TCS 5003
pp: 1–23 (col.fig.: nil)
PROD. TYPE: COM
ED: Sushma
PAGN: Mamatha – SCAN: nil

ARTICLE IN PRESS

ELSEVIER

Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE @ DIRECT®
Theoretical Computer Science ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮

Theoretical Computer Science

www.elsevier.com/locate/tcs

Uncomputability: the problem of induction internalized

Kevin T. Kelly

Department of Philosophy, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Received 15 August 2003; received in revised form 29 October 2003

Abstract

I argue that uncomputable formal problems are intuitively, mathematically, and methodologically analogous to empirical problems in which Hume's problem of induction arises. In particular, I show that a version of Ockham's razor (a preference for simple answers) is advantageous in both domains when infallible inference is infeasible. A familiar response to the empirical problem of induction is to conceive of empirical inquiry as an unending process that converges to the truth without halting or announcing for sure when the truth has been reached. On the strength of the analogies developed, I recommend the adoption of a similar perspective on uncomputable formal problems. One obtains, thereby, a well-defined notion of "hyper-computability" based entirely on classical computational models and on standards of success that have long been regarded as natural in the empirical domain.

Keywords: Hyper-computation; Learning in the limit; Computing in the limit

1. Introduction: relations of ideas, matters of fact

Hume [6] held that all mathematical and logical reasoning is anchored in the mere "relation of ideas" within the mind. According to Hume, ideas are surveyable in their entirety by a mind of sufficient acuity, so that all a priori truths can be reduced to mental inclusions (e.g., "unmarried" is *part* if the concept of "bachelor = unmarried male"). Hume conceived of empirical "matters of fact" quite differently. To be certain of an empirical law, one must already have seen all possible cases covered by the law at an instant, but only finitely many instances are observed by a given stage of

E-mail address: kk3n@andrew.cmu.edu (K.T. Kelly).

0304-3975/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/j.tcs.2003.12.013

inquiry. Thus, Hume arrives at his celebrated problem: if logic does not justify the induction of laws from instances, what does? He answers that inductive generalization is an unjustified habit—something we simply do and get away with *when* we get away with it.

One response to Hume's problem [3,4,8,10,17,19] is to drop the tacit requirement that successful inquiry must halt, ring a bell [9], or otherwise infallibly signal having found the right answer. Then inquiry can be said to succeed in the sense that it stabilizes, eventually, to the right answer, possibly with some surprises and retractions of earlier answers along the way. In this manner, empirical inquiry can be both fallible and truth-directed. One would prefer a scientific method that is guaranteed to signal its arrival at the truth, but there is no such procedure for drawing general conclusions from particular instances. In such cases, a weaker kind of success must be entertained *if* one is to speak of scientific success at all. [1]

Hume seems to assume that formal problems are infallibly solvable and that empirical problems are not, but neither claim is true in general. The empirical question "will it rain tomorrow?" is decidable infallibly by waiting to see [2] and uncomputable formal problems are not infallibly solvable by algorithmic means. Indeed, infallibly solvable empirical problems are quite analogous to computably decidable formal problems and empirical problems that have no infallible solution are strongly analogous to uncomputable formal problems. That is no accident. In light of Turing's philosophical analysis of algorithmic computability in terms of Turing machines [21], algorithmic unsolvability arises out of finiteness and locality conditions of the agent quite analogous to those that give rise to the problem of induction, e.g., the agent (the Turing machine's read–write head) cannot scan or write on infinitely many tape squares in an instant or discriminate letters from an infinite alphabet (because the differences would end up being sub-microscopic), etc.

My aim in this paper is to emphasize some analogies, intuitive, mathematical, and methodological, between formal and empirical reasoning. [3] Intuitively, uncomputable problems like the halting problem seem to demand certainty that something will never happen (e.g., that a computation will never halt) based only on a finite run of experience (the computation has not halted yet). Mathematically, some well-known theorems in the theory of computable functions can be interpreted as providing deep connections between formal and empirical reasoning. Methodologically, I argue that "Ockham's razor", a systematic bias in favor of "simple" answers in the face of uncertainty, can

[1] Alternatively, one can retain the halting condition if one substitutes some notion of "confirmation" or "coherence" for truth as an aim of inquiry. My view, for what it is worth, is that this "ersatz" approach gives up too quickly on truth and underestimates the problem of *computing* the "ersatz" confirmation or coherence relation. As I have discussed this issue at length elsewhere [11,12,15], I will not do so here.

[2] Yes, empirical infallibility in such cases demands the exclusion of such philosophical doubts as being a brain in a vat that is fed neural stimulation giving rise to rain-like sensations. But one must also suspend doubts about properly following an algorithm or about the possible inconsistency of arithmetic in the formal case, so again the two cases are analogous. Infallibility is always relative to some restricted range of possibilities.

[3] I have discussed analogies between computability and empirical reasoning before in several places including [10,11,13–15].

be shown to facilitate convergence to the truth in both the empirical and the formal domains when infallible solutions are infeasible.

On the basis of the preceding analogies, I recommend a convergent, defeasible perspective on uncomputable formal problems as well as on inductive empirical problems. [4] This approach yields a natural concept of "hyper-computation" based entirely on classical computational models. The basic idea is not new. It was pioneered by Putnam [19] and Gold [4] and has since then served as a fertile source of ideas within computational learning theory. [5]

2. The problem of induction

Hume freely admitted his debt to ancient skepticism. In the late middle ages, Buridan recounted the ancient argument for inductive skepticism as follows:

> Let us assume... that from the will of God, whenever you have sensed iron, you have sensed it to be hot. It is sure that... you would judge the iron which you see and which in fact is cold, to be hot, and all iron to be hot. And in that case there would be false judgments, [even though] you have as much experience of iron as you now in fact have of fire. [6]

Quaint as it is, Buridan's argument establishes a negative theorem about what can be learned from empirical evidence and how. A few definitions are required in order to state precisely what the argument shows. The world affords the scientist with an unending *input stream* σ of discrete inputs coded as natural numbers. Let $\mathbf{N}^{\mathbf{N}}$ denote the set of all input streams. An *empirical proposition* says something about what this infinite sequence will be like. For example, "always hot" says that the input stream will be an unending sequence of "hot" observations. Each empirical proposition is identified with the set of input streams of which it is true, and hence is a subset of $\mathbf{N}^{\mathbf{N}}$. An *empirical presupposition* P is an empirical proposition that delimits the range of possible input streams over which one would like to succeed. An *empirical question* is a countable collection Π of mutually exclusive empirical propositions called *potential answers* to the question. If σ is an input stream satisfying an answer to Π, let $\mathrm{ans}_\Pi(\sigma)$ denote the unique answer in Π that contains σ. Answers are infinite sets. To give methods something concrete to output, let numbers be assigned to answers by an injective mapping $\alpha : \Pi \to \mathbf{N}$ called the *answer coding function* for Π. An *empirical problem* is a triple (P, Π, α), where P is an empirical presupposition, Π is an empirical question whose potential answers cover P, and α is an answer coding function for Π. It is assumed that every input stream in the presupposition satisfies some potential answer to the question. In Buridan's problem, the empirical presupposition is that the input stream will consist entirely of observations of "hot" or "non-hot" and the empirical question is whether or not every observation will be "hot" (i.e., whether the input

---

[4] Cf. In this respect, my position agrees with those expressed in the papers by Kugel and Burgin in this volume.

[5] For reviews and bibliographies, cf. [8,10,17].

[6] J. Buridan, Questions on Aristotle's Metaphysics, Book II, Question 1, translation in [7].

stream is the constant "hot" sequence). Code "hot" as 1 and "non-hot" as 0. Then the presupposition of Buridan's problem is the set $2^{\mathbf{N}}$ of all infinite Boolean sequences and the question is the partition $\Pi = \{\{\mathbf{1}\}, 2^{\mathbf{N}} - \{\mathbf{1}\}\}$, where $\mathbf{1}$ is the unit constant function. The question is binary, so let α arbitrarily code $\{\mathbf{1}\}$ as 1 and $2^{\mathbf{N}} - \{\mathbf{1}\}$ as 0. So Buridan's empirical problem can be represented as the triple $(2^{\mathbf{N}}, \Pi, \alpha)$.

