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 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
The belief in the existence of such a bridge is occasionally implicit in interpretations of
quantum behaviour? The PvNP problem: Why we need to review current approaches towards
determining the computational complexity of Integer Factoring
Can we think about the infinite in a consistent way, and would such a concept be helpful?
The Unexplained Intellect
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
Integer Factoring is not in P
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 – I
Why we need to define Effective Computability formally and weaken the Church and Turing Theses – II
Why we shouldn’t fault J. R. Lucas and Roger Penrose for their Gödelian arguments against computationalism – I
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 PwNP 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 Hawking’s 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 – II
The Mechanist’s Challenge to John R. Lucas
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 … contradiicted 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, or 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
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 rigorist: 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

...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.

Wilfried Sieg

We now attempt to build such a bridge formally, which is essentially one between the arithmetical ‘Decidability and Calculability’ described by Philosopher Wilfried 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).

§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 AllAboutAI/CAP World Congress 2012-Alan Turing 2012, held from July 6 to July 12 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 $\overline{\text{PA}}_{\text{standard}}$ of the first order Peano Arithmetic PA is finitarily sound if, and only if, Aristotle’s particularisation holds over $\mathbb{N}$, and the latter is the case if, and only if, PA is $\omega$-consistent.

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

(iii) The axioms of PA are always true under the finitary interpretation $\overline{\text{PA}}_{\text{Algo;thmic}}$ and the rules of inference of PA preserve the properties of satisfaction/truth under $\overline{\text{PA}}_{\text{Algo;thmic}}$.

We concluded that:

Theorem 1: The interpretation $\overline{\text{PA}}_{\text{Algo;thmic}}$ 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 $\overline{[F]}$ 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 ‘algorithmically’ complete in the sense that:

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

Proof: We have by definition that $\overline{[\forall y][F(x)]]}$ interprets as true under the interpretation $\overline{\text{PA}}_{\text{Algo;thmic}}$, if, and only if, $[F(x)]$ is algorithmically computable as always true in $\mathbb{N}$.

Since $\overline{\text{PA}}_{\text{Algo;thmic}}$ is finitarily sound, it defines a finitary model of PA over $\mathbb{N}$—say $M_{\text{PA,thmic}}$—such that:

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

If $\overline{[\forall y][F(x)]]}$ is PA-provable, then it is not the case that $[F(x)]$ is algorithmically computable as always true in $\mathbb{N}$.

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

(i) There is a finitary model—say $M_{\text{PA,thmic}}$—of PA in which $[F(x)]$ is algorithmically computable as always true in $\mathbb{N}$.
In such case, if sound implies that Aristotle’s particularisation holds over the natural numbers under any finitarily reasonable f. bounded arithmetic that, from a proof of no PA numeral \[ n \] is PA-provable for any given PA numeral \[ n \]. Hence, for any given numeral \[ n \] the PA formula \[ xB(\{\bar{n}\}) \] must hold for some \( x \). The lemma follows. □

\section{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 \( \mathcal{I}_{PA(N, \text{Standard})} \) over \( N \), then there is a PA formula \( \phi \) which is algorithmically verifiable as always true over \( N \) under \( \mathcal{I}_{PA(N, \text{Standard})} \), even though \( \phi \) 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 \( \overline{R}(x) \)—such that \( \overline{R}(x) \) is not PA-provable, but \( \overline{R}(n) \) is instantiationally PA-provable for any given PA numeral \( n \). Hence, for any given numeral \( n \), the PA formula \( xB(\{\bar{n}\}) \) must hold for some \( x \). The lemma follows. □

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 \( \overline{R}(x) \) interprets as an algorithmically verifiable, but not algorithmically computable, tautology over \( N \).

**Proof** Gödel has shown that \( \overline{R}(x) \) interprets as an algorithmically verifiable tautology over \( N \). By Corollary 2 \( \overline{R}(x) \) is not algorithmically computable as always true in \( N \). □

**Corollary 4:** PA is not \( \omega \)-consistent. [9]

**Proof** Gödel has shown that if PA is consistent, then \( \overline{R}(n) \) is PA-provable for any given PA numeral \( n \). By Corollary 2 and the definition of \( \omega \)-consistency, if PA is consistent then it is not \( \omega \)-consistent. □

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

**Proof** 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 [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 Aristotelian logic.

\section{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 \( \Sigma^0_1 \) formula \( \phi(\forall x)f(x) \) is provable in the Bounded Arithmetic if, and only if, there is an algorithm that, for any given numeral \( n \), decides the \( \Delta^0_{1+n} \) formula \( f(n) \) as ‘true’:

\[ \text{If } (\forall x)(\exists y)f(x, y) \text{ is provable, then there should be an algorithm to finitely as a function of } x. \]

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

\section{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 \( f(n, m) \) 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 \( (\exists y)f(n, y) \), 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]:

\[ \text{There is an algorithm that decides } f(n) \text{ as 'true' for any given numeral } n. \]

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 \(-f(n, m)\) 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-unprovable, 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 \( \mathcal{N} \), 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 \( \mathcal{G}_k \), which is, essentially, PA with the following 'weakened' Induction Axiom Schema, \( \text{PIND}[k] \):

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

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

\[
\left\{ f(0) \land \forall x (f(x) \rightarrow (f(2 \times x) \land f(2 \times x + 1))) \right\} \rightarrow \forall x f(x)
\]

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

\[
\left\{ f(0) \land \forall x (f(x) \rightarrow (f(k \times x) \land f(k \times x + 1) \land \ldots \land f(k \times x + k - 1))) \right\} \rightarrow \forall x f(x)
\]

Now we have the PA theorem:

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

It follows that the following is also a PA theorem:

\[
\left\{ f(0) \land (\forall x (f(x) \rightarrow f(x+1))) \rightarrow \{ f(0) \land (\forall x (f(x) \rightarrow (f(k \times x) \land f(k \times x + 1) \land \ldots \land f(k \times x + k - 1)))) \right\}
\]

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


2004, Amsterdam.


Notes

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

Return to 4: Go31, p.26(2): \( (\forall n) \neg (\exists \epsilon \leq n \exists B_k (1 \Gamma_k \epsilon \cap r) \) 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 \( \omega \)-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 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 \( \mathcal{T}_{PA[N \text{ Standard}]} \).

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.xxvii; Rg67, 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 \( \omega \)-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 \( n \) denotes the largest natural number lower bound of the rational \( \frac{x}{2} \)

Author’s working archives & abstracts of investigations
2 comments

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

LECENT COMMENTS

Abel Peralta on Can Gödel be held guilty of im...

Bhupinder Singh Anand on Can Gödel be held guilty of im...

Abel Peralta on Can Gödel be held guilty of im...

Bhupinder Singh Anand 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


Privacy & Cookies: This site uses cookies. By continuing to use this website, you agree to their use.

To find out more, including how to control cookies, see here: Cookie Policy

Close and accept