KEVIN T. KELLY AND OLIVER SCHULTE

CHURCH'S THESIS AND HUME'S PROBLEM

ABSTRACT. We argue that uncomputability and classical scepticism are both reflections of inductive underdetermination, so that Church's thesis and Hume's problem ought to receive equal emphasis in a balanced approach to the philosophy of induction. As an illustration of such an approach, we investigate how uncomputable the predictions of a hypothesis can be if the hypothesis is to be reliably investigated by a computable scientific method.

1. RELATIONS OF IDEAS AND MATTERS OF FACT

Following an ancient tradition, David Hume boldly divided the objects of inquiry into two kinds: *relations of ideas* and *matters of fact* (Hume, 1984). Relations of ideas embrace all mathematical and logical inquiry, whereas matters of fact are the principal concerns of empirical science and daily life. The view that mathematics concerns relations of ideas has two important consequences. First, mathematical questions can be answered independently of all empirical data and second, the ideas upon which such questions depend can be scanned all at once by the "mind's eye," resulting in certainty concerning their relations. After a brief discussion of this happy situation in mathematics, Hume turned to the apparently more problematic case of inquiry concerning matters of fact. Here, a general law covers a potentially unbounded stream of empirical data that can refute it at any time in the future, so there is no time by which certainty about the law can be achieved. In sharp contrast with the concepts exhaustively scanned by the mathematician's inner eye, the unbounded observations relevant to an empirical generalization always outrun the scope of the empirical scientist's outer eye. *Hume's problem* is both an observation and a challenge. The observation is that inductive[1] inferences can never be certain or demonstrative in the way that mathematical inferences were supposed to be and the challenge is to explain what is good about them, other than that we habitually draw them.

Church's thesis is the proposition that the intuitively algorithmically computable functions are all recursive, or equivalently, Turing computable. Turing showed that the thesis follows from locality conditions and finite bounds on the resources and perceptual abilities of a human following an algorithm.[2] The locality and perceptual boundedness assumptions are virtually identical to those invoked in classical arguments for inductive skepticism. For example:

> I assume then that the computation is carried out on one-dimensional paper, i.e. on a tape divided into squares. I shall also suppose that the number of smbols which may be printed is finite. If we were to allow an infinity of symbols, then there would be symbols differing

[1] In this paper, *induction* will always refer to empirical inquiry rather than to mathematical induction.

[2] For an extended discussion of the relevance of Turing's argument, cf. (Sieg, 1994).

> to an arbitrarily small extent. ... The difference from our point of view between the single and compound symbols is that the compound symbols, if they are too lengthy, *cannot be observed at one glance.*
>
> The behavior of the computer at any moment is determined by the symbols which he is observing, and his "state of mind" at that moment. We may suppose that *there is a bound B to the number of symbols or squares which the computer can observe at one moment.* If he wishes to observe more, he must use successive observations.[3]

This is very different from Hume's picture in which the mathematician's inner eye exhaustively scans two ideas and recognizes with certainty that the one is contained in the other. Indeed, Turing's computer is in a situation much more analogous to that of Hume's empirical scientist. Both the scientist and Turing's computer are limited by a finitely bounded perspective: the empirical scientist can see only a finite initial segment of an infinite data stream at any given time and the Turing machine can have seen only finitely many squares on an infinite tape by any given time. In a typical sort of computability argument, the Turing machine generates instances of some formal relation successively, waiting for a test condition to be met. It is natural to speak as though the machine is seeking to "verify" or "refute" some formal hypothesis in light of formally generated "data" that arise on its tape and that cannot be anticipated in advance due to the machine's bounded perspective.

