathcal{W}_D \) over \( \mathcal{D} \).

I now consider the application of Tarski’s definitions to various interpretations of first-order Peano Arithmetic \( \text{PA} \).

5.1 The standard interpretation of \( \text{PA} \)

The standard interpretation \( I_{\text{PA}(N, \text{Standard})} \) of \( \text{PA} \) is obtained if, in \( I_{\mathcal{S}(D)} \):

(a) we define \( S \) as \( \text{PA} \) with standard first-order predicate calculus as the underlying logic;

(b) we define \( \mathcal{D} \) as \( N \);

(c) for any atomic formula \([A(x_1, x_2, \ldots, x_n)]\) of \( \text{PA} \) and sequence \((a_1, a_2, \ldots, a_n)\) of \( N \), we take \( \| \text{SATCON}(I_{\text{PA}(N)}) \| \) as:

\[
\| A^*(a_1^*, a_2^*, \ldots, a_n^*) \| \text{ holds in } N \text{ and, for any given sequence } (b_1^*, b_2^*, \ldots, b_n^*) \text{ of } N, \text{ the proposition } A^*(b_1^*, b_2^*, \ldots, b_n^*) \text{ is decidable in } N;\]

(d) we define the witness \( \mathcal{W}(N, \text{Standard}) \) informally as the ‘mathematical intuition’ of a human intelligence for whom \( \| \text{SATCON}(I_{\text{PA}(N)}) \| \) is always effectively decidable in \( N \);

\textbf{Lemma 18} \( A^*(x_1, x_2, \ldots, x_n) \) is always effectively decidable in \( N \) by \( \mathcal{W}(N, \text{Standard}) \).

\textbf{Proof} If \([A(x_1, x_2, \ldots, x_n)]\) is an atomic formula of \( \text{PA} \) then, for any given sequence of numerals \([b_1, b_2, \ldots, b_n]\), the \( \text{PA} \) formula \([A(b_1, b_2, \ldots, b_n)]\) is an atomic formula of the form \([c = d]\), where \([c]\) and \([d]\) are atomic \( \text{PA} \) formulas that denote \( \text{PA} \) numerals. Since \([c]\) and \([d]\)

\textsuperscript{61}Where the string \([\exists \ldots]\) is defined as—and is to be treated as an abbreviation for—the string \([\neg(\forall \ldots)\neg]\). We do not consider the case where the underlying logic is Hilbert’s formalisation of Aristotle’s logic of predicates in terms of his e-operator \([H122]\), pp.465-466).
are recursively defined formulas in the language of PA, it follows from a standard result\textsuperscript{62} that, if PA is consistent, then \([c = d]\) is algorithmically computable as either true or false in \(N\). In other words, if PA is consistent, then \([A(x_1, x_2, \ldots, x_n)]\) is algorithmically computable (since there is a Turing machine \(T_m\) that, for any given sequence of numerals \([b_1, b_2, \ldots, b_n]\), will accept the natural number \(m\) if, and only if, \(m\) is the Gödel number of the PA formula \([A(b_1, b_2, \ldots, b_n)]\), and halt with output 0 if \([A(b_1, b_2, \ldots, b_n)]\) interprets as true in \(N\); and halt with output 1 if \([A(b_1, b_2, \ldots, b_n)]\) interprets as false in \(N\)). The lemma follows. \(\square\)

(e) we postulate that Aristotle’s particularisation holds over \(N\)\textsuperscript{63}

Clearly, (e) does not form any part of Tarski’s inductive definitions of the satisfaction, and truth, of the formulas of PA under the above interpretation. Moreover, its inclusion makes \(I_{PA(N, \text{Standard})}\) extraneously non-finitary\textsuperscript{64}.

The question arises: Can we formulate the ‘standard’ interpretation of PA without assuming (e) extraneously?
I answer this question affirmatively in Section \textsuperscript{63} where:

(1) I replace the ‘mathematical intuition’ of a human intelligence by defining an ‘objective’ witness \(W(N, \text{Instantiational})\) as the meta-theory \(M_{PA}\) of PA;

(2) I show that \(W(N, \text{Instantiational})\) can decide whether \(c^* = d^*\) is true or false by instantiationally computing the Boolean function \(A^*(x_1, x_2, \ldots, x_n)\) for any given sequence of natural numbers \((b_1^*, b_2^*, \ldots, b_n^*)\);

(3) I show that this yields an instantiational interpretation of PA over \([N]\) that is sound if, and only if, (e) holds.

(4) \(W(N, \text{Instantiational})\) is thus an instantiational formulation of the standard interpretation of PA over \(N\) (which is presumed to be sound).

I note further that if PA is \(\omega\)-inconsistent, then Aristotle’s particularisation does not hold over \(N\), and the interpretation \(I_{PA(N, \text{Standard})}\) is not sound.

### 5.2 Gödel’s non-standard interpretation of PA

A non-standard (Gödelian) interpretation \(I_{PA(N, N\text{-non-standard})}\) of a putative \(\omega\)-consistent PA is obtained if, in \(I_{S(D)}\):

(a) we define S as PA with standard first-order predicate calculus as the underlying logic;

(b) we define D as an undefined extension \(N_\omega\) of \(N\);

(c) for any atomic formula \([A(x_1, x_2, \ldots, x_n)]\) of PA and sequence \((a_1, a_2, \ldots, a_n)\) of \(N_\omega\), we take \(|\text{SATCON}(I_{PA(N, \omega)})|\) as:

\[
\|A^*(a_1^*, a_2^*, \ldots, a_n^*)\|\text{ holds in }N_\omega\text{ and, for any given sequence }\langle b_1, b_2, \ldots, b_n\rangle\text{ of }N_\omega,\text{ the proposition }A^*(b_1, b_2, \ldots, b_n)\text{ is decidable as either holding or not holding in }N_\omega;\]

\textsuperscript{62}For any natural numbers \(m, n, \text{ if } m \neq n, \text{ then PA proves }\neg(m = n)\) (\textsuperscript{Me64}, p.110, Proposition 3.6). The converse is obviously true.

\textsuperscript{63}Hence a PA formula such as \([\exists x]F(x)\) interprets under \(I_{PA(N, \text{Standard})}\) as ‘There is some natural number \(n\) such that \(F(n)\) holds in \(N\).

\textsuperscript{64}\textsuperscript{Br08}.
(d) we postulate that \( \| \text{SATCON}(I_{PA}(\mathcal{N})) \| \) is always decidable by a putative witness \( W_{\mathcal{N}_\omega} \), and that \( W_{\mathcal{N}_\omega} \) can, further, determine some numbers in \( \mathcal{N}_\omega \) which are not natural numbers;

(e) we assume that PA is \( \omega \)-consistent.

Clearly, the interpretation \( I_{PA(\mathcal{N}, \text{Non-standard})} \) of a putative \( \omega \)-consistent PA cannot claim to be finitary. Moreover, if PA is \( \omega \)-inconsistent, then the Gödelian non-standard interpretation \( I_{PA(\mathcal{N}, \text{Non-standard})} \) of PA is also not sound.

