DOI: 10.1017/psa.2026.10273
This is a manuscript accepted for publication in *Philosophy of Science*.
This version may be subject to change during the production process.

Computable Bayesian Epistemology

Josiah Lopez-Wild

Department of Philosophy, University of Warwick, United Kingdom josiah.lopezwild@gmail.com

Abstract

Bayesian epistemology is broadly concerned with providing norms for rational belief and learning using the mathematics of probability theory. Many authors have worried that the theory is too idealized to accurately describe real agents. In this paper I argue that an emerging program, *computable Bayesian epistemology*, can describe more realistic agents while retaining sufficient generality. I situate this program by placing it among the ongoing debate about ideal versus bounded rationality. I then present the basics of computable analysis and demonstrate its usefulness by proving a simple result: there are no computable finitely additive probability measures.

1. Introduction

Bayesian epistemology is broadly concerned with providing norms for rational belief and learning using the mathematics of probability theory. Most (but not all) Bayesians agree on two core norms:

(1) (Probabilism). An agent's beliefs should take the form of a probability function P over some algebra $\mathscr{F}$ of subsets of an outcome space Ω.
(2) (Conditionalization). Upon learning that some subset $A \in \mathscr{F}$ is true, the agent's updated belief in any $B \in \mathscr{F}$ should take the form of the conditional probability $P(B \mid A)$ of B conditional on A.

Many authors have worried, for various reasons, that this picture is too idealized to accurately describe real agents like you and I. Some have pointed out that the standard view seems to imply *logical omniscience*, a requirement that Bayesian agents assign probability 1 to all logical consequences of their knowledge (those propositions to which they already assign probability 1).[1] Others have pointed out that we often struggle to assign perfectly precise probabilities to events (Walley (1991), Seidenfeld et al. (2012*a*)). For example, how likely do you think it is to rain tomorrow? There seems to be no principled reason to assert, e.g., a 60% chance as opposed to a 63.45% chance.

Despite our best intentions we humans often discover that we apparently suffer from various biases and fallacies of belief that, on the face of it, show our beliefs to be probabilistically incoherent—both synchronically and diachronically (see especially

[1] Though see Hacking (1967), Pettigrew (2021) for a compelling response to this worry.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Kahneman et al. (1982), Allais (1953), Ellsberg (1961)). Perhaps it is unreasonable to expect real agents to update by conditionalization. There are in principle infinitely many propositions I could assign some probability to. Am I really expected to update them all exactly whenever I learn something new? What if I am uncertain about the precise content of my new evidence?[2]

These worries stem from a more fundamental concern about boundedness. We know from both experience and theory that humans, nonhuman animals, computer programs, etc. are all bounded in their reasoning capacity. We have good reason to believe that bounded agents cannot adhere to the prescriptions of Bayesianism. One might conclude that, therefore, bounded agents cannot be Bayesian. Yet Bayesianism is on the rise in the behavioral sciences. In cognitive science, for example, a large body of research accounts for these bounds by assuming some underlying computational model, such as a finite-state automaton (Simon (1957), Newell and Simon (1972), Lewis et al. (2014), Lieder and Griffiths (2020)).

It is not obvious how the bounded Bayesian models in the behavioral sciences map onto the ideal Bayesian agent. The former assume strict computational limitations, resulting in behavior that, from an ideal Bayesian perspective, is nowhere near Bayes-optimal. This is a descriptive issue for the philosophical picture. But there is also a normative issue: how can bounded agents like us be expected to adhere to norms that we cannot, as a matter of fact, actually attain? Why do we need ideal Bayesianism?

To resolve these questions we need a principled theory of bounded Bayesian epistemology. There is a young research program in formal epistemology concerning computable agents: Zaffora Blando (2020), Zaffora Blando (2025), Zaffora Blando (2024), Huttegger et al. (2026*a*), Huttegger et al. (2026*b*), Belot (2023*a*), Belot (2023*b*), and Belot (2025), and Lopez-Wild (2025), for example. This literature proposes a natural framework for this theory: computable analysis (Weihrauch (2000)). Using computable analysis we can study *computable* Bayesian agents, presenting a natural starting point for a Bayesian epistemology that is applicable to bounded agents like ourselves.

This program is new and is, of necessity, rather technical. Nonetheless it marks an important development in formal epistemology and deserves to be more widely known. To this end, this paper has three primary aims: to present this program in an accessible way to non-specialists, to defend its philosophical importance, and to prove a theorem that illustrates this importance. In §2 I introduce the debate around ideal and bounded rationality and illustrate it with two recent papers–one by Jennifer Carr (Carr (2022)) and one by David Thorstad (Thorstad (2024)). In §3 I present an outline of the relevant tools from computable analysis. This section defines computability notions for real numbers, real functions, and probability measures in particular. This presentation is far from exhaustive—I have aimed to present only what is necessary and to do so in a conceptually motivated way. In §4 I use this background to give a simple proof of the following: if a probability measure is computable then it is countably additive. In §5 I consider and reject two concerns regarding this result.

[2]See e.g. Jeffrey conditionalization (Jeffrey (1965)).

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

2. Setting the Stage: Ideal vs. Bounded Rationality

On the face of it, the pronouncements of ideal Bayesianism and those of bounded Bayesianism are in tension. Discussing the Bayesian framework he defended in Savage (1954), Savage says that

> . . . the postulates of the theory imply that you should behave in accordance with the logical implication of all that you know (Savage (1967)).[3]

In other words, Savage's theory "implies that if e entails h you should be as confident of h as of e" (Hacking (1967), p. 311). This worry is known in the literature as the *problem of logical omniscience*, and has generated a sizable literature (see Hacking (1967), Garber (1983), Gaifman and Snir (1982) and Seidenfeld et al. (2012*b*), for some examples). The problem, of course, is that it seems entirely unreasonable to expect a bounded agent to be confident in *all* logical consequences of their beliefs; this kind of "logical closure" condition cannot be satisfied by any bounded agent.

The problem of logical omniscience is a particular instance of a general tension between ideal and bounded epistemology. One might respond to the problem by arguing that Bayesianism is unsuitable as a theory of epistemology, and thus we require a new, bounded theory. On the other hand, one might argue that the theory is not *intended* to describe bounded agents, but nonetheless retains its normative status. In this section we will explore this tension by way of two recent papers: one (Carr (2022)), is a defense of ideal epistemology against bounded rationality, while the other (Thorstad (2024)) is a defense of bounded rationality, in part against Carr's arguments. With the details of this debate laid out we will be able to see how computable Bayesianism falls between ideal epistemology and bounded rationality.

Carr's aim is to defend the value of ideal epistemology. She concedes that theories of ideal epistemology have consequences that are not true of real agents. In her words, "Ideal epistemologists are concerned with questions about what perfectly rational, cognitively idealized, computationally unlimited believers would believe" ((Carr, 2022, 1132)). By contrast, "non-ideal epistemologists are concerned with questions about epistemic norms that are satisfiable by most humans much of the time". These glosses certainly make non-ideal epistemology sound more useful. Ideal agents are not like us at all; why study norms that only they can satisfy?

Carr believes that ideal epistemology has virtues that non-ideal epistemology cannot have. She charges non-ideal epistemology with two defects: it is (i) *conventional*, and (ii) *seriously context-sensitive.* By contrast, she argues, ideal epistemology is non-conventional and not seriously context-sensitive. This makes ideal epistemology "normatively robust", which, she argues, is a desirable property of any theory of epistemology.

Let's consider these two defects in turn. First, a theory of epistemology is *conventional* if its standards of evaluation are arrived at via convention. Carr has in mind Lewis' theory of conventions (Lewis (1969)), but we assume other theories of the nature of convention come to similar conclusions. So if a theory of epistemology defines terms like "rational", "justified", etc. in terms of conventions, we say that theory is conventional.

[3]This quote is taken from Seidenfeld et al. (2012*b*), where they note that the published version of Savage's article is misprinted; most of this quote does not appear in that version.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Some non-ideal epistemologists endorse some kind of conventionalism. For a view that could plausibly be called conventional, see Dogramaci (Dogramaci (2015), Dogramaci (2017)).