An *empirical method* for problem (P, Π, α) responds to each finite, initial sequence of an input stream with some code number in the range of α or with '?', which indicates a refusal to choose an informative answer. Let $\mathbf{N}^*$ denote the set of all finite sequences of natural numbers. Thus, an empirical method for problem (P, Π, α) is a map of type $M : \mathbf{N}^* \to (\mathrm{rng}(\alpha) \cup \{\text{'?'}\})$. What makes an empirical method "empirical" is that it never gets to see the whole, infinite input stream at once; it only gets to see ever larger initial segments and must "leap" from the current observations to some opinion whose truth may depend on what the tail of the input stream will be like for eternity. The aim of guessing is, straightforwardly enough, to find the right answer. Say that method M *solves* empirical problem (P, Π, α) *in the limit* just in case in each input stream σ satisfying the presupposition of the problem, there is a stage after which each output produced by M along σ is the (unique) potential answer in Π satisfied by σ. Notice that this success concept requires only stabilization to the right answer. The transition from error to truth is silent. No bell or halting state certifies success when it occurs.

It is easy to construct a method that solves Buridan's problem in the limit: select the answer "always hot" until a "non-hot" input is encountered and switch to the answer "not always hot" thereafter. This method not only converges to the right answer eventually; it is guaranteed, in the worst case, to retract its earlier views at most once (when "always hot" is replaced with "not always hot"). In general, a *retraction* occurs when an informative answer is dropped for some distinct answer, informative or uninformative. Retractions are the painful but unavoidable symptom of fallibility. But *needless* retractions are another matter entirely. It is desirable to minimize retractions in the design of empirical methods in much the same way that computational time and space are routinely minimized in the design of computing strategies.

The method just described *starts with* "always hot", in the sense that the method's initial output is "always hot". When a method is guaranteed to succeed with one retraction starting with h, say that the method is a *refutation method* for h. Such a method favors h over $\neg h$ until some problem arises and then prefers $\neg h$ forever after. Since the rejection of h cannot be "taken back" without another retraction, the rejection of h is analogous to the halting of a computation, for it certifies that the truth has been found. If h is true, however, the truth has been "found" from the outset but is never announced by an infallible sign. That is the characteristic situation of successful empirical science. Similarly, one can say that a method is a *verification method* for "not always hot", since it starts out with the denial of "not always hot" and switches to "not always hot" when verifying evidence is received. A verification method for h is a method that solves the problem with one retraction starting with $\neg h$.

Finally, what Buridan's argument proves is that *there is no verification method for "always hot" in the empirical problem he describes*. For suppose there were one. As a verification method, it starts with "not always hot". The method succeeds in each input stream of hot or non-hot observations, so feed it the constantly hot sequence until it outputs "always hot" (which it must, on pain of failing to converge to the right answer in the constantly hot input stream). That is one retraction. Now switch over to non-hot inputs, forcing the method to retract again to "not always hot", for a total of two retractions. Contradiction. Buridan's God is merely a colorful personification of the preceding, mathematical construction of a possible input stream on which the learner fails, a construction that depends on the intrinsic difficulty of the problem rather than upon the actual presence of a malicious agent in nature.

For another example, let $h =$ "there will be exactly one hot observation" and let the question be whether h is true. It is easy to solve this problem in the limit: output $\neg h$ until the first hot observation, output h until the second hot observation, and output $\neg h$ forever after. This procedure neither refutes nor verifies either side of the question because it retracts twice (starting with $\neg h$) if there are at least two hot observations. Furthermore, no possible method succeeds with just one retraction. For assuming that the method succeeds, God (or nature) can present cold observations until the method converges to $\neg h$. Then God can present a hot observation followed by all cold observations until the method retracts to h. One more hot observation forces a retraction to $\neg h$, for a total of at least two retractions. If the hypothesis in question is "there will be exactly one or strictly more than two hot observations", then three retractions are required, and so forth. If it is presupposed that there will be at most finitely many hot observations and the question is how many there will be, the problem is not solvable under any worst-case retraction bound. Some problems are not even solvable in the limit. Suppose you wish to know whether there will be finitely or infinitely many hot observations. Buridan's God can show you hot observations while you say "finitely many" and non-hot observations while you say "infinitely many". Whichever answer you converge to, you are wrong and if you do not converge you also fail.

Slight modifications of Buridan's problem give rise to increasingly complex problems empirically solvable in successively weaker senses. The harder problems do not merely embody the problem of induction; they involve *nested* problems of induction, of which retractions are merely the painful, outward sign. In the base case, there are problems that require no retractions at all: these are the empirical problems that involve no problems of induction at all and that can, therefore, be answered with infallibility, in analogy to solvable formal problems.

3. The "problem of computation"

It sounds perfectly natural to speak of the "problem of induction" (the impossibility of a verification procedure for many scientific questions) but one rarely, if ever, speaks of "the problem of computation" (the impossibility of a verification procedure for many formal questions). And yet, the two situations are quite similar, both on the face of it and at a deeper, structural level. In this section, I describe how formal problems give rise to degrees computational unverifiability matching the degrees of empirical unverifiability discussed in the preceding section.

To begin with, a *formal question* is just like an empirical question except that the possible input streams are replaced with single, numerical inputs. More precisely, a formal problem is a triple (P, Π, α), where P is a subset of $\mathbf{N}$ called a *formal presupposition*, Π is a partition of P called a *formal question*, and α is an injective assignment of code numbers to answers in Π called an *answer coding function*.

Formal problems present single number inputs that can be "received" all at once, whereas empirical problems present infinite sequences of numbers that can only be "received" in a piece-meal fashion. So far, Hume's position seems right: in formal problems you *have* the input and you *have* the concept to be applied to it, so you merely have to focus your "mind's eye" on the two of them to see with mathematical certainty whether the concept applies to the input. In empirical reasoning, infallibility may be impossible because you never see the whole input stream all at once.

But the dichotomy between infallible formal reasoning and fallible empirical reasoning was already questionable in Hume's day. If full "clarity and distinctness" could be achieved on each input, then formal reasoning would, indeed, always terminate with certainty. But if full "clarity and distinctness" is never achieved on some inputs, and if the process of achieving it has bumps and surprises along the way, one may as well think of formal reasoning as an ongoing, fallible process analogous to empirical inquiry [16].

The theory of computability underscores the preceding point with mathematical precision. In the familiar halting problem, the input domain is the set $\mathbf{N}$ of all natural numbers and the question is whether the Turing machine with code number n eventually returns an output when started on input n. Let K denote the set of all n for which the answer to the question is affirmative. Let $\alpha(K) = 1$ and $\alpha(\mathbf{N} - K) = 0$. Then the *halting problem* is the triple $(\mathbf{N}, \{K, \mathbf{N} - K\}, \alpha)$. In the ensuing discussion, I will use $\neg K$ as an abbreviation for $\mathbf{N} - K$.

Intuitively speaking, the difficulty posed by the halting problem is empirical. When an algorithm takes a long time to return an answer, one begins to suspect that the algorithm will never terminate, but how can one be sure? No amount of waiting yields certainty that the computation will never halt, any more than it can result in certainty that every observation will be "hot".

This empirical argument falls short of a proof, however, for perhaps the achievement of clarity and distinctness with respect to the input and the concept to be applied to it involves something more clever than just sitting around and waiting for a simulated computation to halt—after all, you already have the program and the input and everything about the computation is mathematically determined by this pair. As it happens, such means are bound to fail, but the usual proof of this fact is a static, diagonal argument with more affinity to Cantor than to Hume.