5.3 An instantiational interpretation of PA in PA

I next consider the instantiational interpretation \( I_{PA(\mathcal{N}, \text{Instantiational})} \) of PA where:

(a) we define \( S \) as PA with standard first-order predicate calculus as the underlying logic;

(b) we define \( D \) as PA;

(c) for any atomic formula \( [A(x_1, x_2, \ldots, x_n)] \) of PA and any sequence \( [(a_1, a_2, \ldots, a_n)] \) of PA numerals, we take \( \| \text{SATCON}(I_{PA}(\mathcal{N})) \| \) as:

\[
\| [A(a_1, a_2, \ldots, a_n)] \| \text{ is provable in PA and, for any given sequence of numerals } [(b_1, b_2, \ldots, b_n)] \text{ of PA, the formula } [A(b_1, b_2, \ldots, b_n)] \text{ is decidable as either provable or not provable in PA} \|
\]

(d) we define the witness \( W_{(\mathcal{N}, \text{Instantiational})} \) as the meta-theory \( M_{PA} \) of PA.

**Lemma 19** \( A(x_1, x_2, \ldots, x_n) \) is always effectively decidable in PA by \( W_{(\mathcal{N}, \text{Instantiational})} \).

**Proof** It follows from Gödel’s definition of the primitive recursive relation \( xB \)—where \( x \) is the Gödel number of a proof sequence in PA whose last term is the PA formula with Gödel-number \( y \)—that, if \( [A(x_1, x_2, \ldots, x_n)] \) is an atomic formula of PA, \( M_{PA} \) can effectively decide instantiationally for any given sequence \( [(b_1, b_2, \ldots, b_n)] \) of PA numerals which one of the PA formulas \( [A(b_1, b_2, \ldots, b_n)] \) and \( [\neg A(b_1, b_2, \ldots, b_n)] \) is necessarily PA-provable. □

Now, if PA is consistent but not \( \omega \)-consistent, then there is a Gödelian formula \( [R(x)] \) such that (see Section 5):

(i) \( [(\forall x)R(x)] \) is not PA-provable;

(ii) \( [\neg (\forall x)R(x)] \) is PA-provable;

(iii) for any given numeral \( [n] \), \( [R(n)] \) is PA-provable.

---

\(^{65}\)In which case we cannot validly conclude from Gödel’s formal reasoning in (Go31) that PA must have a non-standard model.

\(^{66}\)Go31, p. 22(45).
However, if $\mathcal{I}_{PA(N, \text{Instantiational})}$ is sound, then (ii) implies contradictorily that it is not the case that, for any given numeral $[n]$, $[R(n)]$ is PA-provable.

It follows that if $\mathcal{I}_{PA(N, \text{Instantiational})}$ is sound then PA is $\omega$-consistent and, ipso facto, Aristotle’s particularisation must hold over $\mathcal{N}$.

Moreover, if PA is consistent, then every PA-provable formula interprets as true under some sound interpretation of PA. Hence $M_{PA}$ can effectively decide whether, for any given sequence of natural numbers $(b_1^*, b_2^*, \ldots, b_n^*)$ in $\mathcal{N}$, the proposition $A^*(b_1^*, b_2^*, \ldots, b_n^*)$ holds or not in $\mathcal{N}$.

It follows that $\mathcal{I}_{PA(N, \text{Instantiational})}$ is an instantiational formulation of the ‘standard’ interpretation of PA in which we do not need to extraneously assume that Aristotle’s particularisation holds over $\mathcal{N}$.

The interpretation $\mathcal{I}_{PA(N, \text{Instantiational})}$ is of interest because, if it were a sound interpretation of PA, then PA would establish its own consistency.

### 5.4 A set-theoretic interpretation of PA

I consider next a set-theoretic interpretation $\mathcal{I}_{PA(ZF, \text{Cantor})}$ of PA over the domain of ZF sets, which is obtained if, in $\mathcal{I}_{S(D)}$:

(a) we define $S$ as PA with standard first-order predicate calculus as the underlying logic;
(b) we define $D$ as ZF;
(c) for any atomic formula $[A(x_1, x_2, \ldots, x_n)]$ and sequence $[(a_1, a_2, \ldots, a_n)]$ of PA, we take $\| \text{SATCON}(\mathcal{I}_{S(D)}) \|$ as:

$$\| [A^*(a_1^*, a_2^*, \ldots, a_n^*)] \text{ is provable in ZF and, for any given sequence } [b_1^*, b_2^*, \ldots, b_n^*] \text{ of ZF, the formula } [A^*(b_1^*, b_2^*, \ldots, b_n^*)] \text{ is decidable as either provable or not provable in ZF} |$$

(d) we define the witness $W_{ZF}$ as the meta-theory $M_{ZF}$ of ZF which can always decide effectively whether or not $\| \text{SATCON}(\mathcal{I}_{S(ZF)}) \|$ holds in ZF.

Now, if the set-theoretic interpretation $\mathcal{I}_{PA(ZF, \text{Cantor})}$ of PA is sound, then every sound interpretation of ZF would, ipso facto, be a sound interpretation of PA. In Appendix C, Section 12 I show, however, that this is not the case, and so the set-theoretic interpretation $\mathcal{I}_{PA(ZF, \text{Cantor})}$ of PA is not sound.

### 5.5 A purely algorithmic interpretation of PA

I finally consider the purely algorithmic interpretation $\mathcal{I}_{PA(N, \text{Algorithmic})}$ of PA where:

(a) we define $S$ as PA with standard first-order predicate calculus as the underlying logic;
(b) we define $D$ as $\mathcal{N}$;
(c) for any atomic formula $[A(x_1, x_2, \ldots, x_n)]$ and sequence $(a_1, a_2, \ldots, a_n)$ of $\mathcal{N}$, we take $\| \text{SATCON}(\mathcal{I}_{PA(N)}) \|$ as:

\[\text{67 cf. G"odel's Theorem XI in [Go31], p.36.}\]
\[ A^*(a_1^n, a_2^n, \ldots, a_n^n) \] holds in \( N \) and, for any given sequence \( (b_1^n, b_2^n, \ldots, b_n^n) \) of \( N \), the proposition \( A^*(b_1^n, b_2^n, \ldots, b_n^n) \) is decidable as either holding or not holding in \( N \).

(d) we define the witness \( W(N, \text{Algorithmic}) \) as a Turing machine \( TM_A \) for whom \( \|\text{SATCON}(I_{PA(N)})\| \) is always effectively decidable in \( N \):

**Lemma 20**  \( A^*(x_1, x_2, \ldots, x_n) \) is always effectively decidable in \( N \) by \( W(N, \text{Algorithmic}) \).

**Proof** If \( [A(x_1, x_2, \ldots, x_n)] \) is an atomic formula of \( PA \) then, for any given sequence of numerals \( [b_1, b_2, \ldots, b_n] \), the PA formula \( [A(b_1, b_2, \ldots, b_n)] \) is an atomic formula of the form \( [c = d] \), where \( [c] \) and \( [d] \) are atomic PA formulas that denote PA numerals. Since \( [c] \) and \( [d] \) are recursively defined formulas in the language of PA, it follows from a standard result\(^6\) that, if \( PA \) is consistent, then \( [c = d] \) is algorithmically computable as either true or false in \( N \). In other words, if \( PA \) is consistent, then \( [A(x_1, x_2, \ldots, x_n)] \) is algorithmically computable (since there is a Turing machine \( TM_A \) that, for any given sequence of numerals \( [b_1, b_2, \ldots, b_n] \), will accept the natural number \( m \) if, and only if, \( m \) is the Gödel number of the PA formula \( [A(b_1, b_2, \ldots, b_n)] \), and halt with output 0 if \( [A(b_1, b_2, \ldots, b_n)] \) interprets as true in \( N \); and halt with output 1 if \( [A(b_1, b_2, \ldots, b_n)] \) interprets as false in \( N \)). The lemma follows. \( \square \)

It follows that \( I_{PA(N, \text{Algorithmic})} \) is an algorithmic formulation of the ‘standard’ interpretation of \( PA \) in which we do not extraneously assume that Aristotle’s particularisation holds over \( N \).