Non-ideal epistemology, such as bounded rationality, fixes parameters such as processing speed and memory on a case-by-case basis. If we want to study human decision-making then we need information about how much memory the average person has, how people represent the options available to them, how quickly they think, etc. These parameter choices are entirely different from those we would make if we were modeling the visual system or a social network as boundedly rational "agents". In this sense, they are determined *conventionally*.

Second, a theory of epistemology is *seriously context-sensitive* if there is no "normatively privileged resolution of one or more of the context-sensitive parameters" of the theory ((Carr, 2022, 1135)). Carr points out that non-ideal theories tacitly divide cognitive limitations into two camps: those that "lower the bar" for rationality, and those that don't. For example, non-ideal epistemologists usually think that limited information processing speeds, memory, etc. are limitations that are relevant for epistemology. By contrast, Carr argues that "our dispositions toward implicit biases, unreliable heuristics, delusional reasoning, misinterpreting statistical phenomena as having causal implications" are *not* generally considered relevant ((Carr, 2022, 1152)). The question is: why do the former, but not the latter, matter for epistemology? Carr argues that these two groups are divided on a conventional basis; there is no normatively privileged way to draw the line between them.

So, Carr concludes, non-ideal epistemology is conventional and context-sensitive. Ideal epistemology (like Bayesianism) is not. But, she claims, we want a theory that is neither conventional nor context-sensitive, because we want a theory that illuminates *general* principles of epistemology. Compare: a theory of thermodynamics that *only* explains why my cup of coffee cools, but no other instances of heat dissipation, is less satisfying than a general theory that pertains to all instances of energy transfer. A theory of rational belief that applies only to certain people at certain times is similarly less satisfying than one that explains rational belief in any context.

Nonetheless we are stuck with a highly unrealistic theory. Ideal Bayesians, for example, must still hold that (i) Probabilism is rational, (ii) humans cannot satisfy Probabilism, and hence (iii) all humans are irrational. There is no interesting difference between a careful planner and someone who makes life decisions based on tea leaves and horoscopes—both are irrational. Nor is ideal epistemology *ameliorative*: it does not provide suggestions for how to improve our own epistemic well-being.

Thorstad (Thorstad (2024)) defends bounded rationality. In response to the charge of conventionalism, Thorstad points out that bounded rationality *may* but *need not* be conventionalist. An epistemologist who wanted to incorporate bounds into their work could adopt any of the popular normative requirements held by ideal epistemologists. Bounded rationality theorists often assume some computational model whose limitations are intended to capture the agent's bounds (Lieder and Griffiths (2020)). This model is not a convention, except in the weak sense in which the use of any scientific model is an agreed-upon practice by some group of scientists. So conventionalism is not a concern for bounded rationality.

What about context-sensitivity? Thorstad argues that we *can* draw a principled line between those limitations that matter for epistemology and those that don't. To do so he

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

appeals to an agent's *cognitive architecture*, the facts about an agent's cognitive capabilities that do not change over time. As an example he says "it is an architectural fact that our working memory has a fixed capacity, but a non-architectural fact which beliefs are currently held in working memory" ((Thorstad, 2024, 404)). The proposal is that facts about cognitive architecture matter for epistemology, while other facts about an agent's limitations do not.

Moreover, bounded rationality enjoys benefits that ideal epistemology does not. Bounded rationality vindicates the principle that "ought implies can". By making explicit the bounds that agents have, anything deemed "boundedly rational" *can* be done by the agent. Ideal epistemology clearly does not vindicate this principle. As Carr says, "In non-ideal epistemology, 'ought' implies 'can' in some substantive sense... In ideal epistemology, it doesn't" ((Carr, 2022, 1133)). This is an intuitive principle for epistemology—I should not be deemed irrational for failing to have beliefs, update those beliefs, follow a decision rule, etc., when I am incapable *in principle* of doing so.

One consequence that Thorstad explicitly endorses, however, is that the standard of rationality *does* vary from person to person: "what is rational for an agent depends on her abilities and the cost of exercising them" ((Thorstad, 2024, 402)). So it is not seriously context-sensitive *which* cognitive limitations matter, since we have a principled dividing line; but whether a given strategy, decision rule, etc., is rational for someone is context-sensitive, since it depends on their capabilities. This sort of context-sensitivity is exactly the sort that ideal epistemologists want to avoid.

Let's take stock. Carr argues that non-ideal epistemology is conventionalist and seriously context-sensitive, whereas ideal epistemology is not; for this reason, we need ideal epistemology. But, ideal epistemology suffers from its inability to distinguish more or less irrationality, and is not ameliorative. Thorstad argues that non-ideal epistemology is not conventionalist, and that the alleged context-sensitivity can be resolved in a principled way. But, boundedly rational epistemology still suffers from a context-sensitivity that many epistemologists will not like. It does not provide us with a general theory of rationality, but rather a theory of rationality-for-someone-in-some-context.

I suggest that computable Bayesian epistemology enjoys the generality of ideal epistemology while better respecting the intuition that 'ought' implies 'can'. It supplements the classical Bayesian framework with the assumption that the agent's beliefs are represented by a *computable* probability function. Computable Bayesianism abstracts from time and space limitations, memory constraints, information-theoretic bounds, and issues of noise. The theory is therefore not context-sensitive: the underlying computational model never changes. Instead it asks: is it possible, even in principle, for a computable agent to satisfy Bayesian norms? And, if not, are there weaker alternatives that they can satisfy? In this way we can develop a version of Bayesian epistemology that accounts for boundedness but does not, as Savage worried, "entail paradox" (Savage (1967)).

Moreover, because it is not context-sensitive the results are quite general. They apply to any agent whose cognitive powers are no stronger than a universal Turing machine. So if we can prove that some component of Bayesian epistemology (e.g. Probabilism or Conditionalization) is not computable in general, then no bounded agent can satisfy that component. On these grounds I argue that it cannot be an epistemic norm for bounded

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

agents. In this way we recover a version of 'ought' implies 'can': if no computable agent can do it, then they ought not.[4]

One might wonder why we should care about computability at all. Why should epistemology be concerned with the computable? Is this limitation necessary, and is it a reasonable formalization of "boundedness"? First, note that "computable function" is a generous notion. A function could require millions of years to finish its computation and would still be considered computable, despite being of no practical use to us. In truth, all actual applications of Bayesian methods (e.g., Bayesian statistics) use probability measures that are *feasibly* computable, in the sense of computational complexity theory. Moreover, research in bounded rationality tends to assume that agents are much more bounded than a general Turing machine, often opting for finite-state or push-down automata, for example (Lewis et al. (2014), Lieder and Griffiths (2020)). This literature makes these assumptions to reflect the fact that bounded agents have limited time, memory, and processing speed available for their reasoning tasks. By contrast, the general Turing machines at the foundation of computable Bayesianism have unbounded time and memory. From this perspective, mere computability is not a limitation at all, but rather is too *permissive* of an assumption. It would therefore be quite natural to study Bayesianism through the lens of computational complexity theory.[5] While this would be quite interesting, computational complexity theory as a whole is not as well understood as computability theory, and so the corresponding Bayesian theory would be quite incomplete, and fail to derive general epistemic principles. So, for now, the literature has settled on computability as a fruitful, motivated, and unified middle ground.[6]

To further the case for computable Bayesianism we need to state the theory precisely. This requires mathematics that goes beyond standard probability theory. The theory uses tools from computable analysis, a branch of mathematics that studies the computational content of results from classical real analysis—in particular, measure and probability theory. I will introduce this extra mathematics in the next section.

3. Computable Analysis from First Principles

It is important to screen off confusions before entering into the technical details. Computable analysis is *not* a replacement for classical analysis. We are not in the business of defining new and different numbers, functions, etc. Instead we are discovering a finer-grained classification of classical structures. We will work with the exact same set of real numbers as we always have, but now we can show that some of them have a special property: there exist algorithms that exactly compute them.[7] Similarly we are not changing the notion of "function", but rather showing that some functions can be implemented algorithmically—there is a uniform procedure for transforming arguments into values.

[4] Where this should be read "$\neg$ Ought(φ)", rather than "Ought($\neg\varphi$)".

[5] A natural starting point would be Ko's work on the complexity theory of real analysis (Ko (2012)).