Despite this intuitive analogy, there is still a strong tendency to treat induction and computation as disparate topics.[4] In the case of formal inquiry, one commonly seeks procedures guaranteed to arrive at a correct result on any input, and uncomputability is taken seriously. In the case of empirical inquiry, it is much more usual to speak of justification, probability, confirmation, and historical case studies. Computational considerations are either deemed irrelevant or are tacked onto the end of an ideal account, as when numerical computations are used to approximate ideal Bayesian updating.[5] We question this pervasive tendency. The purpose of this paper is to argue that both Hume's problem and Church's thesis are reflections of assumptions that give rise to inductive underdetermination, and to investigate some surprising methodological issues that arise when uncomputability and inductive scepticism are accorded equal status.

2. CONVERGENCE AND CERTAINTY

One intuitive distinction between algorithms and inductive methods is that the former confer certainty whereas the latter do not. This certainty derives

[3](Turing, 1936), our emphasis.

[4]For a recent example of this tendency, cf. (Levi, 1990).

[5]There are notable exceptions to this claim. On the formal side, there has been intererest in random algorithms for primality testing. On the empirical side, Hilary Putnam (1963; 1965) explicitly proposed adopting a computation theoretic perspective on induction, and his suggestion has been developed by computability theorists and linguists under the rubric of formal learning theory, cf. (Osherson *et al.*, 1986; Angluin and Smith, 1983). Formal learning theory is the inspiration for this paper.

from two factors: (1) a logical guarantee that the algorithm will produce the right answer on each input in a specified class, and (2) the fact that the algorithm halts, thereby signalling to the user in an unambiguous way what its output is. Hume's problem might be expressed by saying that there is no procedure for inductive inference that has both properties, where inputs are taken to be the infinite data streams that may arise for all the scientist knows. The standard response to this difficulty is to exempt inductive inference from condition (1). For example, it may be required only that the method have a high personal probability of returning a correct conclusion, or that the overall outputs of the procedure be mutually coherent, explanatory, or simple. We will refer to this as the *standard approach* to inductive inference. Once the standard approach is adopted, the theory of inductive inference looks very different from the theory of computability: the former becomes preoccupied with probability distributions and their accompanying analysis, while the latter employs a distinctive kit of tools—diagonalization, dovetailing and reduction—characteristic of mathematical logic. The many differences between these alternative conceptual schemes in turn reinforce Hume's dichotomy between empirical and formal reasoning.

But an alternative response is to relax condition (2) without relaxing condition (1), so that an inductive method is guaranteed to *converge* to the right answer, but need not inform the user when it has done so. William James, for example, endorsed this alternative, which we will refer to as *logical reliabilism*[6]:

> We may talk of the *empiricist* and the *absolutist* way of believing the truth. The absolutists in this matter say that we not only can attain to knowing truth, but we can know when we have attained to knowing it; while the empiricists think that although we may attain it, we cannot infallibly know when. To *know* is one thing and to know for certain *that* we know is another.

Weakening the sense of convergence to an output so that there is no longer a determinate sign that the truth has been found (e.g., halting) can enlarge the range of empirical hypotheses about which we can be guaranteed to find the truth. But this strategy is not restricted to the empirical case. Even though algorithms confer certainty when we have them, Church's thesis informs us that most formal problems do not have algorithmic solutions, just as Hume's problem informs us that most empirical questions cannot be eventually decided by the data. Weakening the notion of convergence to an output in the formal case can likewise expand the scope of formal inquiry without relaxing the demand that the method succeed on every possible input.

In fact, various successively weaker notions of convergence to an answer have been entertained in both empirical and in computational contexts. Consider a system that produces a growing sequence of outputs through time. The system might be a Turing machine with an infinite output tape or it might be an inductive method producing test results in response to increasing data. The system converges to a value (say 1) *with certainty* just in case it provides

[6] (James, 1948, pp.95–96)

a determinate sign that it has found the truth. Turing machines usually signal certainty by going into a designated halting state that the user can observe. An equally good convention would be to have some special symbol (e.g., '!', for 'Eureka!') such that the first output following the first occurrence of the special symbol is the output the system is certain of. Every subsequent output is dismissed as noise, as though the method has switched itself off after finishing its job, and every output before the first occurrence of '!' is uncertain. This works in both the Turing case and in the empirical case.