I shall show that if \( I_{PA(N, \text{Algorithmic})} \) is sound, then \( PA \) is not \( \omega \)-consistent. Hence Aristotle’s particularisation does not hold over \( N \), and the interpretation is finitary and intuitionistically unobjectionable. Moreover—since the Law of the Excluded Middle is provable in an \( \omega \)-inconsistent \( PA \) (and therefore holds in \( N \)—it achieves this without the discomforting, stringent, Intuitionistic requirement that we reject the underlying logic of \( PA \)!

I now show that \( I_{PA(N, \text{Algorithmic})} \) is sound, and consider the consequences for the \( PVNP \) problem and for Church’s Thesis.

### 6 The algorithmic interpretation of \( PA \) is sound

In Section\(^6\) of this investigation I defined the two interpretations \( I_{PA(N, \text{Standard})} \) and \( I_{PA(N, \text{Algorithmic})} \) in terms of Tarski’s\(^6\) inductive definitions of the satisfaction, and truth, of the formulas of a formal system under an interpretation. It thus follows by induction on \( k \) that\(^7\).

#### 6.1 Interpreting quantification

**Lemma 21**  \( (\forall x) A(x) \) is true under \( I_{PA(N, \text{Standard})} \) if, and only if, \( PA \) proves \( \neg((m = n)) \) \( (\text{Med}3) \), p.110, Proposition 3.6). The converse is obviously true.

\(^6\)For any natural numbers \( m, n \), if \( m \neq n \), then \( PA \) proves \( \neg ((m = n)) \).\(^6\)

\(^6\)\(^6\)\(^6\)

\(^7\)cf. \(^6\)(p.51-53).

\(^7\)See Definition 8.
The lemma follows from the definition of $I_{PA(N, \text{Algorithmic})}$ by induction on $k$.

**Lemma 22** (Universal:Algorithmic) A $\Pi_k$ PA formula such as $[(\forall x)A(x)]$ interprets as true under $I_{PA(N, \text{Algorithmic})}$ if, and only if, $A^*(x)$ is algorithmically computable as always true in $N$.

**Proof** The lemma follows from the definition of $I_{PA(N, \text{Algorithmic})}$ by induction on $k$.

**Lemma 23** (Existential:Standard) A $\Sigma_k$ PA formula such as $[(\exists x)A(x)]$ interprets as true under $I_{PA(N, \text{Standard})}$ if, and only if, it is not true that, for any given natural number $n$, $A^*(n)$ is false in $N$, and we may conclude that there exists some natural number $n$ such that $A^*(n)$ holds in $N$.

**Proof** The lemma follows from the definition of $I_{PA(N, \text{Standard})}$ by induction on $k$.

**Lemma 24** (Existential:Algorithmic) A $\Sigma_k$ PA formula such as $[(\exists x)A(x)]$ interprets as true under $I_{PA(N, \text{Algorithmic})}$ if, and only if, $\neg A^*(x)$ is not algorithmically computable as always true in $N$, but we may not conclude that there exists some natural number $n$ such that $A^*(n)$ holds in $N$.

**Proof** The lemma follows from the definition of $I_{PA(N, \text{Algorithmic})}$ by induction on $k$.

6.2 Interpreting the PA axioms

We note first that:

**Lemma 25** The PA axioms $PA_1$ to $PA_8$ are algorithmically computable as always true over $N$ under the interpretation $I_{PA(N, \text{Algorithmic})}$.

**Proof** Since $[x + y]$, $[x * y]$, $[x = y]$, $[x']$ are defined recursively, the PA axioms $PA_1$ to $PA_8$ interpret as recursive relations that do not involve any quantification. The lemma follows.

Further:

**Lemma 26** For any given PA formula $[F(x)]$, the Induction axiom schema $[F(0) \rightarrow ((\forall x)(F(x) \rightarrow F(x'))) \rightarrow (\forall x)F(x)]$ interprets as true under $I_{PA(N, \text{Algorithmic})}$.

**Proof** By Tarski’s Definitions 4 to 9.

---

72 Note that $[(\exists x)A(x)]$ is merely the abbreviation for $[\neg(\forall x)\neg A(x)]$.

73 Since Aristotle’s particularisation is assumed to hold in $N$ under $I_{PA(N, \text{Standard})}$.

74 Since $A^*(x)$ may be a Halting-type of relation such that, for any given natural number $n$, it is meta-mathematically—even if not algorithmically—decidable that $A^*(n)$ is false. As I show in Section 7, Gödel’s relation $R(x)$ is precisely such a relation.

75 cf. [Go31], p.17.
(a) If \([F(0)]\) interprets as false under \(I_{PA(N, Algorithmic)}\) the lemma is proved.

(b) If \([F(0)]\) interprets as true and \([(\forall x)(F(x) \rightarrow F(x'))]\) interprets as false under \(I_{PA(N, Algorithmic)}\), the lemma is proved.

(c) If \([F(0)]\) and \([(\forall x)(F(x) \rightarrow F(x'))]\) both interpret as true under \(I_{PA(N, Algorithmic)}\), then by the Satisfaction Theorem \(3\) and the algorithmic interpretation \(I_{PA(N, Algorithmic)}\) defined in Section \(5.5\), \([F(x) \rightarrow F(x')]\) is algorithmically computable as always true over \(N\) under \(I_{PA(N, Algorithmic)}\).

There is thus a Turing machine \(T_M F\) such that, for any natural number \(n\), \(T_M F\) will accept the natural number \(m\) if, and only if, \(m\) is the Gödel number of \([F(n) \rightarrow F(n')]\) and will halt with output ‘true’.

Since \([F(0)]\) interprets as true under \(I_{PA(N, Algorithmic)}\), it follows that there is a Turing machine \(T_M F\) such that, for any natural number \(n\), \(T_M F\) will accept the natural number \(m\) if, and only if, \(m\) is the Gödel number of \([F(n)]\) and will halt with output ‘true’.

Hence \([(\forall x)F(x)]\) is algorithmically computable as always true under \(I_{PA(N, Algorithmic)}\).

Since the above cases are exhaustive, the lemma follows.  

I note that the interpretation \(I_{PA(N, Algorithmic)}\) settles the Poincaré-Hilbert debate\(^{76}\) in the latter’s favour. Poincaré believed that the Induction Axiom could not be justified finitarily, as any such argument would necessarily need to appeal to infinite induction. Hilbert believed that a finitary proof of the consistency of PA was possible.

Similarly:

**Lemma 27** Generalisation preserves truth under \(I_{PA(N, Algorithmic)}\).

**Proof** The two meta-assertions:

\[
[F(x)] \text{ interprets as true under } I_{PA(N, Algorithmic)}
\]

and

\[
[(\forall x)F(x)] \text{ interprets as true under } I_{PA(N, Algorithmic)}
\]

both mean:

\[
[F(x)] \text{ is algorithmically computable as always true under } I_{PA(N, Algorithmic)}.
\]

It is also straightforward to see that:

**Lemma 28** Modus Ponens preserves truth under \(I_{PA(N, Algorithmic)}\).  

\(^{76}\)See HILBERT, p.472; also SEP, p.59; WEIZSÄCKER, p.482; KAY, p.502-503.