[6] Much like computable Bayesianism, formal learning theory, especially Kelly's (Kelly (1996)), often makes explicit computability assumptions. Both Kelly's project and computable Bayesianism make use of the mathematics of computability theory and its close connection to topology (this connection runs through computable analysis and descriptive set theory; for the latter, see Moschovakis (2025) or Mansfield and Weitkamp (1985)). Unlike Kelly's program, we define probability functions on hypothesis spaces, whereas Kelly makes use of learning rules.

[7] In the limit of infinite time; more on this later.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Since our focus is explicitly Bayesian this means in particular that we are not changing probability theory. Rather, the goal is to define and study the well-behaved portions of probability theory. As it turns out, most probability measures, random variables, operations, etc. that formal epistemologists and statisticians use in practice are computable. So this is not a radical proposal that would alter practice.

To start we consider standard computability theory. Computability theory is generally defined on the natural numbers $\mathbb{N}$. All naturals n are computable in the sense that there is a Turing machine which, for example, computes the binary code for n. A function $f : \mathbb{N} \to \mathbb{N}$ is computable if there is a Turing machine which, given a code for n, takes finitely many steps to output a code for $f(n)$. A set $A \subseteq \mathbb{N}$ is computable if there is a Turing machine which, given $n \in \mathbb{N}$ as input, outputs either a code for "Yes" if $n \in A$ or a code for "No" otherwise. See (Soare (2016)) or any other introductory textbook on computability theory for further details.

Importantly, in computability theory we generally assume that the Turing machine operates on codes from a fixed finite alphabet which *represent* natural numbers. For example most textbooks use the alphabet $\{0, 1\}$ of binary codes. The set of all possible finite binary codes, usually denoted $2^{<\omega}$, is countably infinite. Indeed if Σ is a finite set then $\Sigma^{<\omega}$ is countable. So we can easily extend computability notions to any countably infinite set X of objects, not just $\mathbb{N}$. To do so we fix some finite alphabet Σ and define a surjective function $\delta : \Sigma^{<\omega} \to X$. Then a string $p \in \Sigma^{<\omega}$ is a *code* (or *name*) for some element $x \in X$ just in case $\delta(p) = x$. We want every element of X to have a name, so we require δ to be surjective. Note that any given $x \in X$ may have multiple names. As an example of this process we can define finite codes for all rational numbers via the map $\langle i, j, k\rangle \mapsto (i - j)/(k + 1)$. This is a computable surjection from $\mathbb{N}^3$ onto $\mathbb{Q}$. Then we can define computable rational numbers as those with computable codes (in this case, all rationals are computable) and computable functions between rationals are defined as computable functions between their codes.

Transferring computability notions to rational numbers is easy because rationals can be given finite codes. By contrast real numbers are not always finitely representable. Consider the decimal representation of $\pi = 3.1415926\ldots$ We know that this representation never terminates, so any finite initial segment is *not* equal to π. Nonetheless π should be considered a computable real (and *is* computable according to the standard definition below) since we have algorithms for computing increasingly precise representations of it (Brent (2020)). We can prove that in the limit these algorithms compute π exactly. So, intuitively, π should count as a computable real number. But clearly we cannot expect a computable real number to be one whose entire representation is computable in a finite amount of time.

Instead, a computable real number is a number which can be *computably approximated arbitrarily well.* Consider π. Suppose we have run the algorithm for a finite amount of time and it has output 3.1415. If we run the algorithm for longer then it will output more digits to append to this number. But, knowing how decimal representation works, we know that π, whatever it really is, falls somewhere in the interval $[3.1415, 3.1416)$. We have already produced an approximation to π that is within an error bound of 1.0×10^{-4}. And when the next digit (in this case a '9') is printed, the error bound tightens to 1.0×10^{-5}. So we have an algorithm that computes approximations to π with a *computable* error bound : having printed n digits, we know the error bound is $1.0 \times 10^{-(n-1)}$.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

More generally we define a real number r to be computable if there is a uniformly computable sequence $(q_n)_{n\in\mathbb{N}}$ of rational numbers such that $|r - q_n| \leq 2^{-n}$ for all n.[8] The sequence is "uniformly" computable in the sense that there is a single algorithm which, on input n, outputs q_n. Thus a Turing machine can compute an approximation to r to as accurate a degree as desired, and that computation will terminate. Computing r *exactly*, though, need not be a finite process. Commonly used reals like π, e, or any rational are all computable reals. Note, though, that most reals are not computable. To see this, note that each computable real has at least one corresponding Turing machine program that computes approximations to it. There are only countably many Turing machines, so only countably many computable real numbers.

Real numbers are not the only structure in probability theory, though. We need a more general way to define computability notions for things like probability measures and random variables. We'll use the commonly accepted foundation for computable analysis, the Type-Two Theory of Effectivity (TTE); see Weihrauch (2000), Brattka et al. (2008), Braverman and Cook (2006) for introductions.

The trick to computing real numbers was to compute approximations—specifically those which converge to the object at a computable rate. And the way we did this was to define a Turing machine that printed longer and longer names of those approximations. Unlike standard computability theory, though, it's perfectly fine if that computation runs forever. We just need the output to encode good approximations. In the limit the machine produces an infinite code which names the desired real number.

So if we wanted a Turing machine which could compute, for example, functions from real numbers to real numbers, we would also need it to take those infinite codes as inputs ("oracles", in computability theory). More precisely, we define a *type-2 Turing machine* as a Turing machine with:

(1) finitely many one-way infinite read-only input tapes;
(2) finitely many two-way infinite read-write work tapes;
(3) a one-way infinite output tape.

The input and output tapes are "one-way infinite" in the sense that they have a left end, or first cell, and then extend infinitely far to the right. At the start of computation there is a machine head placed at the left end of the output tape and each input tape. Over the course of the computation the machine may move each input head or the output head to the right, but *not* left; backtracking is not allowed. The machine may read the content of the input tapes but cannot change the contents; similarly the machine may write on the output tape but cannot change what it has written.

To get a feel for how this works, let's imagine a type-2 machine that implements a computable real function $f : \mathbb{R} \to \mathbb{R}$, and on the input tape we write the code for a real r such that $f(r) = \pi$. Since π has an infinite code the machine will clearly never halt. Instead we require that over time it writes a sequence that *in the limit* encodes π. Since a code for r is written on the input, the machine may query r for information at any given step. However, the machine clearly does not have enough time to survey *all* of r; it can only read one digit at a time. So, naturally, at any finite time it will only have read

[8] The rate of convergence 2^{-n} can be replaced by any other computable rate, and defines precisely the same set of computable real numbers.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

a finite initial segment of the code for r. Moreover after any finite amount of time the machine will have written at most a finite initial segment of the code for π. These finite initial segments encode the *rational approximations* that we mentioned earlier when discussing computable reals. Thus with more time the machine will output longer codes that encode better rational approximations to $f(r) = \pi$.

The one-way output requirement is necessary for computations on infinite sequences to be well-defined because we only ever witness finite approximations to the final computation. If we allowed the machine to rewrite its output then we could never know *any* information about the output with certainty. For suppose that after some time t the machine has output some finite sequence s; for all we know, at some future time $t + n$ the machine will erase s and write instead some other sequence t. Thus at any time we would never know if the current output contains any information about the limiting result of the computation, rendering the computation useless. This is evidently not the case for the algorithms for computing π; we have proofs that their finite approximations are correct. Thus we say that a type-2 machine computes a function $f : X \to Y$ if, given a (finite or infinite) code for a point $x \in X$, it either (i) runs for a finite amount of time and outputs a finite code for a point $f(x) \in Y$, or (ii) runs for an infinite amount of time and outputs finite codes which converge to an infinite code for a point $f(x) \in Y$.

