Foundations of Mathematics, Logic & Computability

Reviewing classical interpretations of Cantor's, Gödel's, Tarski's, and Turing's reasoning and addressing some grey areas in the foundations of mathematics, logic and computability

The Holy Grail of Arithmetic: Bridging Provability and Computability

November 11, 2013 in Foundations of mathematics, logic and computation, Philosophy | Tags: algorithmically computable, algorithmically verifiable, Aristotle's particularisation, arithmetic, Bringsjord, Church Turing Thesis, completed infinity, computationalism, consistency, constructive, finitary, first order, FOL, formal, formal language, Fortnow, foundations, Gödel, Gödelian argument, Goldin, Hilbert, human, human intelligence, logic, mathematical truth, mathematics, mechanical intelligence, mechanist, Mendelson, natural numbers, omega-consistency, PA, Peano Arithmetic, Peano Arithmetic PA, Peter Wegner, philosophy, reasoning, Rosser, Rule C, science, semantic, simple functional language, standard interpretation, syntactic, Tarski, Turing, Turing machine, Turing machines, undecidable, unpredictability, Wegner, Wilfrid Sieg

§1 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

Wegner and Goldin's thesis, and of their reasoning.

Thus Fortnow divides his faith between the standard interpretations of classical mathematics (and, possibly, the standard set-theoretical models of formal systems such as standard Peano Arithmetic), and the classical computational theory of Turing machines.

He relies on the former to provide all the proofs that matter:

> `Not every mathematical statement has a logical proof, but logic does capture everything we can prove in mathematics, which is really what matters';

and, on the latter to take care of all essential, non-provable, truth:

> `… what we can compute is what computer science is all about'.

Can faith alone suffice?

However, as we shall argue in a subsequent post, Fortnow's faith in a classical Church-Turing Thesis that ensures:

> `… Turing machines capture everything we can compute',

may be as misplaced as his faith in the infallibility of standard interpretations of classical mathematics.

The reason: There are, prima facie, reasonably strong arguments for a Kuhnian (Ku62) paradigm shift; not, as Wegner and Goldin believe, in the notion of computational problem solving, but in the standard interpretations of classical mathematical concepts.

However, Wegner and Goldin could be right in arguing that the direction of such a shift must be towards the incorporation of non-algorithmic effective methods into classical mathematical theory (as detailed in the Birmingham paper); presuming, from the following remarks, that this is, indeed, what `external interactions' are assumed to provide beyond classical Turing-computability:

> (vi) `… that Turing machine models could completely describe all forms of computation … contradicted Turing's assertion that Turing machines could only formalize algorithmic problem solving … and became a dogmatic principle of the theory of computation'.
>
> (vii) `… interaction between the program and the world (environment) that takes place during the computation plays a key role that cannot be replaced by any set of inputs determined prior to the computation'.
>
> (viii) `… a theory of concurrency and interaction requires a new conceptual framework, not just a refinement of what we find natural for sequential [algorithmic] computing'.
>
> (ix) `… the assumption that all of computation can be algorithmically specified is still widely accepted'.

A widespread notion of particular interest, which seems to be recurrently implicit in Wegner and Goldin's assertions too, is that mathematics is a dispensable tool of science, rather than its indispensable mother tongue.

Elliott Mendelson

However, the roots of such beliefs may also lie in ambiguities, in the classical definitions of foundational elements, that allow the introduction of non-constructive—hence non-verifiable, non-computational, ambiguous, and essentially Platonic—elements into the standard interpretations of classical mathematics.

For instance, in a 1990 philosophical reflection, Elliott Mendelson's following remarks (in Me90; reproduced from Selmer Bringsjord (Br93)), implicitly imply that classical definitions of various foundational elements can be argued as being either ambiguous, or non-constructive, or both:

> `Here is the main conclusion I wish to draw: it is completely unwarranted to say that CT is unprovable just because it states an equivalence between a vague, imprecise notion (effectively computable function) and a precise mathematical notion (partial-recursive function). … The concepts and assumptions that support the notion of partial-recursive function are, in an essential way, no less vague and imprecise than the notion of effectively computable function; the former are just more familiar and are part of a respectable theory with connections to other parts of logic and mathematics. (The notion of effectively computable function could have been incorporated into an axiomatic presentation of classical mathematics, but the acceptance of CT made this unnecessary.) … Functions are defined in terms of sets, but the concept of set is no clearer than that of function and a foundation of mathematics can be based on a theory using function as primitive notion instead of set. Tarski's definition of truth is formulated in set-theoretic terms, but the notion of set is no clearer than that of truth. The model-theoretic definition of logical validity is based ultimately on set theory, the foundations of which are no clearer than our intuitive understanding of logical validity. … The notion of Turing-computable function is no clearer than, nor more mathematically useful (foundationally speaking) than, the notion of an effectively computable function.'