The strong impression that the halting problem involves something like the problem of induction is vindicated, however, by an alternative proof strategy that looks quite similar to Buridan's argument for inductive skepticism.[7] As in Buridan's argument, the proof shows that there is no computable *verifier* for non-halting (i.e., for $\neg K$), in the sense of a Turing machine that eventually halts with 1 (i.e., "yes") if and only if it is provided with the index of a machine that does not halt on its own index. The "fooling" strategy of Buridan's God can be implemented against a would-be verifier

[7] Such an argument is attributed to Scott in [1].

M_m of $\neg K$ as follows. Let d be a Turing machine index with the following property. On arbitrary input x, the Turing machine M_d indexed by d ignores x altogether and simulates the computation of M_m on input d. If the simulated computation of M_m on input d ever halts with the answer $\alpha(\neg K)$, M_d returns some arbitrary output (say, whatever M_m halts with on input d) on input x. If the simulated computation of M_m on input d never halts, M_d simulates it forever and never produces an output on input x. So it is as if machine M_d "pretends" never to halt on input d (or on any other input) until M_m becomes irrevocably convinced that M_d never halts on input d. Then M_d halts on all inputs, including d, so d is in K, but M_m has irrevocably committed itself (by halting with $\alpha(\neg K)$) to the view that d is in $\neg K$. If M_m never halts on input d, then M_d never halts on input d, so d is in $\neg K$, but M_m never halts with output $\alpha(\neg K)$. Either way, d witnesses that M_m fails to verify $\neg K$.

This is essentially an instance of Buridan's skeptical argument, with M_d in place of Buridan's God and "never halts" in place of "never cold". True, the "fooling strategy" in this argument does not have control over empirical inputs to the would-be verifier to be fooled. But the proposed fooling strategy M_d modifies its own behavior through time as it watches the would-be verifier use all the means at its disposal to try to figure out what M_d will do in light of the very code of M_d. So it is as though the fooling strategy M_d has replicated Buridan's classical fooling strategy in the would-be verifier's *internal* experience.

It is easy to sympathize with Hume here. The input is whatever it is and the deductive means you apply to it are entirely up to you, so how can your own, inner experience of what you have *already* received be hijacked by a malicious skeptical strategy? The key to the argument is the self-referential assumption that there exists a Turing machine M_d that can simulate the computation of would-be verifier M on input d. By modifying its own behavior in light of the outcome of the computation $M(d)$, machine M_d effectively seizes control of M's internal experience. It suffices to obtain M_d up to input–output behavior, since the manner in which M_d produces its input–output behavior is irrelevant to the membership of d in K. Let M_u be the universal Turing machine,[8] which has the property that for each Turing machine index i and input $y, M_u(i, y)$ returns the output (if any) of the computation $M_i(y)$. Recall that m is the index of the Turing machine we wish to "fool". Hence, the partial function $\psi(y,x) \approx \phi_u(m, y)$,[9] whose value is the result of ignoring x and passing along the result of simulating M_m on input y, is Turing computable. To complete the construction, one must show that there exists a Turing machine index d such that for all x, $\phi_d(x) \approx \psi(d,x) \approx \phi_u(m,d)$. In other words, $\phi_d(x)$ passes along $M_m(d)$, the final response (if any) of would-be verifier M_m of $\neg K$. To obtain such a d, apply the *s-m-n* theorem to obtain a total, computable function s such that $\phi_{s(y)}(x) \approx \psi(y,x)$. Then by the Kleene recursion theorem, there exists a Turing index d such that $\phi_{s(d)} = \phi_d$.

[8] The basic computability results cited in this paragraph are presented in many texts on the theory of computable functions. A nice, elementary source is [2].

[9] The relation $\approx$ signifies that either both functions have the same definite value or that both functions are undefined.

Although $\neg K$ is not computationally verifiable, $\neg K$ is computationally *refutable* (in the sense that some Turing machine halts with "no" on input n iff n is in K). On input n, simply simulate the computation of M on input n until the computation halts and halt with "no" when it has done so. So just like "all observations are hot", the formal question "the computation of M on input n never halts" is computationally refutable and not verifiable.

The preceding connection between formal and empirical reasoning is strengthened by allowing Turing machines to output successive answers on an output tape in response to a given input without ever halting. One can then redefine computational verification just as in the empirical case, as convergence to the right answer with at most one retraction starting with "no", and similarly for refutation. This is equivalent to the usual definition of verifiability in terms of halting.[10] Say that a Turing machine solves a formal problem *in the limit* just in case the machine converges to the right answer eventually, no matter which possible input is provided.

Formal problems can also involve close analogues of the "nested" problems of induction mentioned earlier. Let K_1 denote the set of all Turing machine indices i such that the computation of M_i on input i returns exactly one output (in sequence, according to the convention described in the preceding paragraph). There is an obvious method for solving K_1 with 2 retractions starting with "no": just simulate the computation M_i on input i. Say "no" until an output is produced, say "yes" until a second output is produced and then say "no" forever after. Also, an extended skeptical argument shows that two retractions are not enough starting with "yes". Let d be a Turing machine index that feeds itself to the given machine M and refuses to produce any outputs until the computation of M retracts to "no". Then M_d writes one output on its output tape and refuses to write any more outputs until M retracts to "yes". Finally, M_d writes another output, forcing M to retract again to "no", for a total of three retractions. Putnam [19] noticed the analogy between such formal predicates and the problem of induction and referred to formal predicates that can be decided with n retractions as *n-trial predicates*. The theory of such predicates is tidier if one also keeps track of whether the first output is "yes" or "no", as in the empirical case [11].

Next, suppose you know in advance that you will be given only indices of machines that produce at most finitely many outputs and the question is how many outputs a given machine will produce. The obvious method is to count the current number of outputs of the simulated computation. That method does not succeed under any finite retraction bound, but no method possibly could. For let M aspire to succeed with k retractions. Index d can feed itself to M and elicit M to $k+1$ retractions by the preceding recipe. Since d produces at most $k+1$ outputs, it satisfies the problem's formal presupposition.

To obtain a formal problem that is not even computably solvable in the limit, ask whether the given index gives rise to an infinite sequence of outputs (just what was presupposed in the preceding example). To see why, mimic the empirical, skeptical

[10] Given a verifier in the original sense, simulate its program on the given input and periodically output "no" until the simulated verifier halts with "yes". Thereafter, output only "yes". In the other direction, simulate the new-style verifier until it outputs its first "yes". Then output "yes" and halt.

argument presented in the preceding section, using the Kleene recursion theorem to achieve self-reference in the manner illustrated in the preceding examples. Thus, each of the empirical problems in the preceding section has been shown to have an analogue in the formal domain that is solvable in a closely analogous, fallible sense.

4. Topological complexity

The structural features of empirical problems that are responsible for the problem of induction are neither logical nor probabilistic, but *topological*. A *topological space* consists of a set W, together with a collection of subsets of W that are called *open* sets. The open sets of a topological space on set W must satisfy the following, four axioms:

(1) W is open.
(2) $\varnothing$ is open.
(3) Arbitrary unions of open sets are open.
(4) Finite intersections of open sets are open.

Think of W as a set of *possible worlds* or ways the world might be for all one knows. Let *propositions* be subsets of W (i.e., each proposition is identified with the set of worlds in which it is true). Here is an informal argument that the verifiable propositions over W constitute the open sets of a topological space, where verifiability means that there exists an empirical method of some sort that halts with "yes" if the proposition is true and that always says "no" otherwise.

The set W is the vacuous proposition that is true in all possible worlds. This proposition is trivially verifiable (say "yes" no matter what). The contradictory proposition $\varnothing$ is verifiable (say "no" no matter what). Finite conjunctions (intersections) of verifiable propositions are verifiable (wait for a "yes" for each conjunct before returning "yes") and an arbitrary disjunction (union) of verifiable propositions is verifiable (wait for a "yes" for at least one disjunct before returning "yes"). Hence, the verifiable propositions are the open sets in a topological space on W, which may be called *verifiability space*. Furthermore, axiom (3) cannot be strengthened to arbitrary intersection under this interpretation, for suppose you have an infinite conjunction of verifiable propositions. The respective verifications could arrive at ever later times, so there is no time by which you can be sure that all of the conjuncts are verified (the problem of induction). So the striking asymmetry between axiom (3) and axiom (4), which is characteristic of all topological reasoning, is a reflection of the problem of induction. Topology is often thought of as "plastic geometry". It is equally, if not more generally, the mathematical theory of ideal verifiability.