So we have a Turing machine model designed for computation on infinite data structures; now we need the data. TTE allows us to define computability notions on richer mathematical structures by "encoding" those structures as sequences of symbols, much like a computer would. We do so via a *representation*, a surjective function from either $\Sigma^{<\omega}$ or Σ^{ω} onto the structure we wish to encode. For example, let $\nu_{\mathbb{N}} : 2^{<\omega} \to \mathbb{N}$ be the usual binary encoding of natural numbers; e.g., $\nu_{\mathbb{N}}(100) = 4$. This is a computable representation of $\mathbb{N}$. We let $\langle \cdot, \cdot \rangle$ denote some computable bijective tupling function, with obvious n-ary extension $\langle x_0, x_1, \ldots, x_{n-1} \rangle := \langle \langle x_0, x_1, \ldots \rangle, x_{n-1}, \rangle$. We then define codes of tuples via $\nu_{\mathbb{N}^n} : 2^{<\omega} \to \mathbb{N}^n$, defined

$$\nu_{\mathbb{N}^n}(w) := (x_0, \ldots, x_{n-1}) \iff w = \langle x_0, \ldots, x_{n-1} \rangle.$$

Similarly our earlier computable surjection between $\mathbb{N}$ and $\mathbb{Q}$, the map $\nu_{\mathbb{Q}}(\langle i, j, k \rangle) := (i - j)/(k + 1)$, is a representation.

We can define representations on $\mathbb{R}$ that make precise our earlier discussion of rational approximations. For our purposes it is efficient to assume that Σ at least contains $0, 1$, and a symbol # that has been set aside as a "blank" or "dummy" symbol. We can define a representation $\rho : \Sigma^{\omega} \to \mathbb{R}$ for the real numbers as follows:

$$\rho(w_0 \# w_1 \# \ldots) := r \iff |r - \nu_{\mathbb{Q}}(w_i)| < 2^{-i}$$

for all $i \in \mathbb{N}$. If $\sigma \in 2^{\omega}$ is such that $\rho(\sigma) = r$ for some real $r \in \mathbb{R}$, then r is computable in the sense defined earlier iff σ is.

Now we can define a very general notion of a computable function. Let X and Y be sets with representations $\nu_X : \Sigma^{\omega} \to X$ and $\nu_Y : \Sigma^{\omega} \to Y$. Then a function $f : X \to Y$ is (ν_X, ν_Y)-computable iff there is a computable function $F : \Sigma^{\omega} \to \Sigma^{\omega}$ such that for all

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

$\sigma \in \mathrm{dom}(F)$, $f(\nu_X(\sigma)) = \nu_Y(F(\sigma))$, that is, the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{\quad f \quad} & Y \\ {\scriptstyle \nu_X}\uparrow & & \uparrow{\scriptstyle \nu_Y} \\ \mathrm{dom}(F) & \xrightarrow[\quad F \quad]{} & \Sigma^\omega \end{array}$$

We call F a *realizer* of f. In this precise sense we can vindicate the earlier intuition that f is computable if there is a computable map from codes of arguments of f to codes of values of f. Notice that the computability of f is relativized to the representations on which F operates. We leave it to the reader to determine that when $X, Y = \mathbb{R}$ and $\nu_X, \nu_Y = \rho$ one can derive the definition of computable real function defined earlier in the paper.

3.1. Admissible Representations

So we have a precise definition of computable functions between represented sets (X, ν_X) and (Y, ν_Y). But this notion of computability depends on the representations we choose; as it turns out, different representations make different elements $x \in X$ or functions $f : X \to Y$ computable. We commonly say that different representations *induce different computability notions.* This fact might make computable analysis seem too relativistic to be interesting—one might suspect that objects, functions, etc. are not computable *simpliciter*, but only with respect to some representations. To fix this relativity we define a class of "good" representations that all induce the same computability notion. We call these representations *admissible*.

The central idea behind admissible representations is this: computability is a feature of topology. More precisely: topology is the study of approximations in a space. Briefly, a *topological space* is a set X together with a collection τ of subsets of X, called *open sets*, that satisfy: (i) $\emptyset \in \tau$ and $X \in \tau$, (ii) τ is closed under arbitrary unions, and (iii) τ is closed under finite intersections. For admissible representations we also assume that the topological space is T_0 and second countable. A topological space is T_0 if for any distinct points $x, y \in X$, there is an open set U that contains x but does not contain y.[9] A topological space is second-countable if it has a countable basis.

So suppose we have a set X with a topology (family of open sets) τ, and suppose (X, τ) is T_0 and second-countable. For concreteness suppose $\mathscr{B} = \{B_n \mid n \in \omega\}$ is the countable basis of (X, τ). Since $\mathscr{B}$ is countable, we can identify each B_n with the natural number n, and thereby assign $\mathscr{B}$ a representation $\nu_{\mathbb{N}} : \Sigma^{<\omega} \to \mathbb{N}$. Thus if $\beta : \mathbb{N} \to \mathscr{B}$ is defined $n \mapsto B_n$, then $\alpha = \beta \circ \nu_{\mathbb{N}} : \Sigma^{<\omega} \to \mathscr{B}$ is a representation of the countable basis $\mathscr{B}$. Weihrauch (2000) calls the triple (X, τ, α) an *effective topological space.*

In any T_0, second-countable topological space (X, τ), each point $x \in X$ is uniquely determined by the set $\{B_n \mid x \in B_n\}$ of basis elements that contain x. And we already have a representation α for those basis elements. So we could define a representation $\delta_X : \Sigma^\omega \to X$ of the whole space X by letting the code for a point $x \in X$ be a list of all basis elements B_n that contain x. There are at most countably many such B_n, and each B_n is given a finite code by α. So we could simply concatenate all those codes (placing the dummy symbol # between them) to define a sequence $\sigma \in \Sigma^\omega$ that encodes the point

[9] By symmetry this means that there is also an open set V that contains y but does not contain x.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

$x \in X$. More precisely,

$$\delta_X(\sigma) = x :\iff \{B_n \in \mathscr{B} \mid x \in B_n\} = \{\alpha(s) \mid s \sqsubset \sigma\}$$

where "$s \sqsubset \sigma$" means that s is a finite subword of σ, i.e., s appears (contiguously) somewhere in the code σ. Weihrauch (2000), p. 67, calls this a *standard representation.* Notice that if σ is computable then there is some algorithm that lists the basis elements containing $\delta_X(\sigma)$; one can show that this representation, if defined for $\mathbb{R}$, defines exactly the same set of computable real numbers as our earlier definition.

The standard representation of an effective topological space has a number of important topological properties (Weihrauch (2000), p. 67) that make it the "gold standard" for representations. So whenever possible we want to use a standard representation, or any representation that induces the same computability notion. We call any such representation *admissible.* Speaking precisely, we say that a representation $\delta : \Sigma^\omega \to X$ is admissible if there is a computable function $f : \Sigma^\omega \to \Sigma^\omega$ and a computable function $g : \Sigma^\omega \to \Sigma^\omega$ such that $\delta(\sigma) = \delta_X(f(\sigma))$ and $\delta(g(\pi)) = \delta_X(\pi)$ for any $\sigma \in \mathrm{dom}(\delta)$ and $\pi \in \mathrm{dom}(\delta_X)$. Here f is a translation from δ to δ_X, while g is a translation in the other direction.

This is all the technical background we need to talk about computable probability theory. The highlights are: we can define computability on infinite structures via codes which give increasingly accurate approximations of objects. We pick out admissible representations by making sure those representations interact properly with the topology of the represented space. The standard representation of an effective topological space is admissible, and any representation which is equivalent to it is equally good, because it defines the same computability notions. With these tools in hand, let's discuss probability theory.