\(^{77}\)See Definition \(\S\)
We thus have that:

**Lemma 29** The axioms of PA are always true under the finitary interpretation \( I_{PA(N, \text{Algorithmic})} \), and the rules of inference of PA preserve the properties of satisfaction/truth under \( I_{PA(N, \text{Algorithmic})} \).

Hence:

**Theorem 4** The interpretation \( I_{PA(N, \text{Algorithmic})} \) of PA is sound.

We thus have a finitary proof that:

**Theorem 5** PA is consistent.

### 7 A Provability Theorem for PA

I now show that PA can have no non-standard model, since it is ‘algorithmically’ complete in the sense that:

**Theorem 6** (Provability Theorem for PA) A PA formula \([F(x)]\) is PA-provable if, and only if, \([F(x)]\) is algorithmically computable in \( \mathcal{N} \).

**Proof** We have by definition that \([\forall x F(x)]\) interprets as true under the interpretation \( I_{PA(N, \text{Algorithmic})} \) if, and only if, \([F(x)]\) is algorithmically computable in \( \mathcal{N} \).

Since \( I_{PA(N, \text{Algorithmic})} \) is sound, it defines a finitary model of PA over \( \mathcal{N} \)—say \( M_{PA(\beta)} \)—such that:

- If \([\forall x F(x)]\) is PA-provable, then \([F(x)]\) is algorithmically computable in \( \mathcal{N} \);
- If \([\neg(\forall x F(x)]\) is PA-provable, then it is not the case that \([F(x)]\) is algorithmically computable in \( \mathcal{N} \).

Now, we cannot have that both \([\forall x F(x)]\) and \([\neg(\forall x F(x)]\) are PA-unprovable for some PA formula \([F(x)]\), as this would yield the contradiction:

1. There is a finitary model—say \( M_{1\beta} \)—of PA+\([\forall x F(x)]\) in which \([F(x)]\) is algorithmically computable in \( \mathcal{N} \).
2. There is a finitary model—say \( M_{2\beta} \)—of PA+\([\neg(\forall x F(x)]\) in which it is not the case that \([F(x)]\) is algorithmically computable in \( \mathcal{N} \).

The lemma follows.

**Corollary 6** PA is categorical.

By the argument in Theorem 6 it follows that:

**Corollary 7** The PA formula \([\neg(\forall x R(x)]\) defined in Lemma 5 is PA-provable.
Corollary 8 Under any sound interpretation of PA, Gédel’s $[R(x)]$ interprets as an instantiationally computable, but not algorithmically computable, tautology over $N$.

Proof Gédel has shown that $[R(x)]$ interprets as an instantiationally computable tautology. By Corollary 7 $[R(x)]$ is not algorithmically computable as always true in $N$.

Theorem 7 $P \neq NP$.

Proof By Corollary 8 $[R(x)]$ is instantiationally computable, but not algorithmically computable over $N$. The theorem follows immediately from Lemma 1.

Corollary 9 PA is not $\omega$-consistent.

Proof Gédel has shown that if PA is consistent, then $[R(n)]$ is PA-provable for any given PA numeral $[n]$. By Corollary 7 and the definition of $\omega$-consistency, if PA is consistent then it is not $\omega$-consistent.

Corollary 10 The standard interpretation $I_{PA(N, \text{Standard})}$ of PA is not sound, and does not yield a model of PA.

Proof By Corollary 1 if PA is consistent but not $\omega$-consistent, then Aristotle’s particularisation does not hold over $N$. Since the ‘standard’, interpretation of PA appeals to Aristotle’s particularisation, the lemma follows.

Since formal quantification is currently interpreted in classical logic so as to admit Aristotle’s particularisation over $N$ as axiomatic, the above suggests that we may need to review number-theoretic arguments that appeal unrestrictedly to classical Aristotelian logic.

7.1 The Provability Theorem for PA and Bounded Arithmetic

In a 1997 paper, Samuel R. Buss considered Bounded Arithmetics obtained by:

76 Gédel refers to this formula only by its Gédel number $r$; [Go31], p.25, eqn.12.
77 [Go31], p.26(2): “$(n)\sim(nB_n(17Gen r))$ holds”
78 This conclusion is contrary to accepted dogma. See, for instance, Davis’ remarks in [Da82], p.129(iii) that “...there is no equivocation. Either an adequate arithmetical logic is $\omega$-inconsistent (in which case it is possible to prove false statements within it) or it has an unsolvable decision problem and is subject to the limitations of Gédel’s incompleteness theorem”.
79 [Go31], p.26(2).
80 I note that finitists of all hues—ranging from Brouwer [Br08] to Alexander Yessenin-Volpin [He04]—have persistently questioned the soundness of the ‘standard’ interpretation $I_{PA(N, \text{Standard})}$.
81 [HI25], p.382; [HA28], p.48; [Be59], pp.178 & 218.
82 In the sense of being intuitively obvious. See, for instance, Da52, p.xxiv; [Rg87], p.308 (1)-(4); [CF99], p.174 (4); [BR03], p.102.
83 For instance—as shown in Sections 4 and 4.1—Rosser’s construction of an undecidable arithmetical proposition in PA (see [Ro36])—which does not explicitly assume that PA is $\omega$-consistent—implicitly presumes that Aristotle’s particularisation holds over $N$.
84 [Bu97].
(a) limiting the applicability of the Induction Axiom Schema in PA
only to functions with quantifiers bounded by an unspecified natural
number bound \( b \);
(b) ‘weakening’ the statement of the axiom with the aim of differ-
entiating between effective computability over the sequence of nat-
ural numbers, and feasible ‘polynomial-time’ computability over a
bounded sequence of the natural numbers.\(^8^7\)

Presumably Buss’ intent—as expressed below—is to build a bridge between
provability in a Bounded Arithmetic and Computability so that a \( \Pi_k \) formula,
say \( \forall x \, f(x) \), is provable in the Bounded Arithmetic if, and only if, there is an
algorithm that, for any given numeral \( [n] \), decides the \( \Delta_{(k/(k-1))} \) formula \( [f(n)] \)
as ‘true’:

If \( \forall x \, (\exists y \, f(x, y)) \) is provable, then there should be an algorithm
to find \( y \) as a function of \( x \).\(^8^8\)

Since we have proven such a Provability Theorem for PA in the previous
section, the first question arises:

**Does the introduction of bounded quantifiers yield any computational
advantage?**

Now, one difference\(^8^9\) between a Bounded Arithmetic and PA is that we can
presume in the Bounded Arithmetic that, from a proof of \( \exists y \, f(n, y) \), we may
always conclude that there is some numeral \( [m] \) such that \( f(n, m) \) is provable
in the arithmetic; however, this is not a sound conclusion in PA.

Reason: Since \( \exists y \, f(n, y) \) is simply a shorthand for \( \neg \forall y \, \neg f(n, y) \), such
a presumption implies that Aristotle’s particularisation holds over the natural
numbers under any sound interpretation of PA.

To see that (as Brouwer steadfastly held) this may not always be the case,
interpret \( \forall x \, f(x) \) as\(^9^0\):

There is an algorithm that decides \( [f(n)] \) as ‘true’ for any given
numeral \( [n] \).

In such case, if \( \forall x \, (\exists y \, f(x, y)) \) is provable in PA, then we can only conclude
that:

There is an algorithm that, for any given numeral \( [n] \), decides that it
is not the case that there is an algorithm that, for any given numeral
\( [m] \), decides \( \neg f(n, m) \) as ‘true’.