Consequently, standard interpretations of classical theory may, inadvertently, be weakening a desirable perception—of mathematics as the lingua franca of scientific expression—by ignoring the possibility that, since mathematics is, indeed, indisputably accepted as the language that most effectively expresses and communicates intuitive truth, the chasm between formal truth and provability must, of necessity, be bridgeable.

Cristian Calude, Elena Calude and Solomon Marcus

The belief in the existence of such a bridge is occasionally implicit in interpretations of

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

BLOGS I FOLLOW

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
Shtetl-Optimized
Nanoexplanations
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 Comptability', 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).

$\S 2$ Bridging provability and computability: The foundations

In the paper titled "Evidence-Based Interpretations of PA" that was presented to the Symposium on Computational Philosophy at the AISB/IACAP World Congress 2012-Alan Turing 2012, held from 2^{nd} to 6^{th} 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 there that:

> (i) The standard interpretation $\mathcal{I}_{PA(N,\ Standard)}$ of the first order Peano Arithmetic PA is finitarily sound if, and only if, Aristotle's particularisation holds over N; and the latter is the case if, and only if, PA is ω-consistent.
>
> (ii) We can define a finitarily sound algorithmic interpretation $\mathcal{I}_{PA(N,\ Algorithmic)}$ 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 $\mathcal{I}_{PA(N,\ Algorithmic)}$, but we do not presume that Aristotle's particularisation is valid over N.
>
> (iii) The axioms of PA are always true under the finitary interpretation $\mathcal{I}_{PA(N,\ Algorithmic)}$, and the rules of inference of PA preserve the properties of satisfaction/truth under $\mathcal{I}_{PA(N,\ Algorithmic)}$.

We concluded that:

> **Theorem 1:** The interpretation $\mathcal{I}_{PA(N,\ Algorithmic)}$ of PA is finitarily sound.
>
> **Theorem 2:** PA is consistent.

$\S 3$ 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.

$\S 4$ 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 `algorithmically' 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 N.
>
> **Proof:** We have by definition that $[(\forall x)F(x)]$ interprets as true under the interpretation $\mathcal{I}_{PA(N,\ Algorithmic)}$ if, and only if, $[F(x)]$ is algorithmically computable as always true in N.
>
> Since $\mathcal{I}_{PA(N,\ Algorithmic)}$ is finitarily sound, it defines a finitary model of PA over N—say $\mathcal{M}_{PA(\beta)}$—such that:
>
> > If $[(\forall x)F(x)]$ is PA-provable, then $[F(x)]$ is algorithmically computable as always true in 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 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 $M1_{\beta}$—of PA+$[(\forall x)F(x)]$ in which $[F(x)]$ is algorithmically computable as always true in N.

(ii) There is a finitary model—say $M2_{\beta}$—of PA+$[\neg(\forall x)F(x)]$ in which it is not the case that $[F(x)]$ is algorithmically computable as always true in N.

The lemma follows. $\Box$

$\S5$ **The holy grail of arithmetic**

We thus have that:

Corollary 1: PA is categorical finitarily.

Now we note that:

Lemma 2: If PA has a sound interpretation $\mathcal{I}_{PA(N,\ Sound)}$ over N, then there is a PA formula $[F]$ which is algorithmically verifiable as always true over N under $\mathcal{I}_{PA(N,\ Sound)}$ even though $[F]$ is not PA-provable.

Proof In his seminal 1931 paper on formally undecidable arithmetical propositions, Kurt Gödel has shown how to construct an arithmetical formula with a single variable—say $[R(x)]$ [1]—such that $[R(x)]$ is not PA-provable [2], but $[R(n)]$ is instantiationally PA-provable for any given PA numeral $[n]$. Hence, for any given numeral $[n]$, the PA formula $xB\lceil[R(n)]\rceil$ must hold for some x. The lemma follows. $\Box$

By the argument in Theorem 3 it follows that:

Corollary 2: The PA formula $[\neg(\forall x)R(x)]$ defined in Lemma 2 is PA-provable.

Corollary 3: Under any sound interpretation of PA, Gödel's $[R(x)]$ interprets as an algorithmically verifiable, but not algorithmically computable, tautology over N.