3.2. Computable Probability Theory

How should we define a computable probability measure? To be precise, suppose we have a set Ω of possible outcomes and suppose P is a probability measure on Ω. We'll give $\mathbb{R}$ its standard representation ρ. As a warm-up suppose Ω is finite. Then P is a function from the powerset $\mathscr{P}(\Omega)$ of Ω to the unit interval $[0, 1]$. To talk about the computability of P we need a representation for $\mathscr{P}(\Omega)$. $\mathscr{P}(\Omega)$ is a Boolean algebra. The most commonly used topology for Boolean algebras is the *order topology*: the topology generated by the subbasis of open rays

$$(A, \to) = \{B \in \mathscr{P}(\Omega) \mid A \subset B\} \quad \text{and} \quad (\leftarrow, A) = \{B \in \mathscr{P}(\Omega) \mid B \subset A\}$$

for all $A \in \mathscr{P}(\Omega)$. We let τ_o denote the order topology on a given algebra. Be careful about what this means: the points of the topological space $(\mathscr{P}(\Omega), \tau_o)$ are subsets $A \subseteq \Omega$, and the open sets $U \in \tau_o$ are sets of such subsets.[10]

Since Ω is finite, $\mathscr{P}(\Omega)$ is finite. So we can give each point $A \in \mathscr{P}(\Omega)$ a finite code, and any surjective map $\delta_{\text{fin}} : \Sigma^{<\omega} \to \mathscr{P}(\Omega)$ is admissible. In particular, since every code is finite, every point $A \in \mathscr{P}(\Omega)$ is assigned a computable code. As we saw before, to say that P is computable is to say that there is a computable realizer $F_P : \Sigma^{<\omega} \to \Sigma^\omega$ such that $\rho(F_P(\sigma)) = P(\delta_{\text{fin}}(\sigma))$. The result is that if Ω is a finite set then P is computable if

[10] Since $\mathscr{P}(\Omega)$ is finite, the order topology coincides with the discrete topology generated by the family of singletons. Thus under either topology *every* subset of $\mathscr{P}(\Omega)$ is open.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

and only if $P(A)$ is a computable real number for all $A \subseteq \Omega$. And this is exactly what one naïvely expects a computable probability measure to be: an algorithm that computes a probability for any given set.

Suppose instead that Ω is a countably infinite set. Again we use the order topology on $\mathscr{P}(\Omega)$. Since $\mathscr{P}(\Omega)$ is uncountably infinite we cannot use finite codes for our representation. However we can show that $(\mathscr{P}(\Omega), \tau_o)$ is an effective topological space. To see this, enumerate the points $x_n \in \Omega$. Note that $\mathscr{P}(\Omega)$ is a σ-algebra generated by the countable family of singletons $\{x_n\}$. First we need to show that $(\mathscr{P}(\Omega), \tau_o)$ has a countable basis. It suffices to show that there is some countable family of rays

$$(B_n, \to), (\leftarrow, B_n)$$

that generates every open ray $(A, \to)$ and $(\leftarrow, A)$ for each $A \in \mathscr{P}(\Omega)$. This is true: let the B_n be either finite or cofinite subsets of Ω (of which there are countably many). Then if A is either finite or cofinite the result is immediate, whereas when A is both infinite and coinfinite there must exist index sets $J, K \subseteq \omega$ such that

$$(A, \to) = \bigcup_{j \in J} \{(B_j, \to) \mid A \subseteq B_j\}$$
$$(\leftarrow, A) = \bigcup_{k \in K} \{(\leftarrow, B_k) \mid B_k \subseteq A\}.$$

So $(\mathscr{P}(\Omega), \tau_o)$ has a countable basis. Moreover it is T_0: pick any distinct $A, B \in \mathscr{P}(\Omega)$. There are two cases: either $A \not\subseteq B$ or $B \not\subseteq A$. Without loss of generality assume the former case. Then $(A, \to)$ is an open set containing A but not B, and $(\leftarrow, B)$ is an open set containing B but not A.

Therefore $\mathscr{P}(\Omega)$ with the order topology has a standard representation $\delta_\Omega : \Sigma^\omega \to \Omega$. In this case a code for a set $A \subseteq \Omega$ is a list of the open rays $(B_n, \to), (\leftarrow, B_n)$ that contain A, where the B_n are either finite or cofinite. If we identify a subset A with the set of index numbers of elements of A, one can show that the subsets assigned computable codes are precisely the computable subsets of $\mathbb{N}$ (in the standard sense of computability theory). These subsets are mapped to computable real numbers by any computable realizer F_P of P. Therefore a probability measure on a countable set Ω is computable just in case it assigns computable real numbers to all computable subsets.

Moving to uncountable spaces Ω is largely the same as the countable case. The primary difference is we need to determine the σ-algebra $\mathscr{F}$ of subsets of Ω on which the measure P is defined—for example, if Ω is a topological space, we often let $\mathscr{F}$ be the Borel σ-algebra. But once we do that we can run the same line of argument to show that $(\mathscr{F}, \tau_o)$ is an effective topological space, *assuming* (and this is important!) that $\mathscr{F}$ is countably generated. If $\mathscr{F}$ is not countably generated then $(\mathscr{F}, \tau_o)$ is not second-countable. There are still ways to define representations in this case, but it becomes quite messy and is well beyond what we need for this discussion. So in what follows we'll assume that $\mathscr{F}$ is countably generated, which is true for almost all spaces that probabilists and statisticians usually work in. Moreover there are other admissible representations for the Borel σ-algebra that naturally track the construction process of the set (Weihrauch and Tavana (2014)), analogous to the coding schemes used in effective descriptive set theory (Mansfield and Weitkamp (1985)).

We can of course define computability notions for many other parts of probability theory: the integral, random variables, L^p spaces, etc. But for our purposes we can be

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

content with probability measures. The interested reader can find more in, for example, Ackerman et al. (2019) and Hoyrup and Rute (2021).

Having gotten this far one might begin to wonder whether there was some other way to define computability notions for the structures we have considered. Historically there have been multiple different schools of computable analysis, and they all worked with slightly different definitions of central concepts. But, interestingly, they were all shown to be either (i) equivalent, or (ii) inadequate for their intended target. Rute (2020) is a nice overview of this history that discusses the different approaches in mathematical detail.

4. Coherence: Finite versus Countable Additivity

Let's return to Bayesianism. Already we will encounter novel consequences: in this section I prove that there are no computable probability measures that are merely finitely additive.

A probability function is always required to satisfy an additivity condition: if A, B are disjoint sets, then P satisfies

$$P(A \cup B) = P(A) + P(B). \tag{1}$$

But modern measure-theoretic probability also assumes that probability measures are countably additive, satisfying the stronger condition

$$P\left(\bigcup_n A_n\right) = \sum_n P(A_n) \tag{2}$$

where $\{A_n\}_{n\in\mathbb{N}}$ is a countably infinite sequence of disjoint measurable sets. We call a probability function P that satisfies (2) a "countably additive probability measure"; by contrast we call a probability function P that satisfies (1) but not (2) a "(merely) finitely additive probability measure".

Why accept countable additivity? Kolmogorov (Kolmogorov (1950)) originally introduced countable additivity axiomatically as a mathematical expedient. Many long-run convergence theorems, such as the Central Limit Theorem or martingale convergence theorems, rely on countable additivity (Diaconis and Freedman (1986)). These theorems form the foundation for results such as Bayesian convergence to the truth, the Blackwell-Dubins merging of opinions theorem (Blackwell and Dubins (1962)), and others, all of which have foundational philosophical importance in Bayesian epistemology.[11]

Mathematically, countable additivity is a natural condition that allows us to derive powerful results from probability theory. This is not contentious. Its philosophical significance is more contentious. Bruno de Finetti (de Finetti (1974)) has argued that countable additivity is not a reasonable requirement for an agent's credences. He argues that it should be rationally permissible to define a fair lottery over the natural numbers. Countable additivity does not allow such a probability measure: if P were countably additive and assigned some nonzero probability p to each natural n, we would have $P(\mathbb{N}) = \sum_{n=1}^{\infty} P(\{n\}) = \infty$, a contradiction. But, de Finetti argues, this is unintuitive.

[11] Though see Purves and Sudderth (1976) for some restricted convergence results for merely finitely additive measures.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Why can we not say each number is equally likely? If a fair countable lottery is ruled out by our theory of credences then either our intuitions about credences are wrong, or we should change the mathematical assumptions. de Finetti opts for the latter.

This debate has continued on into the modern literature (Kadane et al. (1999), Williamson (1999), Easwaran (2013)). Without recapitulating these arguments, I want to point out that there are very few instances of mathematically natural finitely additive probability measures. One way to prove the existence of merely finitely additive measures is de Finetti's coherence theorem. Michael Nielsen has recently shown that this theorem is equivalent to the Hahn–Banach theorem, which is, in a precise sense, "nonconstructive" (Nielsen (2020)).[12] The theorem shows that coherent previsions can be extended to a finitely additive probability on the powerset of the outcome set. The theorem is "nonconstructive" in a sense which can be classified with tools from computable analysis. It follows from Nielsen's theorem that any finitely additive probability is not computable, and so, any *computable* probability is countably additive. The proof does not display *why* these probabilities are not computable, however. In this section I will prove this fact directly, by way of type-2 Turing machines. The proof is embarrassingly simple if one has absorbed the main concepts of computable analysis. Its simplicity is its strength: one can see clearly why computability implies countable additivity.