Here is another way to make a similar point. Let proposition h be non-open in the verification topology. Since h is non-open, h contains a limit point w of $\neg h$, so every open neighborhood of w catches an element of $\neg h$. In other words, $\neg h$ is false in w, but every verifiable proposition true at w is consistent with $\neg h$. Since each evidential proposition true of w is verifiable and finite conjunctions of verifiable propositions are verifiable, w is a world at which $\neg h$ is false, but each finite body of evidence true of w is compatible with $\neg h$. That is the problem of induction! So the problem of induction is a topological invariant of empirical problems.

In the preceding discussion I assumed some intuitive features of verifiability and showed that they imply the closure axioms for open sets in a topological space. It is also revealing to show that in a naturally selected topology on input streams, the open sets are precisely the verifiable propositions (in the explicit sense of empirical verifiability defined in Section 2 above). Let θ be a finite sequence of natural numbers. Let $[\theta]$ denote the set of all infinite sequences that extend θ. Let *basic* open sets be sets of form $[\theta]$. Let *open* sets be unions of basic open sets. Now restrict the resulting space to the empirical presupposition P. This is a widely studied topological space [5] called the *Baire space* restricted to P. Now it can be proved (rather than intuitively assumed, as in the preceding paragraph) that open sets are verifiable: wait until the input stream extends a basis element contained in the open set before saying “yes”. The converse can also be proved: if h is verifiable, then h may be expressed as the union of all basis elements corresponding to finite input sequences on which the method says “yes”. Dually, the refutable propositions are exactly the closed propositions and the decidable (verifiable and refutable) propositions are the *clopen* (closed and open) propositions.

A limit point of $\neg h$ in the restricted Baire space is an input stream whose finite initial segments can always be extended to input streams in $\neg h$. So if a limit point of $\neg h$ happens to satisfy h, then h is true but inputs never guarantee the truth of h, which is again the problem of induction. So the problem of induction arises, topologically speaking, precisely when the actual input stream is a limit point of a false answer. This happens exactly when the actual world is on the *boundary* of at least two answers (i.e., it is a limit point of both answers). So *the problem of induction is the problem of boundary points.*

Verification and refutation make sense only with respect to a fixed hypothesis h. More generally, an empirical problem (in the sense defined above) is solvable with zero retractions iff each answer is open: just wait for verification of a potential answer. Since the answers constitute a partition of the presupposition, it follows that each potential answer is also closed, or clopen for short. So the easily solved empirical problems can be characterized in terms of the topological structure of the problems themselves.

The idea generalizes to problems requiring k retractions. The *difference complexity* [18] of an empirical problem (P, Π, α) is no greater than k iff there exists a finite, ascending sequence $(S_0 \subseteq \cdots \subseteq S_k)$ of open sets such that

1. $S_k = P$,
2. for each $i \leqslant k$, each potential answer in Π is open in the restricted space $S_i - S_{i-1}$, where by convention $S_{-1} = \varnothing$.

It follows that *a problem is solvable with k retractions iff it has difference complexity k*. [11]

[11] Given a method that succeeds with k retractions and given $i \leqslant k$, let S_i be the set of all input streams on which the method retracts at least $k - i$ times. Then S_i is open because it is a union of basic open sets, so (1) is satisfied. Also, the method retracts along an input stream in $S_i - S_{i-1}$ exactly $k - i$ times. Since the method succeeds, the answer output by the method after the $k - i$th retraction is true. So answer A is definable within $S_i - S_{i-1}$ as the set of all input streams on which method M produces A after retraction $k - i$, which is a union of basic open sets in the restricted space $S_i - S_{i-1}$, so (2) is satisfied. Conversely, suppose that the difference complexity of (P, Π, α) is k. Let method M output answer A iff A is verified given $S_i - S_{i-1}$, where i is least such that S_i is verified by the current inputs (recall that $S_{-1} = \varnothing$ by convention).

This condition can be trivially reformulated in a way that will make the analogy to computability easier to see. Impose the *discrete topology* on $\mathbf{N}$, in which every singleton $\{n\}$ is open. Thus, a map f from the Baire space to $\mathbf{N}$ is *continuous* iff for each n in $\mathbf{N}$, the inverse image $f^{-1}(n)$ is open in the Baire space. Since each answer in Π is the pre-image of some code number under the composed mapping $(\alpha \circ \mathrm{ans}_\Pi)$, it follows immediately that the difference complexity of an empirical problem (P, Π, α) is no greater than k iff there exists a finite, ascending sequence $(S_0 \subseteq \cdots \subseteq S_k)$ of open sets such that

1. $S_k = P$,
2. for each $i \leqslant k$, the function $(\alpha \circ \mathrm{ans}_\Pi)$ is continuous on the restriction of the Baire space to $S_i - S_{i-1}$, where by convention $S_{-1} = \varnothing$.

Topological properties of sets often have "point-wise" characterizations. For example, an open set is just a set whose members are all interior points. In the present application, an input stream is an interior point of an answer just in case it eventually presents inputs that verify the answer. The idea generalizes in a natural way to k retractions. Define

1. input stream σ is a 0 *interior point* of problem (P, Π, α) iff σ is an interior point of some answer A in Π;
2. input stream σ is a $k+1$-*interior point* of (P, Π, α) iff σ is an interior point of some answer A in Π in the problem that results when P is restricted to input streams that are not k-interior points of (P, Π, α); [12]
3. input stream σ is a k-*limit point* of (P, Π, α) iff σ is *not* a k-interior point of (P, Π, α).

It follows that finite sequence $(S_0 \subseteq \cdots \subseteq S_k)$ witnesses that (P, Π, α) has difference complexity $\leqslant k$ iff for each $i \leqslant k$, S_i contains only i-interior points of (P, Π, α). Hence, a problem is solvable with k retractions iff it contains only k-interior points. So the k-limit points are the input streams in which one faces at least a $k+1$-fold problem of induction: removing them from the problem results in a problem solvable with just k retractions.

Since k-limit points are where k-fold problems of induction are faced, it is worth taking a closer look at them. A 0-limit point is just an input stream satisfying an answer A such that no matter how much you have seen, there is an input stream satisfying some other answer compatible with what you have seen already. This is just a single problem of induction, as in Buridan's example of inferring "always hot". A $k+1$ limit point is an input stream satisfying an answer A such that no matter how much you have seen, there is a k-limit point satisfying some distinct answer compatible with what you have seen already. For example, the input stream in which no hot observations are seen is a 2-limit point in the problem in which it is known that the color will change at most two times and the question is how many times.

Solvability in the limit has its own topological characterization. A $\mathbf{\Sigma}^0_2$ Borel set is a countable union of closed sets. *An empirical problem is solvable iff each answer is* $\mathbf{\Sigma}^0_2$

[12] The concepts k-limit point and k-interior point can be extended by transfinite induction over an extension of the ordinals giving rise to a transfinite version of the relationship between retractions and empirical complexity [12].

[10]. This condition is equivalent to saying that there is an infinite, increasing, nested, ω-sequence $(S_0 \subseteq \cdots \subseteq S_k \ldots)$ of open sets such that

1. $\bigcup_{i=0}^{\infty} S_i = P$ and
2. for each i, each potential answer in Π is open in the restricted space $S_{i+1} - S_i$.

Recall that the finite retraction characterization is the same, except that the nested sequence of open sets is finite, which provides a nice, structural insight into the difference between the two cases.

5. Formal complexity

Closely analogous concepts of structural complexity apply to strictly computational problems. Say that a function is *computable over restricted domain P* iff there exists a Turing machine M that returns $f(x)$ for each input x in P. Now define that the *effective difference complexity* of formal problem (P, Π, α) is $\leqslant k$ iff there exists a finite, increasing sequence $(S_0 \subseteq \cdots \subseteq S_k)$ of recursively enumerable sets such that

1. $S_k = P$ and
2. for each $i \leqslant k$, the function $(\alpha \circ \mathrm{ans}_\Pi)$ is computable over the restricted domain $S_i - S_{i-1}$, where by convention $S_{-1} = \varnothing$.

