The Holy Grail of Arithmetic: Bridging Provability and Computability


The Holy Grail of Arithmetic: Bridging Provability and Computability

See also this update.

(Notations, non-standard concepts, and definitions used commonly in these investigations are detailed in this post.)

Peter Wegner and Dina Goldin

In a short opinion paper, 'Computation Beyond Turing Machines', Computer Scientists Peter Wegner and Dina Goldin (Wg03) advanced the thesis that:

'A paradigm shift is necessary in our notion of computational problem solving, so it can provide a complete model for the services of today’s computing systems and software agents.'

We note that Wegner and Goldin’s arguments, in support of their thesis, seem to reflect an extraordinarily eclectic view of mathematics, combining both an implicit acceptance of, and implicit frustration at, the standard interpretations and dogmas of classical mathematical theory:

(i) ‘… Turing machines are inappropriate as a universal foundation for computational problem solving, and … computer science is a fundamentally non-mathematical discipline.’

(ii) ‘(Turing’s) 1936 paper … proved that mathematics could not be completely modeled by computers.’

(iii) ‘… the Church-Turing Thesis … equated logic, lambda calculus, Turing machines, and algorithmic computing as equivalent mechanisms of problem solving.’

(iv) ‘Turing implied in his 1936 paper that Turing machines … could not provide a model for all forms of mathematics.’

(v) ‘… Gödel had shown in 1931 that logic cannot model mathematics … and Turing showed that neither logic nor algorithms can completely model computing and human thought.’

These remarks vividly illustrate the dilemma with which not only Theoretical Computer Sciences, but all applied sciences that depend on mathematics—for providing a verifiable language to express their observations precisely—are faced:

Query: Are formal classical theories essentially unable to adequately express the extent and range of human cognition, or does the problem lie in the way formal theories are classically interpreted at the moment?

The former addresses the question of whether there are absolute limits on our capacity to express human cognition unambiguously; the latter, whether there are only temporal limits—not necessarily absolute—to the capacity of classical interpretations to communicate unambiguously that which we intended to capture within our formal expression.

Prima facie, applied science continues, perforce, to interpret mathematical concepts platonically, whilst waiting for mathematics to provide suitable, and hopefully reliable, answers as to how best it may faithfully express its observations verifiably.

Lance Fortnow

This dilemma is also reflected in Computer Scientist Lance Fortnow’s on-line rebuttal of
Can we think about the infinite in a consistent way, and would such a concept be helpful?

The Unexplained Intellectual

Bridging the Finite-Infinite Divide? Should the foundations of mathematics be arithmetical and not set-theoretical?

Three intriguing non-heuristic prime counting function queries

The Evidence-Based Argument for Lucas’ Gödelian Thesis

Generating primes sequentially

The Prime Number Theorem

A Highly Speculative Post About Speculation: Defining Market Equilibrium in a Simplified Stock Exchange

What differentiates the Finitary Logic of these investigations from Classical Hilbertian Logics and Intuitionistic Brouwerian Logics?

Why we need to define Effective Computability formally and weaken the Church and Turing Theses – II

Why we need to define Effective Computability formally and weaken the Church and Turing Theses – I

Why we shouldn’t fault J. R. Lucas and Roger Penrose for their Gödelian arguments against computationalism – II

The case against non-standard models of PA – II

The Holy Grail of Arithmetic: Bridging Provability and Computability

PA is finitarily consistent: A solution to the Second of Hilbert’s Twenty Three Problems

Misunderstanding Gödel: The significance of Feynman’s cover-up factor

Meeting Wittgenstein’s requirement of ‘truth’ in Gödel’s formal reasoning

The PvNP Separation Problem

Notation, non-standard concepts and definitions used in these investigations

The reals are denumerable: The finitary solution that Hilbert probably would not have foreseen for the First of his Twenty Three problems!

How mechanical intelligences and human intelligences can reason in contradictory, yet complementary, ways!

Mathematics does not need a new foundation; it only needs to make its implicit assumptions explicit!

Does mathematics need a new foundation?

Beyond Hawkings’ philosophical pronouncement

Is philosophy really dead?

Do we want our Giants to be our step-ladders or our white canes?

From perfect numbers to a generating function for the unrestricted factorisations of an integer

A suggested mathematical perspective for the EPR argument

Why we shouldn’t fault J. R. Lucas and Roger Penrose for their Gödelian arguments against computationalism – I

The Mechanist’s Challenge to John R. Lucas

Can someone tell me what is so special about Rosser’s proof of formally undecidable arithmetical propositions?