First, it is a fundamental result of computable analysis that all computable functions are continuous (Weihrauch (2000), Theorem 2.2.3). This should not be too surprising, since we've seen that computability is intimately related to topology. More precisely, suppose X has topology τ_X and representation δ_X, while Y has topology τ_Y and representation δ_Y. Recall that we call a function $f : X \to Y$ *computable* if it has a computable realizer; in this case we say that it is (δ_X, δ_Y)-computable. Similarly if f has a continuous realizer we say f is (δ_X, δ_Y)-continuous. But since X and Y are topological spaces it makes sense to ask whether f itself is continuous in the standard sense. Weihrauch proves the following:

Proposition (Weihrauch (2000), Theorem 3.2.11)**.** *For any $f : X \to Y$, f is continuous (in the standard sense) if and only if f is (δ_X, δ_Y)-continuous.*

So being continuous and having a continuous realizer are equivalent properties. This result has the following important corollary:

Proposition (Weihrauch (2000), Corollary 3.2.12)**.** *For any $f : X \to Y$, if f is (δ_X, δ_Y)-computable then it is (δ_X, δ_Y)-continuous, and hence continuous in the standard sense.*

Let's be clear what this proposition says. First, if f has a computable realizer, then that realizer is continuous. Second, since it has a continuous realizer, f is itself continuous as a function between topological spaces. So: every computable function is continuous. It is crucial that we used admissible representations for Proposition 4; it need not hold otherwise.

[12] Similar results are proved in Lauwers (2010) and Lauwers (2017). Nielsen notes that the Hahn-Banach theorem is equivalent to a weak version of the Axiom of Choice.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Returning to our main topic, countable additivity is also a form of continuity, though that is not apparent in the formulation above. Suppose $A_1, A_2, \ldots$ is a sequence of measurable sets such that $A_1 \subseteq A_2 \subseteq \ldots$ and $\bigcup_{i=1}^{n} A_i = A$, and A is itself measurable. We also write $A_n \uparrow A$ to denote this sequence. Then a probability measure P is montonely continuous from below if $P(\bigcup_{i=1}^{n} A_n) \to P(A)$ as $n \to \infty$. Alternatively, suppose $B_1, B_2, \ldots$ is a sequence of measurable sets such that $B_1 \supseteq B_2 \supseteq \ldots$ and $\bigcap_{i=1}^{n} B_i = \emptyset$, written $B_n \downarrow \emptyset$. Then P is monotonely continuous from above (at the empty set) if $P(\bigcap_{i=1}^{n} B_n) \to 0$ as $n \to \infty$.[13] One can show (Ash) that in the presence of the other axioms of probability, monotone continuity from below, monotone continuity from above, and countable additivity are all equivalent (Ash and Doléans-Dade (2000), Theorem 1.2.8). This means that if P is merely finitely additive then there exists some sequence such that $A_n \uparrow A$ but $P(A_n) \not\to P(A)$, and some sequence such that $B_n \downarrow \emptyset$ but $P(B_n) \not\to 0$.

Monotonely continuous probability certainly looks like a continuous function between topological spaces. But we don't normally think of the algebra $\mathscr{F}$ as a topological space, so we don't normally think of probability measures as being literally continuous. But we now know that to define computability notions for probability we need $\mathscr{F}$ to be a topological space; we can equip it with the order topology τ_o. Doing so one can show that if $A_n \uparrow A$ or $B_n \downarrow \emptyset$, then $A_n \to A$ and $B_n \to \emptyset$ in the order topology.[14] Thus if $P : (\mathscr{F}, \tau_o) \to [0, 1]$ is continuous, then it is monotonely continuous, and hence countably additive. Putting these facts together we have the following.

Theorem. *There is no computable merely finitely additive probability measure on a countably generated algebra.*

Proof. Suppose $\mathscr{F}$ is a countably generated algebra (not necessarily a σ-algebra)of subsets of some set Ω. Then $(\mathscr{F}, \tau_o)$ is an effective topological space, and has a standard representation $\delta_{\mathscr{F}}$. Let $P : \mathscr{F} \to [0, 1]$ be a merely finitely additive probability measure. Then P is not monotonely continuous, hence not continuous with respect to τ_o. By Proposition 4, P is not $(\delta_{\mathscr{F}}, \rho)$-computable. □

This result is quite general. We did not require $\mathscr{F}$ to be a σ-algebra, since the order topology is well-defined on any Boolean algebra, nor do we require the outcome space Ω to have any particular structure, such as a topology.

The obvious choice point in this proof is the choice of topology τ_o. Why is this the right topology for the algebra? First, it's the most commonly used topology for ordered sets. Second, the only other commonly studied topology for algebras, the *topology of order convergence*,is a finer topology—that is, it is a superset of τ_o.[15]It is a basic fact of topology that if $f : (X, \tau_1) \to (Y, \tau_Y)$ is continuous and $\tau_1 \subseteq \tau_2$, then

[13]Indeed, Kolmogorov's original axioms for measure-theoretic probability include the Axiom of Continuity (Axiom VI), which states that probability measures are monotonely continuous from above. He subsequently proves the generalized addition equation as a theorem (Kolmogorov (1950), pp. 14–15).

[14]We'll just prove that $A_n \to A$; the other case is similar. Any open ray $(\leftarrow, U)$ containing A clearly contains every A_n. Since $A_n \uparrow A$ we know that for any open ray $(U, \rightarrow)$ containing A there is $k \geq 0$ such that $A_k \in (U, \rightarrow)$, and so for every $m > k, A_m \in (U, \rightarrow)$. These rays form a subbasis for the order topology, and so for any open set containing A there is some index after which every A_n is an element of that open set, so $A_n \to A$.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

$f : (X, \tau_2) \to (Y, \tau_Y)$ is continuous. So the result also holds if we use the topology of order convergence.

Let's finish this section with an example. Consider the de Finetti lottery on the natural numbers, denoted P_D. This finitely additive probability measure is defined on $\mathscr{P}(\mathbb{N})$. While we can't give a precise description of its behavior (because it is nonconstructive), we at least know that it assigns 0 to every finite set of numbers, and 1 to every cofinite set. Such a measure is guaranteed to exist (assuming the Hahn–Banach theorem) thanks to de Finetti's coherence theorem.

By Theorem 4 we know that this measure is not computable. It is instructive to see why. As usual we equip $\mathscr{P}(\mathbb{N})$ with the order topology. The standard representation $\delta_{\mathscr{P}(\mathbb{N})}$ encodes subsets $A \subseteq \mathbb{N}$ by giving a list of open rays $(B_n, \to), (\leftarrow, B_n) \subseteq \mathbb{N}$ that contain A, where the B_n are either finite or cofinite sets. If P were computable then there would be a (type-2) Turing machine which takes these codes as input and output a series of approximations to the assigned probability. At a machine level this means that the Turing machine, after a finite amount of time, outputs some finite string that encodes some first approximation. Crucially, the machine must output *something* after a finite amount of time, during which it could only have read a finite amount of the code for its input x. Weihrauch (Weihrauch (2000)) calls this the "Finiteness Property" for type-2 machines; it has also been called the "Use Principle" of Turing machines (Soare (2016)).

Suppose that the machine receives as input an infinite sequence σ whose digits are members of some finite alphabet. By the Finiteness Property that machine must output some finite code t after having read only a finite prefix s of σ. And, since the machine is deterministic, this means that if the machine is given as input any *other* code τ that also begins with the string s, then the machine will output t after the same number of steps. So similar input codes are mapped to similar output codes—the function is continuous.[16]

So suppose we have a machine T that realizes the de Finetti measure. It receives as input a list of open rays containing the encoded subset. It has to read some finite prefix of this code and then output some finite string. We can show that, since the de Finetti measure is discontinuous, T must output wrong information on arbitrarily many inputs—there is no correct way to define T. For our discontinuity we'll choose the fact that $P(n) = 0$ for all n, but $P(\mathbb{N}) = 1$. Each subset has multiple codes, since any permutation of the family of open rays containing that subset is an admissible code. In particular, suppose we have a sequence σ that encodes the rays $(\{n\}, \to)$ in increasing order. Suppose σ contains codes for each such ray, so that it encodes $\mathbb{N}$.