Then it follows that *a formal problem is effectively solvable with k retractions iff it has effective difference complexity k.*[13] This is just the topological characterization of empirical success "recursively enumerable" in place of "open" and "computable" in place of "continuous". The corresponding analogy

$$\text{open: recursively enumerable} :: \text{continuous} : \text{computable},$$

is a familiar heuristic in descriptive set theory [18].

There is a point of disanalogy, however, for effective difference complexity admits of no point-wise characterization. Recall that an empirical problem has difference complexity exceeding k iff the problem contains a k-boundary point. Hence, adding a *single* k-boundary point to a problem solvable with k retractions makes the problem intrinsically harder. But there is no single input one can add to a formal problem to make it intrinsically harder. For any single input n that is added to formal presupposition P in

[13] Suppose Turing machine M solves (P, Π, α) with k retractions. Then let S_i denote the set of all n such that M retracts at least $k - i$ times on input n. Let x be in $S_i - S_{i-1}$, where $i \leqslant k$. Then M retracts exactly $k - i$ times on input x. Let $f_i(x)$ be the $k - i$th output of M on input x. Observe that f_i is computable (using M as a subroutine) and the domain of f_i covers $S_i - S_{i-1}$. Also, since M succeeds with k retractions, f_i agrees with $(\alpha \circ \mathrm{ans}_\Pi)$ over the restricted domain $S_i - S_{i-1}$. Conversely, let $(S_0 \subseteq \cdots \subseteq S_k)$ witness that the effective difference complexity of (P, Π, α) is no greater than k. Since each such S_i is recursively enumerable, let Turing machine M_i formally verify membership in S_i. Also, for each $i < k$, let Turing machine L_i compute $(\alpha \circ \mathrm{ans}_\Pi)$ over restricted domain $S_i - S_{i-1}$. On input x, let the computation of M on input x proceed in stages as follows. At stage n, let m be the least i such that M_i halts on input x with output 1 within n steps of computation. Then return the result of the computation of L_m on input x. By construction, M retracts at most k times on input x. Let x be in P. To see that M converges to the right answer on input x, let i be the unique value such that x is in $S_i - S_{i-1}$. Eventually, a stage n is reached at which $M_i(x)$ returns 1 in n steps of computation. Thereafter, $M(x)$ returns the result of the computation $L_i(x)$, which is the correct answer $(\alpha \circ \mathrm{ans}_\Pi)(x)$.

a formal problem (P, Π, α) solvable with k retractions, there is some Turing machine that employs a rote "lookup table" to associate n with the right answer for n and that passes control to a k-retraction solution to the problem.

Formal problems that are not solvable under any retraction bound are also structurally analogous to empirical problems with the same property, for a formal problem with finitely many possible answers is solvable in the limit iff each answer is a $\boldsymbol{\Sigma}_2^0$ arithmetical set, where such a set has form

$$S = \bigcup_{i \in R} \neg W_i,$$

where R is a recursively enumerable set and W_i is the (recursively enumerable) domain of ϕ_i. In other words, countable unions of closed sets in the empirical picture are replaced with r.e. unions of complements of recursively enumerable sets in the formal picture. In general, Borel complexity in topology is analogous to arithmetical complexity in the theory of computability. This analogy is another familiar theme in descriptive set theory [18].

6. Index problems

The analogy between formal and empirical reasoning is tighter still if one focuses on a special collection of formal problems sometimes referred to as *index problems*. [14] An index problem is a formal problem in which the natural number input is viewed as the index of a Turing machine and the question posed concerns only the input–output behavior of the machine indexed by the numerical input. Equivalently, an index problem is a formal problem in which no two numbers that index the same partial computable function satisfy distinct answers.

An index problem is *non-trivial* iff its question has at least one answer that is neither $\mathbf{N}$ nor $\varnothing$. Rice's theorem [2] says that *no non-trivial index problem is effectively solvable without retractions*. The theorem can be proved by means of a "skeptical argument". Let (P, Π, α) be a non-trivial, index problem. Since the problem is an index problem, all indices for the everywhere undefined function $\varnothing$ are in some answer A in Π. Since the problem is also non-trivial, there is some distinct function ϕ whose indices are all in some distinct answer B in Π. Let M be a would-be decision procedure for (P, Π, α). On an arbitrary input x, let the "fooling strategy" M_d simulate the computation of M on input d (via Kleene's recursion theorem) until such time as M returns the unique answer true of all indices for the everywhere undefined function. Thereafter, the fooling strategy returns the result of simulating a program for ϕ on input x. In short, M_d refuses to produce any outputs until M becomes sure that M_d will never produce any outputs, and then produces outputs in accordance with ϕ (note the analogy to Burdian's skeptical argument). Hence, if M never concludes that d is in A, then d is

[14] One usually speaks of "index sets" rather than "index problems" due to the penchant of computability theorists for focusing on binary questions.

indeed in A and if M ever does conclude that d is in A, then d is in B, so M fails to decide the problem in either case.

The story extends to formal verifiability. A simple-minded, empirical strategy for determining some feature of the input–output behavior of a given index is to perform "computational experiments" on the indexed program, running it for various amounts of time on various inputs to see what sorts of outputs are produced.[15] A much more sophisticated approach would involve some formal analysis of the code of the program, itself. The Rice–Shapiro theorem [2] is the remarkable claim that a computational agent can determine no more about the input–output behavior of an *arbitrary* program by looking at the program than it could by performing computational experiments on it, treating it as an otherwise unknown "black box", for *if an empirical agent could not verify the input–output property from experiments*, *then no amount of effective analysis of the code could formally verify the same property over arbitrary Turing machine indices.* This is not a mere analogy: it is a deep and striking mathematical relationship between empirical and formal reasoning that holds for all index problems.

Some topological concepts are required to state the theorem precisely. Let θ be a finite set of input–output pairs (i.e., a finite function). Let $[\theta]$ be the proposition that these input–output pairs have occurred (i.e., $[\theta]$ is the set of all partial computable functions extending finite function θ). Let open propositions about input–output behavior be unions of propositions of form $[\theta]$. By arguments similar to those already given, the open propositions in this sense are exactly the propositions about input–output behavior that could be verified empirically by watching a black box that conceals an unknown Turing machine, to which inputs can be provided from outside.

If S is a set of indices, then let S_e denote the set of all partial computable functions with indices in S. If (P, Π, α) is a formal index problem, then let the *empirical problem generated by* (P, Π, α) be the triple (P_e, Π_e, α_e), where Π_e is just $\{A_e$: A is in $\Pi\}$ and $\alpha_e(A_e) = \alpha(A)$. The Rice–Shapiro theorem then says that *answer A is formally verifiable in problem* (P, Π, α) *only if answer A_e is empirically verifiable in the empirical problem* (P_e, Π_e, α_e).

The Rice–Shapiro theorem can be proved by means of yet another skeptical argument. Suppose that $(\mathbf{N}, \Pi, \alpha)$ is an index problem with possible answer S. Suppose, further, that the set of functions Γ whose indices are all in S is not open in the topological space just described in the preceding paragraph. Then there exists $\psi \in \Gamma$ such that ψ is a limit point of the complement of Γ (with respect to the space of all partial computable functions). In other words, $(*)$ each finite subfunction of ψ is extended by some partial computable function in the complement of Γ.

Consider the case in which some finite subfunction θ of ψ is also in Γ. Then by $(*)$, some partial, computable ξ extending θ is not in Γ. Implement a "fooling strategy" M_d for an arbitrary, would-be formal verifier M of S, as follows. On input x, let M_d jointly simulate in parallel the computation of M on d (via Kleene's recursion theorem) and the computation $\theta(x)$. If the latter computation halts first, then output $\theta(x)$. If $M(d)$ halts first, pass control to a computation of $\xi(x)$ and return the result, if any. Hence,

[15] This is called "dovetailing" [2].