We cannot, however, conclude - as we can in a Bounded Arithmetic - that:

There is an algorithm that, for any given numeral \( [n] \), decides that there
is an algorithm that, for some numeral \( [m] \), decides \( f(n, m) \)
as ‘true’.

\(^8^7\) See also [Pa71].
\(^8^8\) See [Bu97].
\(^8^9\) I suspect the only one.
\(^9^0\) We have seen in the earlier sections that such an interpretation is sound.
Reason: \[\exists y f(n,y)\] may be a Halting-type formula for some numeral \([n]\).
This could be the case if \[\forall x (\exists y f(x,y))\] were PA-unprovable, but \[\exists y f(n,y)\]
PA-provable for any given numeral \([n]\).
Presumably it is the belief that any sound interpretation of PA requires
Aristotle’s particularisation to hold in \(N\), and the recognition that the latter
does not admit linking provability to computability in PA, which has led to
considering the effect of bounding quantification in PA.
However, as we have seen in the preceding sections, we are able to link
provability to computability through the Provability Theorem for PA by recogn-
ising precisely that, to the contrary, any interpretation of PA which requires
Aristotle’s particularisation to hold in \(N\) cannot be sound!
The postulation of an unspecified bound in a Bounded Arithmetic in order
to arrive at a provability-computability link thus appears dispensible.
The question then arises:

Does ‘weakening’ the PA Induction Axiom Schema yield any computa-
tional advantage?

Now, Buss considers a bounded arithmetic \(S_2\) which is, essentially, PA with
the following ‘weakened’ Induction Axiom Schema, \(\text{PIND}^{91}\):
\[
[\{f(0) \& (\forall x)(\exists y f(\lfloor \frac{x}{2} \rfloor) \rightarrow f(x))\} \rightarrow (\forall x)f(x)]
\]
However, \(\text{PIND}\) can be expressed in first-order Peano Arithmetic \(\text{PA}\) as follows:
\[
[\{f(0) \& (\forall x)(f(x) \rightarrow (f(2 \times x) \& f(2 \times x + 1)))\} \rightarrow (\forall x)f(x)].
\]
Moreover, the above is a particular case of \(\text{PIND}(k)\):
\[
[\{f(0) \& (\forall x)(f(x) \rightarrow (f(k \times x) \& f(k \times x + 1) \& \ldots \& f(k \times x + k - 1)))\} \rightarrow (\forall x)f(x)]
\]
Now we have the PA theorem:
\[
[(\forall x)f(x) \rightarrow \{f(0) \& (\forall x)(f(x) \rightarrow f(x + 1))\}]
\]
It follows that the following is also a PA theorem:
\[
[(\forall x)f(x) \rightarrow \{f(0) \& (\forall x)(f(x) \rightarrow (f(k \times x) \& f(k \times x + 1) \& \ldots \& f(k \times x + k - 1)))\}]
\]
In other words, for any numeral \([k]\), \(\text{PIND}(k)\) is equivalent in \(\text{PA}\) to the
standard Induction Axiom of \(\text{PA}\)!
Thus, the Provability Theorem for \(\text{PA}\) suggests that all arguments and con-
cclusions of a Bounded Arithmetic can be reflected in \(\text{PA}\) without any loss of
generality.

\section{Church’s Thesis is false}

One reason why efforts to prove \(P=NP\) remain unsuccessful may lie in recog-
nising that—contrary to accepted dogma\textsuperscript{92}—the term ‘effective computability’
can be precisely defined. For instance, we can define:

\textsuperscript{91}Where \(\lfloor \frac{x}{2} \rfloor\) denotes the largest natural number lower bound of the rational \(\frac{x}{2}\).
\textsuperscript{92}See \(\text{Ri54}\), p.300; \(\text{Me64}\), p.227; \(\text{Rg87}\), p.20; \(\text{EC89}\), p.85; \(\text{BBJ03}\), p23.

30
Definition 10 A number-theoretic function is effectively computable if, and only if, it is computable instantiationally.

Prima facie, this definition adequately captures our intuitive understanding of the term ‘effective computability’.

Now, classical theory argues that (standard results):

Lemma 30 Every Turing-computable function (or relation, treated as a Boolean function) \( F \) is partial recursive, and, if \( F \) is total, then \( F \) is recursive.

Lemma 31 Every partial recursive function (or relation, treated as a Boolean function) is Turing-computable.

It follows that the following—essentially unverifiable but refutable—theses are classically equivalent:

Standard Church’s Thesis A number-theoretic function (or relation, treated as a Boolean function) is effectively computable if, and only if, it is partial-recursive.

Standard Turing’s Thesis A number-theoretic function (or relation, treated as a Boolean function) is effectively computable if, and only if, it is Turing-computable.

8.1 The Church and Turing Theses are false

However, Church’s Thesis ignores the doctrine of Occam’s razor by postulating a strong identity—and not simply a weak equivalence—between an effectively computable number-theoretic function and some algorithmically computable function.

Consequently, Church’s Thesis (Turing’s Thesis) does not admit the possibility of an arithmetical function \( F \) that is computable instantiationally but not algorithmically. It follows that:

Theorem 8 The Church and Turing theses do not hold.

Proof By Corollary Gödel’s \([R(x)]\) is instantiationally computable as always true, but it is not algorithmically computable as always true. The lemma follows.

---

93 cf. [Me64], p.233, Corollary 5.13.
94 cf. [Me64], p.237, Corollary 5.15.
95 cf. [Me64], p.257.
96 Cf. Church’s (original) Thesis The effectively computable number-theoretic functions are the algorithmically computable number-theoretic functions.
97 cf. [Me64], p.227.
98 After describing what he meant by “computable” numbers in the opening sentence of his 1936 paper on Computable Numbers, Turing immediately expressed this thesis—albeit informally—as: “... the computable numbers include all numbers which could naturally be regarded as computable”.
99 cf. [BBJ03], p.33.
It is significant that Gödel (initially) and Church (subsequently—possibly under the influence of Gödel’s disquietude) enunciated Church’s formulation of ‘effective computability’ as a Thesis because Gödel was instinctively uncomfortable with accepting it as a definition that fully captures the essence of ‘intuitive effective computability’.

Gödel’s reservations seem vindicated if we accept that a number-theoretic function can be computable instantiationally, but not algorithmically.

The possibility that ‘truth’ may be be ‘effectively’ decidable instantiationally, but not algorithmically, is implicit in Gödel’s famous 1951 Gibbs lecture,[101] where he remarks:

“I wish to point out that one may conjecture the truth of a universal proposition (for example, that I shall be able to verify a certain property for any integer given to me) and at the same time conjecture that no general proof for this fact exists. It is easy to imagine situations in which both these conjectures would be very well founded. For the first half of it, this would, for example, be the case if the proposition in question were some equation \( F(n) = G(n) \) of two number-theoretical functions which could be verified up to very great numbers \( n \).”[102]

Such a possibility is also implicit in Turing’s remarks[103]:

“The computable numbers do not include all (in the ordinary sense) definable numbers. Let \( P \) be a sequence whose \( n \)-th figure is 1 or 0 according as \( n \) is or is not satisfactory. It is an immediate consequence of the theorem of §8 that \( P \) is not computable. It is (so far as we know at present) possible that any assigned number of figures of \( P \) can be calculated, but not by a uniform process. When sufficiently many figures of \( P \) have been calculated, an essentially new method is necessary in order to obtain more figures.”