Proof Gödel has shown that $[R(x)]$ [3] interprets as an algorithmically verifiable tautology [4]. By Corollary 2 $[R(x)]$ is not algorithmically computable as always true in N. $\Box$

Corollary 4: PA is *not* ω-consistent. [5]

Proof Gödel has shown that if PA is consistent, then $[R(n)]$ is PA-provable for any given PA numeral $[n]$ [6]. By Corollary 2 and the definition of ω-consistency, if PA is consistent then it is *not* ω-consistent. $\Box$

Corollary 5: The standard interpretation $\mathcal{I}_{PA(N,\ Standard)}$ of PA is not finitarily sound, and does not yield a finitary model of PA [7].

Proof If PA is consistent but not ω-consistent, then Aristotle's particularisation does not hold over N. Since the `standard', interpretation of PA appeals to Aristotle's particularisation, the lemma follows. $\Box$

Since formal quantification is currently interpreted in classical logic[8] so as to admit Aristotle's particularisation over N as axiomatic [9], the above suggests that we may need to review number-theoretic arguments [10] that appeal unrestrictedly to classical Aristotlean logic.

$\S6$ **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;

(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 Π_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 [13].

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

$\S7$ **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 $[m]$ such that $[f(n, m)]$ 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: $[(\exists y)f(n, y)]$ 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-*un*provable, but $[(\exists y)f(n, y)]$ 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 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 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:

§8 Does `weakening' the PA Induction Axiom Schema yield any computational advantage?

Now, Buss considers a bounded arithmetic S_2 which is, essentially, PA with the following `weakened' Induction Axiom Schema, PIND [16]:

$$[\{f(0) \ \& \ (\forall x)(f(\lfloor \tfrac{x}{2} \rfloor) \rightarrow f(x))\} \rightarrow (\forall x)f(x)]$$

However, PIND can be expressed in first-order Peano Arithmetic PA as follows:

$$[\{f(0) \ \& \ (\forall x)(f(x) \rightarrow (f(2*x) \ \& \ f(2*x+1)))\} \rightarrow (\forall x)f(x)].$$

Moreover, the above is a particular case of PIND(k):

$$[\{f(0) \ \& \ (\forall x)(f(x) \rightarrow (f(k*x) \ \& \ f(k*x+1) \ \& \ \ldots \ \& \ f(k*x+k-1)))\} \\ \rightarrow (\forall x)f(x)].$$

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:

$$[\{f(0) \ \& \ (\forall x)(f(x) \rightarrow f(x+1))\} \rightarrow \\ \{f(0) \ \& \ (\forall x)(f(x) \rightarrow (f(k*x) \ \& \ f(k*x+1) \ \& \ \ldots \ \& \ f(k*x+k-1)))\}]$$

In other words, for any numeral $[k]$, 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

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

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.

Bu97 Samuel R. Buss. 1997. *Bounded Arithmetic and Propositional Proof Complexity.* In *Logic of Computation.* pp.67-122. Ed. H. Schwichtenberg. Springer-Verlag, Berlin.

CCS01 Cristian S. Calude, Elena Calude and Solomon Marcus. 2001. *Passages of Proof.* Workshop, Annual Conference of the Australasian Association of Philosophy (New Zealand Division), Auckland. Archived at: http://arxiv.org/pdf/math/0305213.pdf. Also in EATCS Bulletin, Number 84, October 2004, viii+258 pp.

Da82 Martin Davis. 1958. *Computability and Unsolvability.* 1982 ed. Dover Publications, Inc., New York.

EC89 Richard L. Epstein, Walter A. Carnielli. 1989. *Computability: Computable Functions, Logic, and the Foundations of Mathematics.* Wadsworth & Brooks, California.

Go31 Kurt Gödel. 1931. *On formally undecidable propositions of Principia Mathematica and related systems I.* Translated by Elliott Mendelson. In M. Davis (ed.). 1965. *The Undecidable.* Raven Press, New York.

HA28 David Hilbert & Wilhelm Ackermann. 1928. *Principles of Mathematical Logic.* Translation of the second edition of the *Grundzüge Der Theoretischen Logik.* 1928. Springer, Berlin. 1950. Chelsea Publishing Company, New York.

He04 Catherine Christer-Hennix. 2004. *Some remarks on Finitistic Model Theory, Ultra-Intuitionism and the main problem of the Foundation of Mathematics.* ILLC Seminar, 2nd April

2004, Amsterdam.