M_d computes θ (so that $\alpha(S)$ is correct of d) if M never halts with $\alpha(S)$ on input d. Otherwise, M_d computes ξ (so that some answer other than $\alpha(S)$ is correct). So M fails to verify S.

Now consider the alternative case in which every finite sub-function of ψ is in the complement of Γ (so that ψ, itself, is infinite). Implement a "fooling strategy" M_d for an arbitrary, would-be formal verifier M of S, as follows. On input x, let M_d simulate the computation of $M(d)$ (via Kleene's recursion theorem). If M does not halt with answer $\alpha(S)$ within x steps of computation, output $\psi(x)$. Else, go into a gratuitous loop on input x. Hence, M_d computes ψ (so that $\alpha(S)$ is correct of d) if M never halts with $\alpha(S)$ on input d. Otherwise, M_d computes a finite sub-function of ψ (so that some answer other than $\alpha(S)$ is correct). So M fails to verify S.

The Rice–Shapiro theorem assumes that, for all you know, you may receive as input *any* element of $\mathbf{N}$. If only indices with a special property are expected, the Rice–Shapiro theorem may fail. For example, suppose that P contains all Turing machine indices for finite functions and a single index z for the zero constant function $\mathbf{0}$. Suppose the formal question Π is whether the given index computes $\mathbf{0}$. The answer "zero constant function" is not open in the empirical problem generated by this problem, but the formal problem is nonetheless solvable by a Turing machine that maintains a lookup table with index z written on it. The proof of the Rice–Shapiro theorem fails in this example because the Kleene recursion theorem may produce a tricky index that is not in P.

The Rice–Shapiro argument generalizes to index problems requiring k retractions in the following way: *if* $(\mathbf{N}, \Pi, \alpha)$ *is formally solvable with k retractions*, *then* $(\mathbf{N}_e, \Pi_e, \alpha_e)$ *is empirically solvable with k retractions*. So iterated problems of induction give rise to iterated formal retractions in the corresponding formal problem. The proof iterates the two cases of the Rice–Shapiro theorem. Suppose that (P_e, Π_e, α_e) has difference complexity $>k$. Then $(\mathbf{N}_e, \Pi_e, \alpha_e)$ has a partial recursive k-limit point ψ. When $k=0$, it follows that ψ is a limit point of some answer A_e. The two cases of the proof of the Rice–Shapiro theorem now arise: either each finite sub-function of ψ satisfies a distinct answer, or some finite sub-function θ of ψ satisfies A_e, in which case some proper extension of θ satisfies a distinct answer, since ψ is a limit point of a distinct answer. In either case, a fooling strategy can be constructed. When $k>0$, no finite sub-function of ψ is a $k-1$-interior point, for if it were, then ψ would be as well. Hence, the set of all input streams of complexity greater than $k-1$ includes all finite sub-functions of ψ. Again, either none of these functions satisfy A_e or one of them does. If none does, construct a fooling strategy that pretends to be ψ until M concludes A (the set of all indices of functions in A_e) and that pretends to be a finite sub-function θ of ψ until M concludes the (distinct) answer satisfied by θ. After that, since θ is itself a $k-1$-limit point, the induction hypothesis guarantees that control can be passed to a fooling strategy that achieves another k retractions, for a total of $k+1$. If some finite sub-function θ of ψ satisfies A_e, then since ψ is a k-limit point of some answer incompatible with A_e, it follows that among $k-1$-limit points, there exists a $k-1$-limit point ξ satisfying a distinct answer from A_e that extends θ. A fooling strategy can pretend to be θ until M concludes A_e and can pretend to be ξ until M retracts to the incompatible answer satisfied by ξ. Since ξ is a $k-1$-limit point, the induction hypothesis says that control can be passed to a fooling strategy that achieves k more retractions, for a total of $k + 1$.

7. Empirical simplicity and Ockham's razor

A characteristic feature of empirical science is *Ockham's razor*, a preference for simple theories when several competing theories account for the current data. But why? There is no shortage of explanations: we like simplicity, simpler theories are easier to understand or compute with, simple theories explain better or are easier to cross-check, etc. But such arguments are instances of wishful thinking, for the simplest theory might be false, regardless of our good reasons for wishing it to be true and the task of science is to find the truth, not to varnish it. If one prefers the simplest theory because one knows in advance that the world is simple, then the complex alternative theories are not really alternatives after all and the empirical question is trivial (it has just one possible answer). If the simplest answer is assumed to be more a priori probable than the other answers, then the other answers probably are not real alternatives. If one prefers the simplest theory because it is better "confirmed or supported" than the other theories, the question arises afresh: what do "confirmation" or "support" have to do with finding the true answer? If one uses the simplest theory to accurately predict new observations even when we know that the simplest theory is false (as in linear regression), then one concedes that Ockham's razor is *opposed* to finding the true theory. In each case, it is hard to see how Ockham's razor could serve the interest of finding the truth. The connection between truth and simplicity is arguably the most fundamental puzzle in the philosophy of science and induction.

Here is an answer to the conundrum that fits with the convergent perspective on inquiry discussed above: choosing the simplest theory compatible with experience is necessary if we are to minimize the number of times we retract earlier answers en route to the truth in the worst case (which, incidentally, will be a *complex* rather than a simple world) [12]. Hence, simplicity does not *indicate* the truth (the world may be complex and may even probably be complex) but simplicity nonetheless helps us to find the truth in the sense that any other bias results in avoidable, worst-case inefficiency *en route* to the truth.

For a very rudimentary illustration of the argument for this claim, suppose you know that there are at most three golf balls in a box and the question is how many balls there are. Each ball is exhibited, without replacement, at some time of Nature's choosing. There are four intuitive senses in which "no balls are in the box" is the simplest of the three possible answers to this problem. First, it involves the least *existential commitment* of all the answers, since it posits no balls. Indeed, Ockham's original statement of his principle was to not multiply entities beyond necessity. Second, it is most *uniform*, in the sense that it is satisfied only if no ball ever appears, whereas the other answers imply a mixture of "no new ball" experience with "new ball" experience. Third, it is the most *testable*, in the sense that it is empirically refutable (if it is false, the appearance of a ball eventually establishes this fact) but the other answers are not refutable (e.g., "one ball" is false in the zero-ball world but is consistent with any finite amount of ball-free experience). Fourth, it has *fewer free parameters* than the other potential answers. If there is no "hot" observation, then there is no question as to when hot observations occur, but if there is a hot observation, it must occur at some time t_1 and if there are two, they must occur at distinct times t_1, t_2, etc.

Now suppose that a method prefers an answer other than "no balls" prior to seeing any balls. Nature can continue to exhibit ball-free experience until the method concludes "no balls" on pain of converging to the wrong answer. Then nature can present a ball followed by ball-free experience until the method concludes "one ball", etc., for a total of four retractions. But an alternative method succeeds with at most three retractions in the worst case: just output "n balls", where n is the number of balls seen so far. That method follows Ockham's razor at each stage.

The *difference complexity* of an *answer* to a problem can be defined as the greatest k such that the answer contains a k-interior point. The answer "no balls" has difference complexity 3, the answer "one ball" has difference complexity 2, and, in general, the answer "n balls" has difference complexity $k-n$, where k is the known upper bound on the number of balls. So simpler answers (in the intuitive, scientific sense) have higher difference complexity in the topological sense. More generally, one may think of difference complexity degrees as degrees of empirical simplicity. Such intuitive reflections of simplicity as uniformity of experience, minimal existential commitment, testability, and fewer independent parameters tend to line up with high difference complexity in a given empirical problem.

Ockham's razor is vaguely understood to be a preference for the simplest hypothesis compatible with current experience. However, in the sense just defined several answers can have the same, maximum simplicity degree. In such cases, the proposed version of Ockham's razor says that one may not select an answer unless it is currently the *unique* answer of maximum complexity. Intuitively, this makes sense: if several hypotheses are simplest, simplicity cannot guide the choice among them.