What does T do with σ? After some finite prefix it must output something. Suppose it has seen m many open rays. So, the information available to the machine thus far is that the encoded set contains the first m natural numbers. Recall that we defined the representation ρ of the real numbers as a sequence of rationals, where the n^{th} rational is within 2^{-n} of the encoded real. The first rational that T outputs must be within $1/2$ of the correct number. The correct number for σ is 1, since it encodes $\mathbb{N}$. So, we could have T simply output 1. But then some other code, σ', which began with the first m

[15]This topology can be given as follows. Let $(A_n)_{n\in\omega}$ be a sequence of elements of $\mathscr{F}$. Then $(A_n) \to A$ if and only if there exist two sequences $(B_n)_{n\in\omega}$ and $(C_n)_{n\in\omega}$ such that (i) if $i \leq j$ then $B_i \subseteq B_j$ and $C_j \subseteq C_i$, (ii) for all i, $B_i \subseteq A_i \subseteq C_i$, and (iii) $\sup B_n = A = \inf C_n$.

[16]The relevant topology on Σ^ω, the space of codes, is the product topology generated by the discrete topology on Σ. This topology has as a basis the family of clopen sets $[s] = \{\sigma \mid \exists n \in \omega : \sigma \restriction n = s\}$.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

rays $(\{m\}, \rightarrow)$ and then continued with the ray $(\leftarrow, \{1, 2, \ldots, m+1\})$ would thereby encode the finite set $\{1, 2, \ldots, m\}$, which is assigned probability 0. So T would output 1 on σ', which is wrong—it isn't within $1/2$ of 0, the correct answer.

Now obviously we could change the rate of convergence for representations of reals. Perhaps instead of 2^{-n} we pick some slower computable bound. The point is that we can repeat this process: we can find some sufficiently long prefix of σ on which T has output sufficient information to be incorrect on some other sequence σ' which agrees with σ to that point. And both $\delta_{\mathscr{P}(\mathbb{N})}$ and ρ are admissible, so any other admissible representations will have the same problem. So a brief excursion into computability theory already reveals interesting consequences for Probabilism: a bounded agent's beliefs *must* be countably additive.

5. A New Defense of An Old Norm

Given the arguments above, Theorem 4 demonstrates that computable Bayesians must have countably additive credences. One might worry that the theorem relies on the fact that type-2 Turing machines seem to "bake in" continuity. While a computable probability measure *does* assign computable reals to computable events, we noted earlier that not all Borel events in an uncountable space are computable, and hence not all events can be assigned a computable real. So, one might argue, it seems that this algorithm that computes probabilities for the computable events "represents" a bounded agents credences on all Borel events simply *because* we have defined type-2 Turing machines in a "continuous" way that allows us to prove Proposition 4. Indeed Huttegger et al. (2026*b*) suggest that these probabilities over arbitrary Borel sets can be taken as an "implicit commitment that the agent holds" (311), and by Carathéodory's extension theorem (see e.g. Ash and Doléans-Dade (2000)), the "explicit" computable probabilities on basic events define a unique probability.

I think this worry misses the mark, and Huttegger, Walsh, and Zaffora Blando need not make a distinction between the "implicit" and "explicit" commitments of an agent in this way. Recall that a probability measure is computable if it has a *computable realizer*; this realizer is a function $F : \Sigma^\omega \rightarrow \Sigma^\omega$ mapping infinite sequences to infinite sequences which defines the machine's behavior. This means that, given some arbitrary Borel event B, the probability $P(B)$ is defined by the behavior of F on some code σ. So it is incorrect to say that we *only* have an algorithm for computing the probabilities of computable events. Rather, we have an algorithm that uses codes of Borel sets as oracles; when these codes are computable, the algorithm prints computable codes of reals. When these codes are not computable, the algorithm prints non-computable codes of reals.[17] So another way to state Theorem 4 is: there is no computable realizer (i.e., type-2 Turing machine) that maps codes for Borel sets to codes for reals in such a way that its induced probability is merely finitely additive. A realizer that did induce a finitely additive probability would require access to a non-computable oracle as side information, and in this sense is not computable.

[17] Indeed one can show that computable realizers preserve the arithmetical complexity of input codes; this property is what characterizes them as computable.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

One might also have thought that since finite additivity is logically weaker than countable additivity it is sometimes more feasible to satisfy the former but not the latter. After all, every countably additive measure is finitely additive, but not vice versa. How, then, do we make sense of our result? I argue that Theorem 4 actually clarifies the debate substantially. The choice is not between satisfying the axiom of countable additivity versus satisfying the axiom of finite additivity. Rather, the choice is between a countably additive measure versus a measure that is merely finitely, but not countably, additive. And, as it turns out, the only instances of the latter kind of probability are non-computable. In fact we can agree with de Finetti that we need not *stipulate as an axiom* that agents must have countably additive credences. Rather, if we agree that bounded agents' credences should be represented by a computable probability measure, then countable additivity is a *theorem*.

6. Conclusion

I have argued that computable analysis provides a natural framework to study Bayesian epistemology for slightly more realistic bounded agents. Despite the fact that the research program is young it has already produced novel results—for example, that there are no computable finitely-additive probability measures, or that agreement on randomness implies merging of opinions (Zaffora Blando (2025)). I have argued that these results matter because normative theories of epistemology should satisfy some version of the principle that "ought" implies "can", which ideal epistemology does not. I concluded that while bounded agents should have countably additive credences, they cannot (if they are coherent) have merely finitely additive ones. This simple result is a contribution to a standing debate that shows that computable Bayesianism holds philosophical promise.

I have also argued that computable Bayesianism avoids that context-sensitivity of more restrictive programs in the tradition of bounded rationality. This is not to say that computable Bayesian epistemology contradicts those theories. On the contrary, this work strengthens results from those fields by showing that some problems are impossible for an agent to solve because of the mathematics of computability theory alone, not because of some contingent fact about their individual limitations.

Of course, much remains to be done. In this paper we have laid out the basics of computable probability theory, but there is a vast literature that I can only gesture at here. This work may well find application in ongoing debates in formal epistemology. There are also many avenues for further research. If conditionalization is sometimes noncomputable, as shown in Ackerman et al. (2019), what is (are) the best alternative update rule(s) for computable agents? How should we measure "better" and "worse" update rules in this case? We also do not have computable versions of Dutch book or expected accuracy theorems. As with finite additivity we may find unexpected barriers to computability, causing us to re-evaluate the normative import of these results. We may find that an epistemology for real agents is surprisingly different from our traditional theories.

Acknowledgments I would like to thank Simon Huttegger, Brian Skyrms, Toby Meadows, Jeff Barrett, Kenny Easwaran, Elijah Spiegel, Antoine Mercier, and two anonymous reviewers for their helpful comments and insightful conversations. My thanks also to participants of the Philosophy of Science: Past, Present, and Future conference held at the University of Minnesota for valuable feedback on the topic.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Funding Statement None to declare.

Declarations None to declare.

References

Ackerman, N. L., Freer, C. E. and Roy, D. M. (2019) On the computability of conditional probability. *Journal of the ACM*. *66*. https://doi.org/10.1145/3321699.

Allais, M. (1953) Le comportement de l'homme rationnel devant le risque: Critiques des postulats et axiomes de l'ecole americaine. *Econometrica*. *21*(4), 503–546. https://doi.org/10.2307/1907921.

Ash, R. and Doléans-Dade, C. (2000) *Probability and Measure Theory*. Academic Press.

Belot, G. (2023a) That does not compute: David Lewis on credence and chance. *Philosophy of Science*. p. 1–10. https://doi.org/10.1017/psa.2023.11.

Belot, G. (2023b) Unprincipled. *The Review of Symbolic Logic*. p. 1–40. https://doi.org/10.1017/S1755020323000151.

Belot, G. (2025) Mechanical turkeys. *Journal of Philosophical Logic*. *54*(1), 197–218. https://doi.org/10.1007/s10992-025-09784-9.