The need for placing such a distinction on a formal basis has also been expressed explicitly on occasion[104]. Thus, Boolos, Burgess and Jeffrey[105] define a diagonal function, \( d \), any value of which can be decided effectively, although there is no single algorithm that can effectively compute \( d \).

Now, the straightforward way of expressing this phenomenon should be to say that there are well-defined number-theoretic functions that are instantiationally computable, but not algorithmically computable[106]. Yet, following Church
and Turing, such functions are labeled as effectively uncomputable\footnote{107}. “According to Turing’s Thesis, since \(d\) is not Turing-computable, \(d\) cannot be effectively computable. Why not? After all, although no Turing machine computes the function \(d\), we were able to compute at least its first few values. For since, as we have noted, \(f_1 = f_1 = f_1 = \) the empty function we have \(d(1) = d(2) = d(3) = 1\). And it may seem that we can actually compute \(d(n)\) for any positive integer \(n\)—if we don’t run out of time.”\footnote{108}

The reluctance to treat a function such as \(d(n)\)—or the function \(\Omega(n)\) that computes the \(n^{th}\) digit in the decimal expression of a Chaitin constant \(\Omega\)\footnote{109}, on the grounds that the ‘time’ needed to compute it increases monotonically with \(n\), is curious\footnote{110}; the same applies to any total Turing-computable function \(f(n)\)\footnote{111}.

9 Conclusions

I have defined what it means for a number-theoretic function to be:

(i) Instantiationally computable;

(ii) Algorithmically computable.

I then show that:

**Conclusion 1** If Aristotle’s particularisation is presumed valid over the structure \(N\) of the natural numbers—as is the case under the standard interpretation of PA—then it follows from the instantiational nature of the constructive definition of the Gödel \(\beta\)-function that a primitive recursive relation can be instantiationally equivalent to an arithmetical relation where the former is algorithmically computable as always true over \(N\) whilst the latter is instantiationally computable, but not algorithmically computable, as always true over \(N\).

I then show that:

**Conclusion 2** \(I_{PA(N, \text{Standard})}\) is sound if, and only if, Aristotle’s particularisation holds over \(N\); and the latter is the case if, and only if, \(PA\) is \(\omega\)-consistent.

\footnote{107}The issue here seems to be that, when using language to express the abstract objects of our individual, and common, mental ‘concept spaces’, we use the word ‘exists’ loosely in three senses, without making explicit distinctions between them (see\footnote{An97}).\footnote{108}\footnote{BBJ03}, p.37.

\footnote{109}Chaitin’s Halting Probability is given by \(0 < \Omega = \sum 2^{-\left|p\right|} < 1\), where the summation is over all self-delimiting programs \(p\) that halt, and \(\left|p\right|\) is the size in bits of the halting program \(p\); see\footnote{C75}.\footnote{110}The incongruity of this is addressed by Parikh in\footnote{Pa71}.

\footnote{111}The only difference being that, in the latter case, we know there is a common ‘program’ of constant length that will compute \(f(n)\) for any given natural number \(n\); in the former, we know we may need distinctly different programs for computing \(f(n)\) for different values of \(n\), where the length of the program will, sometime, reference \(n\).
Under the standard interpretation $\mathcal{I}_{PA(N, \text{Standard})}$ of PA over the domain $N$, if $[A]$ is an atomic formula $[A(x_1, x_2, \ldots, x_n)]$ of PA, then the sequence of natural numbers $(a_1, a_2, \ldots, a_n)$ satisfies $[A]$ if, and only if $A^*(a_1, a_2, \ldots, a_n)$ holds in $N$ and we presume that Aristotle’s particularisation is valid over $N$.

I have now shown that:

**Conclusion 3** We can define a sound interpretation $\mathcal{I}_{PA(N, \text{Algorithmic})}$ of PA over the domain $N$ where, if $[A]$ is an atomic formula $[A(x_1, x_2, \ldots, x_n)]$ of PA, then the sequence of natural numbers $(a_1, a_2, \ldots, a_n)$ satisfies $[A]$ if, and only if $[A(a_1, a_2, \ldots, a_n)]$ is algorithmically computable under $\mathcal{I}_{PA(N, \text{Algorithmic})}$ but we do not presume that Aristotle’s particularisation is valid over $N$.

It follows that:

**Conclusion 4** PA is consistent.

I have then shown that:

**Conclusion 5** (Provability Theorem for PA) A PA formula $[F(x)]$ is PA-provable if, and only if, $[F(x)]$ is algorithmically computable in $N$.

It follows that:

**Conclusion 6** PA is categorical.

In his 1931 paper, Gödel showed how to construct a formula $[R(x)]$ with a single free variable in any Peano Arithmetic such that:

If the first-order Peano Arithmetic PA is assumed to be consistent, then:

(i) for any PA numeral $[n]$, the PA formula $[R(n)]$ is provable in PA;
(ii) the PA formula $[R(x)]$ is not provable in PA.

I have now shown that:

**Conclusion 7** (a) We may conclude from Gödel’s argument that $[R(x)]$ is not algorithmically computable as true in $N$.
(b) $[\neg(\forall x)R(x)]$ is PA-provable;
(c) PA is not $\omega$-consistent;
(d) The ‘standard’ interpretation $\mathcal{I}_{PA(N, \text{Standard})}$ of PA over $N$ is not sound.

Since the above implies that Gödel’s formula $[R(x)]$ is computable instantiationally but not algorithmically as true over $N$, I conclude that:

**Conclusion 8** (i) $P \neq NP$;
(ii) the Church and Turing Theses do not hold.
10 Appendix A: Notation, Definitions and Comments

Notation In this investigation I use square brackets to indicate that the contents represent a symbol or a formula of a formal theory, generally assumed to be well-formed unless otherwise indicated by the context.

In other words, expressions inside the square brackets are to be only viewed syntactically as juxtaposition of symbols that are to be formed and manipulated upon strictly in accordance with specific rules for such formation and manipulation—in the manner of a mechanical or electronic device—without any regards to what the symbolism might represent semantically under an interpretation that gives them meaning.

Moreover, even though the formula ‘\(F(x)\)’ of a formal Arithmetic may interpret as the arithmetical relation expressed by ‘\(F(x)\)’, the formula ‘\((\exists x)R(x)\)’ need not interpret as the arithmetical proposition denoted by the usual abbreviation ‘\((\exists x)R^*(x)\).’ The latter denotes the phrase ‘There is some \(x\) such that \(R^*(x)\).’ As Brouwer had noted,\(^{112}\) this concept is not always capable of an unambiguous meaning that can be represented in a formal language by the formula ‘\((\exists x)R(x)\)’ which, in a formal language, is merely an abbreviation for the formula ‘\((\neg \forall x)\neg R(x)\)’.

By ‘expressed’ I mean here that the symbolism is simply a short-hand abbreviation for referring to abstract concepts that may, or may not, be capable of a precise ‘meaning’. Amongst these are symbolic abbreviations which are intended to express the abstract concepts—particularly those of ‘existence’—involved in propositions that refer to non-terminating processes and infinite aggregates.

Provability A formula \([F]\) of a formal system \(S\) is provable in \(S\) (\(S\)-provable) if, and only if, there is a finite sequence of \(S\)-formulas \([F_1],[F_2],\ldots,[F_n]\) such that \([F_n]\) is \([F]\) and, for all \(1 \leq i \leq n\), \([F_i]\) is either an axiom of \(S\) or a consequence of the axioms of \(S\), and the formulas preceding it in the sequence, by means of the rules of deduction of \(S\).