Suppose that the maximum simplicity degree (i.e., the problem's difference complexity) is n and that method M violates Ockham's razor by choosing a hypothesis that is not uniquely simplest, among hypotheses compatible with experience so far. Then nature can continue to present inputs compatible with some distinct, simplest answer A until M converges to A on pain of converging to the wrong answer, which counts as one retraction. Thereafter, Nature can exact n more retractions as before, for a total of $n+1$.

Furthermore, an obvious method that complies with Ockham's razor succeeds in each case with no more than n retractions. The method outputs the (unique) answer A verified relative to the assumption that the world has simplicity degree $= k$, where k is least such that it is currently verified that the world has simplicity degree $\leqslant k$. This method retracts only when it is verified that the world has a lower simplicity degree than previously thought, and hence retracts at most n times. It converges to the right answer because eventually it is verified that the world has simplicity degree $\leqslant k$, where k is the true simplicity degree, and then the true answer is verified relative to the assumption that the world has simplicity degree $= k$.

It follows from the two preceding paragraphs that *for each problem of finitely bounded difference complexity*, *violating Ockham's razor on the initial conjecture* *results in a higher than necessary worst-case bound on retractions*. For subsequent outputs the argument is similar: violating Ockham's razor then results in a needlessly high, worst-case bound on retractions in the *sub-problem* faced from that point onward. [16]

A paradigmatic application of Ockham's razor is curve fitting. Suppose you know that the curve to be fit is a polynomial of degree no greater than three. The question is to guess the polynomial degree, where it is known that the true curve has degree $\leqslant 3$. That is not very hard: two points determine a line, three points a quadratic, four a cubic, etc. But the game is more interesting when the data points may contain error. Consider a simplified version of curve-fitting in which the method may query any data point and the data points may involve less than $\varepsilon > 0$ error. In this problem, the polynomial degrees run in inverse order to simplicity degrees, so that the answer "cubic" has simplicity degree zero, the answer "quadratic" has simplicity degree one, and so forth. Suppose that the data points seen so far are all closer than ε to some constant c and that a given method violates Ockham's razor by saying that the degree of the true function exceeds zero. Nature can present data within ε of c forever until the method converges to "degree zero". Thereafter, Nature can choose a slightly inclined line that still saves all the data presented so far to within ε and can then continue to present data from the line, etc. for a total of four retractions. The Ockham method that always sides with the simplest hypothesis compatible with experience requires at most three. [17]

In the preceding examples, the size of the box and the a priori bound on the degree of the unknown polynomial are necessary to arrive at a finite bound on the number of retractions required. Neither problem is solvable under any transfinite retraction bound according to the theory just mentioned, so the preceding argument for Ockham's razor does not apply. However, there is still a sense in which Ockham's method is best [20]. Think of a method as "accepting" an answer when it outputs that answer and as rejecting the answer when it outputs *any* alternative answer. Then we can view a method for a problem as a test for any given answer to the problem. It is then desirable that the method decide each answer in the limit with the fewest possible retractions. In both the ball counting problem and the curve fitting problem, a method minimizes worst-case retractions in *each* subproblem of *each* decision problem determined by an answer to the original problem only if it conforms to Ockham's razor at every stage.

8. Ockham's formal razor

It sounds odd to entertain desperate, empirical "guessing" rules like Ockham's razor in purely formal contexts, but the preceding analogies between uncomputability and the problem of induction suggest a second look.

[16] Retractions prior to entering the subproblem are not counted in the sub-problem.

[17] The problems just considered have finite complexity bounds. The retraction-efficiency argument for Ockham's razor can be extended to problems of transfinite ordinal complexity [12] based on ideas due to Freivalds and Smith [3].

In the empirical domain, Ockham's razor is a preference for the uniquely simplest answer compatible with the inputs. Recall that the simplicity degree of an answer to an empirical question is explicated by the answer's difference complexity. Similarly, the methodological simplicity of an answer to a formal question is explicated by its effective difference complexity. Let (P, Π, α) be a formal problem of effective difference complexity k and let A be an answer in Π. Define the *effective difference complexity* of answer A in (P, Π, α) to be the greatest j such that for each sequence $(S_0 \subseteq \cdots \subseteq S_k)$ of recursively enumerable sets satisfying conditions (1) and (2) in the definition of effective difference complexity, $A - S_k$ is non-empty. This is quite analogous to the definition given in the empirical case.

Ockham's razor is a rule for choosing among several possible answers compatible with current experience, but in formal problems at most one answer is compatible with a given input, so it seems that Ockham's razor is gratuitous. What is intended, of course, is that a Turing machine prefer the simplest answer compatible with the machine's "internal" experience on the path toward "clarity and distinctness", but that is a tricky concept to define in general. It makes sense for Turing machines of a certain kind (those that explicitly simulate different computations for ever greater numbers of computational steps or that seek ever longer proofs of contradictions), but not for arbitrary Turing machine programs, most of which are unintelligible.

An alternative, more "behavioristic" statement of Ockham's razor is that one should never output a simpler answer after a more complex answer has been output, where simplicity of answers can be defined in terms of effective difference complexity as was done in the empirical case. In the empirical case, this follows from the usual definition, assuming that the method converges to the truth at all, for if one at some point chooses an answer more complex than the data require, then there exists a (simple) way of continuing the data such that a convergent method must shift back to the simpler answer. The converse holds as well, if one assumes, further, that the method never produces an answer that has already been refuted. Neither argument works for formal reasoning, but one can simply *stipulate* the new statement of Ockham's razor in formal problems.

For an easy illustration, recall the problem in which it is known in advance that the input is an index of a Turing machine that produces at most three sequential outputs and the question is how many sequential outputs will be produced. Let M be a Turing machine that solves the problem in the limit. Let d be the index of a tricky Turing machine that refuses to produce an output until M says "no outputs", that produces one output until M says "one output", etc. The simplest answer is "no outputs". Suppose that M violates Ockham's razor on input d by guessing some non-zero number of outputs before it guesses "no outputs". Then the violator uses four retractions in the worst case when three would have sufficed over all inputs in the problem. Indeed, there are infinitely many variants of the fooling strategy (involving different time lags between outputs, for example) and a retraction-minimal solution to the problem must satisfy Ockham's razor on the indices constructed via Kleene's recursion theorem for all of them.

This is weaker than the empirical result, because M need not satisfy Ockham's razor on every input. Let x be an input that satisfies the simplest answer. Let M be a method that first returns the least simple answer on input x, only to converge to the simplest answer thereafter. Otherwise, M follows Ockham's razor. As long as the problem requires at least one retraction in the worst case, M succeeds under the optimal retraction bound in spite of violating Ockham's razor on some input. This is yet another consequence of the possibility of lookup tables.

For a striking example of the analogy between formal and empirical reasoning, consider a purely formal version of empirical curve fitting. Think of a total, computable function f as a map g from rationals to rationals by decoding naturals as pairs and interpreting pairs as rationals. It is assumed in advance that for some polynomial function h of degree $\leqslant 3$, $g(x)$ is always closer than $\varepsilon > 0$ to $h(x)$ (think of this as observational error). The question is to determine the least polynomial degree $k \leqslant 3$ such that for some polynomial function h of degree $\leqslant 3$, $g(x)$ is always closer than $\varepsilon > 0$ to $h(x)$.

Since the index of f determines everything about g, it "gives away" the answer to the question once for all, but to a computational agent the problem is similar to the empirical one. The tricky index d for this problem pretends to be for a constant function with error $< \varepsilon$ until M gives in and reports that ϕ_d is a constant function. This is accomplished by producing some constant, say zero, on input x if M does not output "constant" on input d in n computational steps. After M says "degree zero", d pretends to be a linear function with non-zero slope until M gives in and believes it is a linear function, etc, to exact a minimum of three retractions from M. Now, suppose that M outputs a higher polynomial degree than zero before saying degree zero. Then M retracts four times on d when the obvious Ockham method succeeds with at most three retractions in the worst case. If no upper bound on polynomial degree is known, one still obtains the result that Ockham's razor is necessary if a single computational method is to decide each answer with a minimum of retractions.