Placing Cohen’s proof of the Independence of the Axiom of Choice...
in perspective

Why Brouwer was justified in his objection to Hilbert's unqualified interpretation of quantification

A foundational perspective on the semantic and logical paradoxes – IV
A foundational perspective on the semantic and logical paradoxes – III
The Butterfly Effect
A foundational perspective on the semantic and logical paradoxes – II
A foundational perspective on the semantic and logical paradoxes – I
The case against Goodstein’s Theorem – IX
The case against Goodstein’s Theorem – VIII
The case against Goodstein’s Theorem – VII
The case against Goodstein’s Theorem – VI
The case against Goodstein’s Theorem – V
A charming glimpse into the mind of a master rigourist: Professor Yehezkel-Edmund Landau
The case against Goodstein’s Theorem – IV
The case against Goodstein’s Theorem – III
The case against Goodstein’s Theorem – II
The case against Goodstein’s Theorem – I
Which is the canonical model of PA?
The case against non-standard models of PA – I
Is Gödel’s undecidable proposition an ‘ad hoc’ anomaly?
Let not posterity judge us as having spent our lives polishing the pebbles and tarnishing the diamonds

math - update
George Lakoff
LobeLog
What’s new
computingclouds.wordpress.com/
Quanta Magazine
The Brains Blog
Logic Matters
A Neighborhood of Infinity
Inquiry Into Inquiry
Combinatorics and more
Mathematics and Computation
Foundations of Mathematics, Logic & Computability
John D. Cook
Shetei-Optimized
Nanoequations
Eric Cavalcanti
East Asia Forum
Azimuth
Tanya Khovanova’s Math Blog

computational theory.

For instance, in an arXived paper *Passages of Proof*, Computer Scientists Cristian Calude, Elena Calude and Solomon Marcus remark that:

> “Classically, there are two equivalent ways to look at the mathematical notion of proof: logical, as a finite sequence of sentences strictly obeying some axioms and inference rules, and computational, as a specific type of computation. Indeed, from a proof given as a sequence of sentences one can easily construct a Turing machine producing that sequence as the result of some finite computation and, conversely, given a machine computing a proof we can just print all sentences produced during the computation and arrange them into a sequence.”

In other words, the authors seem to hold that Turing-computability of a ‘proof’, in the case of an arithmetical proposition, is equivalent to provability of its representation in PA.

Wilfrid Sieg

We now attempt to build such a bridge formally, which is essentially one between the arithmetical ‘Decidability and Calculability’ described by Philosopher Wilfrid Sieg in his in-depth and wide-ranging survey *On Computation*, in which he addresses Gödel’s lifelong belief that an iff bridge between the two concepts is ‘impossible’ for ‘the whole calculus of predicates’ (Wi08, p.602).

**§2 Bridging provability and computability: The foundations**

In the paper titled “Evidence-Based Interpretations of P” that was presented to the Symposium on Computational Philosophy at the AISB/IACAP World Congress 2012-Alan Turing 2012, held from 24-29 July 2012 at the University of Birmingham, UK (reproduced in this post) we have defined what it means for a number-theoretic function to be:

(i) Algorithmically verifiable;
(ii) Algorithmically computable.

We have shown here that:

(i) The standard interpretation $\mathcal{I}_{PA(N, \text{standard})}$ of the first order Peano Arithmetic PA is finitarily sound if, and only if, Aristotle’s particularisation holds over $\mathcal{N}$; and the latter is the case if, and only if, PA is $\omega$-consistent.

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

(iii) The axioms of PA are always true under the finitary interpretation $\mathcal{I}_{PA(N, \text{Agoric})}$ and the rules of inference of PA preserve the properties of satisfaction/truth under $\mathcal{I}_{PA(N, \text{Agoric})}$.

We concluded that:

**Theorem 1**: The interpretation $\mathcal{I}_{PA(N, \text{Agoric})}$ of PA is finitarily sound.

**Theorem 2**: PA is consistent.

**§3 Extending Buss’ Bounded Arithmetic**

One of the more significant consequences of the Birmingham paper is that we can extend the iff bridge between the domain of provability and that of computability envisaged under Buss’ Bounded Arithmetic by showing that an arithmetical formula $[F]$ is PA-provable if, and only if, $[F]$ interprets as true under an algorithmic interpretation of PA.

**§4 A Provability Theorem for PA**

We first show that PA can have no non-standard model (for a distinctly different proof of this convention-challenging thesis see this post and this paper), since it is ‘arithmetically’ complete in the sense that:

**Theorem 3**: (Provability Theorem for PA) A PA formula $[F(x)]$ is PA-provable if, and only if, $[F(x)]$ is algorithmically computable as always true in $\mathcal{N}$.

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

Since $\mathcal{I}_{PA(N, \text{Agoric})}$ is finitarily sound, it defines a finitary model of PA over $\mathcal{N}$—say $\mathcal{M}_{PA(\omega)}$—such that:

If $[(\forall x)[F(x)]]$ is PA-provable, then $[F(x)]$ is algorithmically computable as always true in $\mathcal{N}$.

If $[\neg(\forall x)[F(x)]]$ is PA-provable, then it is not the case that $[F(x)]$ is algorithmically computable as always true 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:

(i) There is a finitary model—say $\mathcal{M}_1$—of PA—$[(\forall x)[F(x)]]$ in which $[F(x)]$ is algorithmically computable as always true in $\mathcal{N}$.  