Alternatively, a system converges to an output n *in the limit* just in case after some finite period of vacillation, it eventually produces only n. Finally, a system capable of outputting rational numbers converges to an output n *gradually* just in case for each rational r, there is a time after which the conjectures of the system all remain within distance r of n.

3. VERIFICATION, REFUTATION, AND DECISION

Three concepts of convergence have been defined: certain, limiting, and gradual. We turn next to the important notion of *reliability*. Empirical and formal inquiry take on different, parallel forms depending on the application. For example, a Turing machine may be called upon to *assess* a given claim (e.g., 397 is the 20th prime) or to *generate* the value of a function (e.g. produce the 20th prime). Similarly, a scientific method may be required to test a given hypothesis or may be required to generate its own hypothesis from data. In the empirical case, generation corresponds, for example, to estimation or theory discovery.[7] In this paper, we will focus on assessment problems only. On the computational side, we assume the usual definitions of Turing machines and their inputs and outputs. On the empirical side, our model of hypothesis assessment is simplified and abstract. A *data stream* is a total function on the natural numbers, to be viewed as an infinite sequence of code numbers arriving discretely through successive moments of time. The code numbers may be thought of as representing discrete experimental outcomes. An *empirical hypothesis* will be identified with the set of all data streams of which it is true. A *hypothesis assessment method* is just a mapping from finite sequences of natural numbers (i.e., finite data sequences) to rationals in the unit interval, together with the certainty mark '!'. For the purposes of this paper it suffices to think of an assessment method as being dedicated to the investigation of a single hypothesis, so we needn't provide the hypothesis as an input.

The situation of a hypothesis assessment method is very similar to that of a Turing machine. Both the Turing machine and the inductive method face a range of possible inputs: numbers to be tested for a property in the case of the Turing machine and infinite data streams against which to check the truth of

[7] A discussion of such problems under the rubric of discovery problems may be found in (Kelly, 1995). Our other paper in this volume pertains to theory discovery.

an empirical hypothesis in the case of the inductive method. In either case, there is a notion of *correct answer*: the correct answer for the Turing machine is 1 if the input number has the intended property and is 0 otherwise, whereas the correct answer for the inductive method is 1 if the hypothesis is true of the actual data stream and is 0 otherwise.

Given a criterion of convergence, there are three notions of reliable assessment that have been applied both in the theory of computability and in discussions of empirical method: verification, refutation, and decision. *Verification* requires that on every input the system converge (in the required sense) to 1 just in case 1 is the correct answer. Note that this requires that the system refuse to converge to 1 when the correct answer is 0. Similarly, *refutation* requires that on every input the system converge to 0 just in case the correct answer is 0. Finally, *decision* requires that the system converge to the correct answer whether it is 0 or 1. Combining notions of reliable assessment with notions of convergence yields nine different *standards of success*, such as refutation with certainty, verification in the limit, and gradual refutation.

On the empirical side, decidability with certainty seems to have been the standard of reliable success assumed in skeptical arguments from classical times up through Hume. Verification with certainty was briefly popular among the logical positivists, who insisted that cognitively significant theories be in principle verifiable. Karl Popper (1968) held that the scientific enterprise is characterized by its resolution to reject refuted theories with certainty rather than to protect them with conventionalistic stratagems. C.S. Peirce proposed an ideal of convergent success (1958) and Hans Reichenbach's vindication of the straight rule of induction (1949) was closely related to gradual verification.