The structure \(\mathcal{N}\) The structure of the natural numbers—namely, \(\{N\) (the set of natural numbers); = (equality); ‘\(\) (the successor function); + (the addition function); * (the product function); 0 (the null element).\}

The axioms of first-order Peano Arithmetic (PA)

\begin{align*}
\text{PA}_1 & \quad ([x_1 = x_2] \rightarrow ((x_1 = x_3) \rightarrow (x_2 = x_3))]; \\
\text{PA}_2 & \quad ([x_1 = x_2] \rightarrow ([x_1] = [x_2]); \\
\text{PA}_3 & \quad \{0 \neq x\}; \\
\text{PA}_4 & \quad ([x'_1 = x'_2] \rightarrow (x_1 = x_2)); \\
\text{PA}_5 & \quad ([x_1 + 0] = x_1]; \\
\text{PA}_6 & \quad ([x_1 + x'_2] = ([x_1 + x_2]); \\
\text{PA}_7 & \quad ([x_1 \times 0] = 0]; \\
\text{PA}_8 & \quad ([x_1 \times x'_2] = ([x_1 \times x_2] + x_1); \\
\text{PA}_9 & \quad \text{For any well-formed formula } [F(x)] \text{ of PA:} \\
& \quad [F(0) \rightarrow (((\forall x)(F(x) \rightarrow F(x')))) \rightarrow (\forall x)F(x)].
\end{align*}

Generalisation in PA If \([A]\) is PA-provable, then so is \([(\forall x)A]\).

Modus Ponens in PA If \([A]\) and \([A \rightarrow B]\) are PA-provable, then so is \([B]\).

\(^{112}\) Brou; see also Andin.
Standard interpretation of PA

The standard interpretation $\mathcal{I}_{PA(N\text{ Standard})}$ of PA over the structure $N$ is the one in which the logical constants have their ‘usual’ interpretations in Aristotle’s logic of predicates and:

(a) the set of non-negative integers is the domain;
(b) the integer 0 is the interpretation of the symbol $[0]$;
(c) the successor operation (addition of 1) is the interpretation of the ‘$\prime$’ function;
(d) ordinary addition and multiplication are the interpretations of $[+]$ and $[*]$;
(e) the interpretation of the predicate letter $=[]$ is the identity relation.

Simple consistency

A formal system $S$ is simply consistent if, and only if, there is no $S$-formula $[F(x)]$ for which both $[(\forall x)F(x)]$ and $[\neg(\forall x)F(x)]$ are $S$-provable.

$\omega$-consistency

A formal system $S$ is $\omega$-consistent if, and only if, there is no $S$-formula $[F(x)]$ for which, first, $[\neg(\forall x)F(x)]$ is $S$-provable and, second, $[F(a)]$ is $S$-provable for any given $S$-term $[a]$.

Soundness (formal system)

A formal system $S$ is sound under an interpretation $\mathcal{I}_S$ if, and only if, every theorem $[T]$ of $S$ translates as ‘$T$’ is true under $\mathcal{I}_S$.

Soundness (interpretation)

An interpretation $\mathcal{I}_S$ of a formal system $S$ is sound if, and only if, $S$ is sound under the interpretation $\mathcal{I}_S$.

Soundness in classical logic

In classical logic, a formal system $S$ is sometimes defined as ‘sound’ if, and only if, it has an interpretation; and an interpretation is defined as the assignment of meanings to the symbols, and truth-values to the sentences, of the formal system. Moreover, any such interpretation is a model of the formal system. This definition suffers, however, from an implicit circularity: the formal logic $L$ underlying any interpretation of $S$ is implicitly assumed to be ‘sound’.

The above definitions seek to avoid this implicit circularity by delinking the defined ‘soundness’ of a formal system under an interpretation from the implicit ‘soundness’ of the formal logic underlying the interpretation. This admits the case where, even if $L_1$ and $L_2$ are implicitly assumed to be sound, $S + L_1$ is sound, but $S + L_2$ is not. Moreover, an interpretation of $S$ is now a model for $S$ if, and only if, it is sound.

Categoricity

A formal system $S$ is categorical if, and only if, it has a sound interpretation and any two sound interpretations of $S$ are isomorphic.

11 Appendix B: Gödel’s Theorem VII(2)

(Excerpted from [Go31] pp.29-31.)

Every relation of the form $x_0 = \phi(x_1, \ldots, x_n)$, where $\phi$ is recursive, is arithmetical and we apply complete induction on the rank of $\phi$. Let $\phi$ have rank $s(s > 1)$. 

\[ \phi(0, x_2, \ldots, x_n) = \psi(x_2, \ldots, x_n) \]

113 See [Me64], p.49.
114 Thus, Aristotle’s particularisation holds over $N$ in the standard interpretation of PA.
115 See [Me64], p.107.
116 My thanks to Professor Rohit Parikh for highlighting the need for making such a distinction explicit.
117 Compare [Me64], p.91.
\[ \phi(k+1, x_2, \ldots, x_n) = \mu[k, \phi(k, x_2, \ldots, x_n), x_2, \ldots, x_n] \]

(where \( \psi, \mu \) have lower rank than \( k \)).

... we apply the following procedure: one can express the relation \( x_0 = \phi(x_1, \ldots, x_n) \) with the help of the concept “sequence of numbers” \( f^{\text{II}} \) in the following manner:

\[
x_0 = \phi(x_1, \ldots, x_n) \sim (\exists f) \{ f_0 = \psi(x_2, \ldots, x_n) \& (\forall k)(k < x_1 \rightarrow f_{k+1} = \mu(k, f_k, x_2, \ldots, x_n) \& x_0 = f_{x_1} \}
\]

If \( S(y, x_2, \ldots, x_n), T(z, x_1, \ldots, x_{n+1}) \) are the arithmetical relations which, according to the inductive hypothesis, are equivalent to \( y = \psi(x_2, \ldots, x_n) \), and \( z = \mu(x_1, \ldots, x_{n+1}) \) respectively, then we have:

\[
x_0 = \phi(x_1, \ldots, x_n) \sim (\exists f) \{ S(f_0, x_2, \ldots, x_n) \& (\forall k)(k < x_1 \rightarrow T(f_{k+1}, k, x_2, \ldots, x_n) \& x_0 = f_{x_1} \}
\]

(17)

Now we replace the concept “sequence of numbers” by “pairs of numbers” by correlating with the number pair \( n, d \), the sequence of numbers \( f^{(n,d)} \): \( f^{(n,d)} = [n]_1 + (k+1)d \), where \([n]_p\) denotes the smallest non-negative remainder of \( n \) modulo \( p \).

Then:

Lemma 1: If \( f \) is an arbitrary sequence of natural numbers and \( k \) is an arbitrary natural number, then there exists a pair of natural numbers \( n, d \) such that \( f^{(n,d)} \) and \( f \) coincide in their first \( k \) terms.

Proof: Let \( l \) be the greatest of the numbers \( k, f_0, f_1, \ldots, f_{k-1} \). Determine \( n \) so that

\[ n \equiv f_i \ [\text{mod} \ (1 + (i + 1)!)] \]

for \( i = 0, 1, \ldots, k - 1 \), which is possible, since any two of the numbers \( 1 + (i + 1)! \ (i = 0, 1, \ldots, k - 1) \) are relatively prime. For, a prime dividing two of these numbers must also divide the difference \( (i_1 - i_2)! \) and therefore, since \( i_1 - i_2 < l \), must also divide \( l! \), which is impossible. The number pair \( n, l! \) fulfills our requirement.