In such case, if
To see that (as Brouwer steadfastly held) this may not always be the case, interpret
sound implies that Aristotle's particularisation holds over the natural numbers under any finitarily
Reason:
finitarily
some numeral
Bounded Arithmetic that, from a proof of
Now, one difference
question arises:
Since we have proven such a Provability Theorem for PA in the previous section, the first
decides the
the Bounded Arithmetic if, and only if, there is an algorithm that, for any given numeral
a Bounded Arithmetic and Computability so that a
Presumably Buss' intent—as expressed below—is to build an iff bridge between provability in
number-theoretic arguments
particularisation over
Since formal quantification is currently interpreted in classical logic
The Provability Theorem for PA and Bounded Arithmetic
In a 1997 paper [11], Samuel R. Buss considered Bounded Arithmetics obtained by:
(a) limiting the applicability of the Induction Axiom Schema in PA only to functions with
quantifiers bounded by an unspecified natural number bound
(b) 'weakening' the statement of the axiom with the aim of differentiating between
effective computability over the sequence of natural numbers, and feasible 'polynomial-
time' computability over a bounded sequence of the natural numbers [12].
Presumably Buss' intent—as expressed below—is to build an iff bridge between provability in a
Bounded Arithmetic and Computability so that a \( \forall x f(x) \) 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 \( \Delta_k \) formula \( [f(n)] \) as 'true':
If \( \forall x [f(x)] \) is provable, then there should be an algorithm to finitely as a
function of \( x \).
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 [14] 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 \([n]\) such that \([f(n)]\) is provable in the arithmetic; however, this is not a finitarily 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 finitarily sound interpretation of PA.
To see that (as Brouwer steadfastly held) this may not always be the case, interpret
\([\forall x] f(x)\) as [15].
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'.

Reason: $\neg (\forall y)(\exists x) f(y, x)$ 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 $\neg (\forall y) f(n, y)$ PA-provable for any given numeral $n$.

Presumably it is the belief that any finitarily sound interpretation of PA requires Aristotle’s particularisation to hold in $\mathcal{N}$, 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 recognising precisely that, to the contrary, any interpretation of PA which requires Aristotle’s particularisation to hold in $\mathcal{N}$ cannot be finitarily 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:

*Is Does ‘weakening’ the PA Induction Axiom Schema yield any computational advantage?*

Now, Buss considers a bounded arithmetic $\mathcal{B}$, which is, essentially, PA with the following ‘weakened’ Induction Axiom Schema, $\text{PIND}$ [16]:

$$\{ f(0) \land (\forall x)(f(x) \rightarrow f(x+1)) \rightarrow (\forall x)f(x) \}$$

However, $\text{PIND}$ can be expressed in first-order Peano Arithmetic PA as follows:

$$\{ f(0) \land (\forall x)(f(x) \rightarrow f(2x+1)) \rightarrow (\forall x)f(x) \}$$

Moreover, the above is a particular case of $\text{PIND}[(k)]$:

$$\{ f(0) \land (\forall x)(f(x) \rightarrow f(kx+1) \land f(kx+2) \land \ldots \land f(kx+k-1)) \}$$

Now we have the PA theorem:

$$[(\forall x)f(x) \rightarrow \{ f(0) \land (\forall x)(f(x) \rightarrow f(x+1)) \}]$$

It follows that the following is also a PA theorem:

$$\{ f(0) \land (\forall x)(f(x) \rightarrow f(x+1)) \rightarrow \{ f(0) \land (\forall x)(f(x) \rightarrow f(kx+1) \land f(kx+2) \land \ldots \land f(kx+k-1)) \} \}$$

In other words, for any numeral $[k]$, $\text{PIND}[(k)]$ is equivalent in PA to the standard Induction Axiom of PA!

Thus, the Provability Theorem for PA suggests that all arguments and conclusions of a Bounded Arithmetic can be reflected in PA without any loss of generality.

**References**


Author’s working archives & abstracts of investigations
The hypothesis formulated by Georg Friedrich Bernhard Riemann in 1859, according to Marcus du Sautoy of Oxford University, is the holy grail of mathematics. “Most mathematicians would trade their soul with Mephistopheles for a proof,” he said. The Riemann hypothesis would explain the apparently random pattern of prime numbers - numbers such as 3, 17 and 31, for instance, are all prime numbers: they are divisible only by themselves and one. Prime numbers are the atoms of arithmetic. They are also the key to internet cryptography: in effect they keep banks safe and credit cards secure. Bayes’ Theorem is perhaps the most important theorem in the field of mathematical statistics and probability theory. For this reason, the theorem finds its use very often in the field of data...
Overall the book is clearly written and well organized, and it contains interesting selections from the writings of prominent figures in the foundations of mathematics throughout. Adjoined to the end is a 25 page timeline, surveying 1834-1970. This is quite neat, but I wish it had been longer. The book should be useful both to people new to computability & logic, as well as those with some previous background, but the target audience is probably those with an interest in philosophy of mathematics. 1) Beginners- Its helpful to learn computability theory and logic together.