On the computational side, numerical properties that are decidable with certainty are said to be *recursive*, those that are verifiable with certainty are said to be *recursively enumerable* or *r.e.*, and those that are refutable with certainty are said to be *co-r.e.* But the application of success standards to formal problems does not stop here. Hilary Putnam (1965) referred to properties formally decidable in the limit as *trial and error predicates*. E. Mark Gold (1965) referred to properties formally verifiable in the limit as *limiting r.e.* and to those that are refutable in the limit as *limiting co-r.e.* Finally, Peter Hàjek (1978) has examined formal problems that are gradually verifiable under the rubric of *experimental logics*. Putnam, Gold, and Hàjek clearly saw the analogy between their criteria of formal success and intuitions about convergence in empirical inquiry.[8]

There is a further analogy. In empirical applications, the scientist may start out with assumptions restricting the set of possible data streams he might encounter to some set $\mathcal{K}$. This has the effect of making the problem easier to solve, since there are fewer possibilities to contend with. Similarly, a formal decision problem can become easier if we restrict the range of natural numbers

[8] So did Putnam's student, Peter Kugel (1977).

over which the machine is required to succeed, as when the halting problem is restricted to indices that either fail to halt on an input or that halt under a fixed complexity bound.[9]

4. THE HALTING PROBLEM AND INDUCTIVE GENERALIZATION

Sextus Empiricus recorded the following, classical argument for inductive skepticism.

> [The dogmatists] claim that the universal is established from the particulars by means of induction. If this is so, they will effect it by reviewing either all the particulars or only some of them. But if they review only some, their induction will be unreliable, since it is possible that some of the particulars omitted in the induction may contradict the universal. If, on the other hand, their review is to include all the particulars, theirs will be an impossible task .... (Sextus, 1985, p.105)

From the logical reliabilist point of view, Sextus' argument is actually a sketch for a proof of an elementary mathematical proposition. Suppose the hypothesis in question contains exactly one data stream ε. For example, the universal empirical hypothesis "You will always see 0" is a singleton containing the everywhere 0 data stream. Let α be an arbitrary empirical assessment method. A wily demon can adopt a strategy for presenting data to α in such a way that $\mathcal{H}$ is false but α becomes certain that $\mathcal{H}$ is true. The demon simply presents all 0s until such time as α declares 1 with certainty. Thereafter the demon presents all 1s. If α never says 1 with certainty, α fails on the everywhere 0 data stream. If α does, then α fails on a data stream that has all 0s prior to announcing certainty and all 1s thereafter. In fact, the same argument works for any non-open hypothesis (in the standard, infinite product topology on ω). A singleton is just one example of a non-open set.

In the formal domain, the most basic negative result is that the complement of the halting problem (the set of all natural numbers n such that the nth Turing machine does not halt on input n) is not r.e. (i.e., not formally verifiable with certainty). It certainly seems as though the difficulty is the same as in Sextus' skeptical argument: as soon as the would-be verification procedure becomes sure that a machine will never halt on its own index, the machine may halt at some later time (for all the verification procedure knows in light of its bounded perspective on the simulated computation). But despite the strong heuristic analogy, the usual proof of the fact in question bears little resemblance to Sextus' skeptical argument. Turing machines do not accept infinite inputs, so there is no place for a demon to feed misleading data to. Instead, we are presented with a static, Cantorian diagonal argument in which the complement of the halting problem is depicted as the counterdiagonal of an infinite table in which each characteristic function of an r.e. set is a

[9] Cf. exercise 2-34 in (Rogers, 1987).

row. This is elegant, but it gives the beginning student the impression that epistemological analogies are ultimately misleading.

They aren't. Let M_i be an arbitrary Turing machine (that we may think of as aspiring to verify the complement of the halting problem with certainty). Now we construct a "demonic machine" M_d dedicated to misleading M_i about whether it will halt. On a given input, it throws away the input and then passes along the result of simulating the aspiring positive test M_i on input d, where d is the demonic machine's own index. (That a program that feeds its own index to another program exists is an immediate consequence of the Kleene fixed point theorem.) Now we have a situation just like that in Sextus' argument: if the aspiring positive test ever declares certainty (by halting), the demonic program is total, so the aspirant is mistaken; and if the aspiring test never declares its certainty by halting, the demonic program halts on no input, so again the aspirant is mistaken. Just like the inductive demon, the demonic machine out-waits the would-be verification procedure.[10]