The preceding efficiency arguments assume that the violation of Ockham's razor consists of a complex guess followed by the simplest possible guess. What goes wrong if the method violates Ockham's razor by saying "$n+1$ outputs" prior to "n outputs", where "$n+1$ outputs" is not the simplest guess? In analogy with the empirical case, one may hope that there are sub-problems of the original problem that could have been solved with fewer retractions had the method not violated Ockham's razor on *some* (tricky-for-the-method) inputs. In the preceding example, let the sub-problem be all indices that produce at least one output. Let M be given. A tricky index d can be constructed that produces one output right away and then continues with the preceding strategy to exact at least two retractions from M if M solves the sub-problem. But if M ever precedes the answer "one output" with some more complex answer in response to d, M will retract at least three times even though the sub-problem is formally solvable with just two retractions. In this way, Ockham's razor applies across time in formal, as well as in empirical problems.

9. Conclusion: Hume and hypercomputability

Formal problems and empirical problems are not exactly the same. In the former, the right answer is determined by what is "given" and in the latter it usually is not. In the former, performance can always be augmented to a finite degree by means of lookup tables and in the latter it cannot (one cannot tell, without seeing the future, whether the actual, empirical world is a world *listed* on the table). Philosophical tradition has seized upon such differences to draw a sharp boundary between formal reasoning concerning mere relations of ideas and empirical reasoning concerning matters of fact. The former can supposedly be made infallible by a process of mental rigor guaranteed to terminate in clarity and distinctness; the latter cannot be infallible, since the right answer is never determined by any finite number of inputs.

I have argued for an alternative view, according to which uncomputability is an "internalized" problem of induction. True, the input is *given* all at once in a formal problem and the input ideally determines the right answer, but the input is not fully *taken* until the computational agent's journey toward full "clarity and distinctness" (i.e., its *computation*) is complete. In uncomputable problems, the process never comes to full fruition, just as empirical inquiry never halts with infallible knowledge of universal laws.

One would like a bell to ring when inquiry has succeeded. Weaker senses of success are tolerated in empirical science only because bells that signal success are infeasible. In light of the many detailed parallels between the problem of induction and uncomputability, a parallel weakening of standards is warranted in the formal domain. If a formal problem is not decidable, perhaps it is verifiable or refutable. If neither of those success concepts is feasible, then perhaps it is defeasibly solvable with no more than n retractions. If there is no such bound n, then perhaps it is decidable in the limit, etc. If empirical science can be said to progress toward the truth in spite of the problem of induction, then ordinary Turing machines can be said, on closely analogous grounds, to progress fallibly toward the truth in spite of uncomputability.

The literature on "hyper-computation" aims at an expanded but plausible sense of computability according to which Turing-uncomputable problems are solvable. There are two paths to this end. Most directly, one can try to "power-up" the computational model itself, by appealing to uncomputable oracles, by incorporating exact real numbers that encode unsolvable problems, by computing in space–times that permit one to see infinite computational traces in an instant etc. (cf. the other articles in this issue). Similarly, one can attempt to "power-up" empirical science by inventing crystal balls, by extending the scientist's present eyes and mind through all of space and time, etc. The trouble is to actually implement any of these hyper-methodologies in a manner that would inspire confidence that the implementation is correct (who checks that the real-valued parameter is set *precisely* to the right value and who checks that the crystal ball *really* reveals the future)?

An alternative approach is to retain standard computational models, whose implementation issues are (relatively) unproblematic, and to follow the lead of empirical science by relaxing the halting condition on algorithmic success when success with halting is not feasible. Such an approach contradicts neither the Church–Turing thesis nor the empirical problem of induction, for these principles govern *infallible* solvability (i.e., solvability with zero retractions). There are *extended* Church–Turing theses and problems of induction for $1, 2, 3, \ldots$ retractions, etc. and all of these theses are mutually consistent. Uncomputability is not a reason to put aside Turing machines, any more than the problem of induction is a reason to abandon empirical science.

Instead, it is a reason to seek Turing machines that converge to the truth in the strongest possible sense.

Hume held that skeptical arguments leave inductive reasoning unjustified, for they reveal it to be fallible, unlike purely formal reasoning. His challenge is to show wherein the justification of fallible reasoning consists. He was doubly mistaken. First, formal reasoning is subject to uncomputability, which is, itself, a kind of "internalized" problem of induction. Second, a method of reasoning (like any other strategy) is *justified* in a given problem insofar as it solves the problem in the best possible sense. So it is essential to the justification of a given method M to show that no possible method converges to the truth in a stronger sense than M does. That requires a skeptical argument to the effect that stronger senses of success are infeasible. Therefore, skeptical arguments are both the principal motivation for Hume's challenge *and the answer to it*. In a similar manner, generalized uncomputability arguments justify the application of convergent programs in standard programming languages to uncomputable problems.

Acknowledgements

The author is indebted to S. Burgin and P. Kugel for detailed and helpful corrections.

References

[1] H.P. Barendregt, The Lambda Calculus: Its Syntax and Semantics, North-Holland, Amsterdam, 1984.
[2] N.J. Cutland, Computability: An Introduction to Recursive Function Theory, Cambridge University Press, Cambridge, 1986.
[3] R. Freivalds, C. Smith, On the role of procrastination in machine learning, Inform. and Comput. 107 (1993) 237–271.
[4] E.M. Gold, Limiting recursion, J. Symbolic Logic 30 (1965) 27–48.
[5] P.G. Hinman, Recursion Theoretic Hierarchies, Springer, New York, 1978.
[6] D. Hume, in: C. Hendell (Ed.), An Inquiry Concerning Human Understanding, Bobbs Merrill, New York, 1984.
[7] A. Hyman, J. Walsh, Philosophy in the Middle Ages, Hackett, Indianapolis, 1982.
[8] S. Jain, D. Osherson, J. Royer, A. Sharma, Systems that Learn, 2nd Edition, MIT Press, Cambridge, 1999.
[9] W. James, The Will to Believe, in The Will to Believe and Other Essays in Popular Philosophy, Dover, New York, 1956, pp. 1–31.
[10] K. Kelly, The Logic of Reliable Inquiry, Oxford University Press, New York, 1996.
[11] K. Kelly, The logic of success, British J. Philos. Sci. 51 (2000) 639–666.
[12] K. Kelly, Efficient Convergence Implies Ockham's Razor, in: Proc. 2002 Internat. Workshop on Computational Models of Scientific Reasoning and Applications, Las Vegas, USA, 24–27 June, 2002.
[13] K. Kelly, O. Schulte, The computable testability of theories with uncomputable predictions, Erkenntnis 43 (1995) 29–66.
[14] K. Kelly, O. Schulte, Church's thesis and Hume's problem, in: M.L. Dalla Chiara, et al., (Eds.), Logic and Scientific Methods, Kluwer, Dordrecht, 1997, pp. 383–398.
[15] K. Kelly, O. Schulte, C. Juhl, Learning theory and the philosophy of science, Philos. Sci. 64 (1997) 245–267.
[16] P. Kitcher, A priori knowledge, Philos. Rev. 86 (1980) 3–23.
[17] E. Martin, D. Osherson, Elements of Scientific Inquiry, MIT Press, Cambridge, 1998.

[18] Y. Moschavakis, Descriptive Set Theory, North-Holland, Amsterdam, 1980.
[19] H. Putnam, Trial and error predicates and a solution to a problem of Mostowski, J. Symbolic Logic 30 (1965) 49–57.
[20] O. Schulte, Means-ends epistemology, British J. Philos. Sci. 51 (1999) 151–153.
[21] W. Sieg, Calculations by man and machine, in: W. Sieg, R. Sommer, C. Talcott (Eds.), Reflections on the Foundations of Mathematics, ASL Lecture Notes in Logic, Vol. 15, 2002, pp. 390–409.