Since the relation \( x = [n]_p \) is defined by

\[ x \equiv n \ (\text{mod} \ p) \& x < p \]

and is therefore arithmetical, then so also is the relation \( P(x_0, x_1, \ldots, x_n) \) defined as follows:

\[
P(x_0, x_1, \ldots, x_n) \equiv (\exists m, d) \{ S([n]_{d+1}, x_2, \ldots, x_n) \& (\forall k)(k < x_1 \rightarrow T([n]_{1+d(k+2)}, k, [n]_{1+d(k+1)}, x_2, \ldots, x_n] \& x_0 = [n]_{1+d(x+1)} \}
\]

which, according to (17) and Lemma 1, is equivalent to \( x_0 = \phi(x_1, \ldots, x_n) \) (in the sequence \( f \) in (17) only its values up to the \((x + 1)\)th term matter). Thus, Theorem VII(2) is proved.

Note: Gödel’s remark that “in the sequence \( f \) in (17) only its values up to the \((x + 1)\)th term matter” is significant for the resolution of the P=NP problem.

The proof of Lemma 10—and consequently of Theorem 10—that \( P \neq \text{NP} \) if the standard interpretation of \( \text{PA} \) is sound—depends upon the fact that the equivalence between \( f^{(n,d)} \) and \( f \) cannot be extended non-terminatingly.

\footnote{118} f denotes here a variable whose domain is the sequence of natural numbers. The \((k+1)\)st term of a sequence \( f \) is designated \( f_k \) (and the first, \( f_0 \)).
12 Appendix C: PA cannot admit a set-theoretical model

Let \([G(x)]\) denote the PA formula:

\[x = 0 \lor \neg(\forall y)(x = y')\]

This translates, under every unrelativised interpretation of PA, as:

If \(x\) denotes an element in the domain of an unrelativised interpretation of PA, either \(x\) is 0, or \(x\) is a ‘successor’.

Further, in every such interpretation of PA, if \(G(x)\) denotes the interpretation of \([G(x)]\):

(a) \(G(0)\) is true;
(b) If \(G(x)\) is true, then \(G(x')\) is true.

Hence, by Gödel’s completeness theorem:

(c) PA proves \([G(0)]\);
(d) PA proves \([G(x) \to G(x')]\).

Gödel’s Completeness Theorem: In any first-order predicate calculus, the theorems are precisely the logically valid well-formed formulas (i.e., those that are true in every model of the calculus).

Further, by Generalisation:

(e) PA proves \([(\forall x)(G(x) \to G(x'))]\);

Generalisation in PA: \([(\forall x)A]\) follows from \([A]\).

Hence, by Induction:

(f) \([(\forall x)G(x)]\) is provable in PA.

Induction Axiom Schema of PA: For any formula \([F(x)]\) of PA:

\([F(0) \to ((\forall x)(F(x) \to F(x')) \to (\forall x)F(x))]\]

In other words, except 0, every element in the domain of any unrelativised interpretation of PA is a ‘successor’. Further, \(x\) can only be a ‘successor’ of a unique element in any such interpretation of PA.

12.1 PA and ZF have no common model

Now, since Cantor’s first limit ordinal, \(\omega\), is not the ‘successor’ of any ordinal in the sense required by the PA axioms, and if there are no infinitely descending sequences of ordinals in a model—if any—of set-theory, PA and Ordinal Arithmetic cannot have a common model, and so we cannot consistently extend PA to ZF simply by the addition of more axioms.

12.1.1 Why PA has no set-theoretical model

We can define the usual order relation ‘<’ in PA so that every instance of the Induction Axiom schema, such as, say:

(i) \([F(0) \to ((\forall x)(F(x) \to F(x')) \to (\forall x)F(x))]\]

\[\text{cf. [Me64], p.261.}\]

\[\text{cf. [Me64], p.187.}\]
yields the PA theorem:

(ii) \[ F(0) \rightarrow ((\forall x)((\forall y)(y < x \rightarrow F(y)) \rightarrow F(x)) \rightarrow (\forall x)F(x)) \]

Now, if we interpret PA without relativisation in ZF in the sense indicated by Solomon Feferman\[121]\ — i.e., numerals as finite ordinals, \([x']\) as \([x \cup \{x\}\), etc. — then (ii) always translates in ZF as a theorem:

(iii) \[ F(0) \rightarrow ((\forall x)((\forall y)(y \in x \rightarrow F(y)) \rightarrow F(x)) \rightarrow (\forall x)F(x)) \]

However, (i) does not always translate similarly as a ZF-theorem (which is why PA and ZF can have no common model), since the following is not necessarily provable in ZF:

(iv) \[ F(0) \rightarrow ((\forall x)(F(x) \rightarrow F(x \cup \{x\})) \rightarrow (\forall x)F(x)) \]

Example: Define \([F(x)]\) as ‘\([x \in \omega]\)’.

A significant point which emerges from the above is that we cannot appeal unrestrictedly to set-theoretical reasoning when studying the foundational framework of PA.

Reason: The language of PA has no constant that interprets in any model of PA as the set \(N\) of all natural numbers.

Moreover, the preceding sections show that the Induction Axiom Schema of PA does not allow us to bypass this constraint by introducing an “actual” (or “completed”) infinity disguised as an arbitrary constant - usually denoted by \(c\) or \(\infty\) - into either the language, or a putative model, of PA.

References

[Be59] Evert W. Beth. 1959. The Foundations of Mathematics. Studies in Logic and the Foundations of Mathematics. Edited by L. E. J. Brouwer, E. W. Beth, A. Heyting. 1959. North Holland Publishing Company, Amsterdam.

[BBJ03] George S. Boolos, John P. Burgess, Richard C. Jeffrey. 2003. Computability and Logic (4th ed). Cambridge University Press, Cambridge.

[Br08] L. E. J. Brouwer. 1908. The Unreliability of the Logical Principles. English translation in A. Heyting, Ed. L. E. J. Brouwer: Collected Works 1: Philosophy and Foundations of Mathematics. Amsterdam: North Holland / New York: American Elsevier (1975): pp. 107-111.

[Br13] L. E. J. Brouwer. 1913. Intuitionism and Formalism. Inaugural address at the University of Amsterdam, October 14, 1912. Translated by Professor Arnold Dresden for the Bulletin of the American Mathematical Society, Volume 20 (1913), pp.81-96. 1999. Electronically published in Bulletin (New Series) of the American Mathematical Society, Volume 37, Number 1, pp.55-64.

[Br23] L. E. J. Brouwer. 1923. On the significance of the principle of the excluded middle in mathematics, especially in function theory. Address delivered on 21 September 1923 at the annual convention of the Deutsche Mathematiker-Vereinigung in Marburg an der Lahn. In Jean van Heijenoort. 1967. Ed. From Frege to Gödel: A source book in Mathematical Logic, 1878 - 1931. Harvard University Press, Cambridge, Massachusetts.

[Br27] L. E. J. Brouwer. 1927. Intuitionistic reflections on formalism. In Jean van Heijenoort. 1967. Ed. From Frege to Gödel: A source book in Mathematical Logic, 1878 - 1931. Harvard University Press, Cambridge, Massachusetts.

\[121\]