Hi25 David Hilbert. 1925. *On the Infinite.* Text of an address delivered in Münster on 4th June 1925 at a meeting of the Westphalian Mathematical Society. In Jean van Heijenoort. 1967.Ed. *From Frege to Gödel: A source book in Mathematical Logic, 1878 – 1931.*Harvard University Press, Cambridge, Massachusetts.

Ku62 Thomas S. Kuhn. 1962. *The structure of Scientific Revolutions.* 2nd Ed. 1970. University of Chicago Press, Chicago.

Me90 Elliott Mendelson. 1990. *Second Thoughts About Church's Thesis and Mathematical Proofs.* In *Journal of Philosophy* 87.5.

Pa71 Rohit Parikh. 1971. *Existence and Feasibility in Arithmetic.* In *The Journal of Symbolic Logic,>i> Vol.36, No. 3 (Sep., 1971), pp. 494-508.*

Rg87 Hartley Rogers Jr. 1987. *Theory of Recursive Functions and Effective Computability.* MIT Press, Cambridge, Massachusetts.

Ro36 J. Barkley Rosser. 1936. *Extensions of some Theorems of Gödel and Church.* In M. Davis (ed.). 1965. *The Undecidable.* Raven Press, New York. Reprinted from *The Journal of Symbolic Logic.* Vol.1. pp.87-91.

Si08 Wilfrid Sieg. 2008. *On Computability* in *Handbook of the Philosophy of Science. Philosophy of Mathematics.* pp.525-621. Volume Editor: Andrew Irvine. General Editors: Dov M. Gabbay, Paul Thagard and John Woods. Elsevier BV. 2008.

WG03 Peter Wegner and Dina Goldin. 2003. *Computation Beyond Turing Machines.* Communications of the ACM, 46 (4) 2003.

Notes

Return to 1: Gödel refers to this formula only by its Gödel number r (Go31, p.25(12)).

Return to 2: Gödel's immediate 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.

Return to 3: Gödel refers to this formula only by its Gödel number r (Go31, p.25, eqn.12).

Return to 4: Go31, p.26(2): '$(n)\neg(nB_{\kappa}(17Gen\ r))$ holds".

Return to 5: 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 ω-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 Gödel's incompleteness theorem".

Return to 6: Go31, p.26(2).

Return to 7: I note that finitists of all hues—ranging from Brouwer Br08 to Alexander Yessenin-Volpin He04—have persistently questioned the finitary soundness of the `standard' interpretation $\mathcal{I}_{PA(N,\ Standard)}$.

Return to 8: See Hi25, p.382; HA28, p.48; Be59, pp.178 \& 218.

Return to 9: In the sense of being intuitively obvious. See, for instance, Da82, p.xxiv; Rg87, p.308 (1)-(4); EC89, p.174 (4); BBJ03, p.102.

Return to 10: For instance Rosser's construction of an undecidable arithmetical proposition in PA (see Ro36)—which does not explicitly assume that PA is ω-consistent—implicitly presumes that Aristotle's particularisation holds over N.

Return to 11: Bu97.

Return to 12: See also Pa71.

Return to 13: See Bu97.

Return to 14: We suspect the only one.

Return to 15: We have seen in the earlier sections that such an interpretation is finitarily sound.

Return to 16: Where $\lfloor \frac{x}{2} \rfloor$ denotes the largest natural number lower bound of the rational $\frac{x}{2}$.

Author's working archives & abstracts of investigations

SHARE THIS:

Facebook Email LinkedIn Google

Loading...

2 comments

Comments feed for this article

November 22, 2013 at 4:45 am
logiciel pour r4
Many thanks….for good site..!!!
Reply

logiciel pour r4
This is a extremely great blog submit. I favored to work with you.

November 24, 2013 at 8:26 am
Bhupinder Singh Anand

Thanks and welcome.

Reply

LEAVE A REPLY

Enter your comment here...

RECENT COMMENTS

Abel Peralta on Can Gödel be held guilty of im…

Bhupinder Singh Anan… on Can Gödel be held guilty of im…

Abel Peralta on Can Gödel be held guilty of im…

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

The case against non-standard models of PA – II »

Bhupinder Singh Anan… on Why Gödel's Theorem…

Abel Peralta on Why Gödel's Theorem…

BLOGROLL

- Mathematical Foundations for the Computer Sciences
- Wilful Blindness
- WordPress.com
- WordPress.org

Computability: Computable has been added to your Cart. Add to Cart. Buy Now.Â 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.