Blackwell, D. and Dubins, L. (1962) Merging of opinions with increasing information. *The Annals of Mathematical Statistics*. *33*, 882–886. https://www.jstor.org/stable/2237864.

Brattka, V., Hertling, P. and Weihrauch, K. (2008) A tutorial on computable analysis . In *New Computational Paradigms*. Springer. https://doi.org/10.1007/978-0-387-68546-5_18.

Braverman, M. and Cook, S. (2006) Computing over the reals: Foundations for scientific computing. *Notices of the AMS*. *53*(3), 318–329.

Brent, R. P. (2020) The Borwein brothers, Pi and the AGM. From Analysis to Visualization. Cham. Springer International Publishing. pp. 323–347. https://doi.org/10.1007/978-3-030-36568-4_21.

Carr, J. R. (2022) Why ideal epistemology? *Mind*. *131*(542), 1131–1162. https://doi.org/10.1093/mind/fzab023.

de Finetti, B. (1974) *Theory of Probability*. vol. 1. Wiley.

Diaconis, P. and Freedman, D. (1986) On the consistency of bayes estimates. *Annals of Statistics*. *14*, 1–67. https://www.jstor.org/stable/2241255.

Dogramaci, S. (2015) Communist conventions for deductive reasoning. *Noûs*. *49*, 776–799. https://doi.org/10.1111/nous.12025.

Dogramaci, S. (2017) Why is a valid inference a good inference? *Philosophy and Phenomenological Research*. *94*, 61–96. https://doi.org/10.1111/phpr.12206.

Easwaran, K. (2013) Why countable additivity? *Thought*. *2*, 53–61. https://doi.org/10.1002/tht3.60.

Ellsberg, D. (1961) Risk, ambiguity, and the Savage axioms. *The Quarterly Journal of Economics*. *75*(4), 643–669. https://doi.org/10.2307/1884324.

Gaifman, H. and Snir, M. (1982) Probabilities over rich languages, testing and randomness. *The Journal of Symbolic Logic*. *47*(3), 495–548. https://doi.org/10.2307/2273587.

Garber, D. (1983) Old evidence and logical omniscience In *Testing Scientific Theories*, Earman, J. (eds). Minnesota University Press.

Hacking, I. (1967) Slightly more realistic personal probability. *Philosophy of Science*. *34*(4), 311–325. https://doi.org/10.1086/288169.

Hoyrup, M. and Rute, J. (2021) Computable measure theory and algorithmic randomness . In *Handbook of Computability and Complexity in Analysis*. Springer. https://doi.org/10.1007/978-3-030-59234-9_7.

Huttegger, S. M., Walsh, S. and Zaffora Blando, F. (2026a) Algorithmic randomness and the weak merging of computable probability measures. *Annals of Pure and Applied Logic*. *177*(8), 103737. https://doi.org/10.1016/j.apal.2026.103737.

Huttegger, S. M., Walsh, S. and Zaffora Blando, F. (2026b) Bayesian convergence for computably bounded agents. *Philosophy and Phenomenological Research*. *112*(1), 307–324. https://doi.org/10.1111/phpr.70079.

Jeffrey, R. (1965) *The Logic of Decision*. McGraw-Hill.

Kadane, J. B., Schervish, M. J. and Seidenfeld, T. (1999) Statistical implications of finitely additive probability . In *Rethinking the Foundations of Statistics*. Cambridge University Press. chapter 2.5, pp. 211–232.

Kahneman, D., Slovic, P. and Tversky, A., editors (1982) *Judgment under Uncertainty: Heuristics and Biases*. Cambridge University Press.
Kelly, K. (1996) *The Logic of Reliable Inquiry*. Oxford University Press.
Ko, K.-I. (2012) *Complexity Theory of Real Functions*. Springer Science & Business Media.
Kolmogorov, A. N. (1950) *Foundations of the Theory of Probability*. Chelsea Publishing Company.
Lauwers, L. (2010) Purely finitely additive measures are non-constructible objects. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.1604383.
Lauwers, L. (2017) Infinite lotteries, large and small sets. *Synthese*. *194*(6), 2203–2209. https://doi.org/10.1007/s11229-016-1052-x.
Lewis, D. (1969) *Convention: A Philosophical Study*. Wiley-Blackwell.
Lewis, R. L., Howes, A. and Singh, S. (2014) Computational rationality: Linking mechanism and behavior through bounded utility maximization. *Topics in Cognitive Science*. *6*, 279–311. https://doi.org/10.1111/tops.12086.
Lieder, F. and Griffiths, T. L. (2020) Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*. *43*, 1–60. https://doi.org/10.1017/S0140525X1900061X.
Lopez-Wild, J. (2025) A computable von Neumann-Morgenstern representation theorem. *Synthese*. *205*(5), 182. https://doi.org/10.1007/s11229-025-04980-1.
Mansfield, R. and Weitkamp, G. (1985) *Recursive Aspects of Descriptive Set Theory*. Oxford University Press.
Moschovakis, Y. N. (2025) *Descriptive set theory*. vol. 155. American Mathematical Society.
Newell, A. and Simon, H. A. (1972) *Human Problem Solving*. Prentice-Hall.
Nielsen, M. (2020) The strength of de Finetti's coherence theorem. *Synthese*. *198*(12), 11713–11724. https://doi.org/10.1007/s11229-020-02825-7.
Pettigrew, R. (2021) Logical ignorance and logical learning. *Synthese*. *198*, 9991–10020. https://doi.org/10.1007/s11229-020-02699-9.
Purves, R. A. and Sudderth, W. D. (1976) Some finitely additive probability. *The Annals of Probability*. *4*(2), 259–276.
Rute, J. (2020) Algorithmic randomness and constructive/computable measure theory In *Algorithmic Randomness: Progress and Prospects*, Franklin, J. N. Y. and Porter, C. P. (eds). Lecture Notes in Logic. Cambridge University Press. pp. 58–114.
Savage, L. J. (1954) *The Foundations of Statistics*. Wiley & Sons.
Savage, L. J. (1967) Difficulties in the theory of personal probability. *Philosophy of Science*. *34*(4), 305–310. https://doi.org/10.1086/288168.
Seidenfeld, T., Schervish, M. J. and Kadane, J. B. (2012a) Forecasting with imprecise probabilities. *International Journal of Approximate Reasoning*. *53*, 1248–1261. https://doi.org/10.1016/j.ijar.2012.06.018.
Seidenfeld, T., Schervish, M. J. and Kadane, J. B. (2012b) What kind of uncertainty is that? using personal probability for expressing one's thinking about logical and mathematical propositions. *The Journal of Philosophy*. *109*(8/9), 516–533.
Simon, H. (1957) *Models of Man*. John Wiley.
Soare, R. I. (2016) *Turing Computability: Theory and Applications*. Springer.
Thorstad, D. (2024) Why bounded rationality (in epistemology)? *Philosophy and Phenomenological Research*. *108*(2), 396–413. https://doi.org/10.1111/phpr.12978.
Walley, P. (1991) *Statistical Reasoning with Imprecise Probabilities*. vol. 42 of *Monographs on Statistics and Applied Probability*. Chapman and Hall.
Weihrauch, K. (2000) *Computable Analysis: An Introduction*. Springer.
Weihrauch, K. and Tavana, N. R. (2014) Representations of measurable sets in computable measure theory. *Logical Methods in Computer Science*. *10*. https://doi/org/10.2168/LMCS-10(3:7)2014.
Williamson, J. (1999) Countable additivity and subjective probability. *The British Journal for the Philosophy of Science*. *50*(3), 401–416. https://doi.org/10.1093/bjps/50.3.401.
Zaffora Blando, F. (2020) *Patterns and Probabilities: A Study in Algorithmic Randomness and Computable Learning*. Ph.D. thesis. Stanford University.
Zaffora Blando, F. (2024) Pride and probability. *Philosophy of Science*. *91*(3), 634–660. https://doi.org/10.1017/psa.2023.177.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press

Zaffora Blando, F. (2025) Bayesian merging of opinions and algorithmic randomness. *The British Journal for the Philosophy of Science*. *76*(4), 921–952. https://doi.org/10.1086/721758.

https://doi.org/10.1017/psa.2026.10273 Published online by Cambridge University Press