5. HIERARCHIES AND REDUCIBILITY

Impose the discrete topology on ω. Let $\mathcal{N}$ denote the set of all total functions on ω with the usual product topology induced by the discrete topology on ω. $\mathcal{N}$ is referred to as the Baire space. Let $\mathcal{R}$ be a relation involving n function arguments and m number arguments. $\mathcal{R}$ is open just in case it is open in the product space $\mathcal{N}^n \times \omega^m$. $\mathcal{R}$ is r.e. just in case there is a Turing machine such that for each input tuple in $\mathcal{N}^n \times \omega^m$, the machine eventually halts on the tuple just in case $\mathcal{R}$ holds of the tuple. In this definition, the functions are input to the machine on infinite read-only tapes, only finitely much of which can be read prior to halting.

The arithmetical complexity classes can now be defined as follows.[11] Let Σ^0_0 denote the collection of recursive relations. Then a Σ^0_{n+1} relation is a relation that can be expressed as $\exists y.\neg\mathcal{R}(\bar{\varepsilon}, \bar{x}, y)$, where $\mathcal{R} \in \Sigma^0_n$. A relation is Π^0_n just in case its negation is Σ^0_n. Finally, a relation is Δ^0_n just in case it is both Σ^0_n and Π^0_n. The Borel complexity classes of finite order are defined in exactly the same way, except that the bold-face Greek characters $\mathbf{\Sigma}^0_n$, $\mathbf{\Pi}^0_n$, and $\mathbf{\Delta}^0_n$ are used and the recursive relations are replaced with the clopen relations in the base case.[12]

The Borel hierarchy was studied at the turn of the century by analysts interested in whittling down the space of all set theoretic functions to some

[10] Similar arguments can be given for a wide range of negative results. For demonic arguments against limiting computation cf. (Kelly, 1995). This inductive representation of computational intractability is common in learning-theoretic arguments (Case and Smith, 1983).

[11] Cf. (Rogers, 1987),(Moschavakis, 1980), or (Hinman, 1978).

[12] In the Baire space, every open set is a countable union of clopen sets. In spaces lacking this property (e.g., the real line), one must initiate the induction with the open sets, finally defining $\mathbf{\Sigma}^0_0$ to be $\mathbf{\Delta}^0_1$.

Sense of Success	*Maximum Problem Complexity*	
	Empirical case	*Formal case*
decidable with certainty	$\mathbf{\Delta}_1^0$	Δ_1^0
verifiable with certainty	$\mathbf{\Sigma}_1^0$	Σ_1^0
refutable with certainty	$\mathbf{\Pi}_1^0$	Π_1^0
decidable in the limit	$\mathbf{\Delta}_2^0$	Δ_2^0
verifiable in the limit	$\mathbf{\Sigma}_2^0$	Σ_2^0
refutable in the limit	$\mathbf{\Pi}_2^0$	Π_2^0
decidable gradually	$\mathbf{\Delta}_2^0$	Δ_2^0
verifiable gradually	$\mathbf{\Pi}_3^0$	Π_3^0
refutable gradually	$\mathbf{\Sigma}_3^0$	Σ_3^0

TABLE 1.1. A characterization of the solvable empirical and formal problems

large class with nice analytical properties.[13] In the 1940's, S. Kleene developed the arithmetical hierarchy in order to study the new concept of computability that had evolved in response to Gödel's incompleteness theorems, apparently without knowledge of the earlier developments in analysis. J. W. Addison (1955) recognized the strong mathematical analogies between the two hierarchies: continunity is like computability, open sets are like r.e. sets, clopen sets are like recursive sets, the closure laws are similar, and the initial countable segment of the Borel hierarchy can be defined just like the arithmetical hierarchy except that the r.e. relations are replaced with open relations. Addison's (Σ, Π, Δ) notation for the hierarchies underscores these analogies.

The relevance of all this for our purposes is that the arithmetical hierarchy exactly characterizes the various senses of *formal* problem solvability defined above, and the corresponding cells in the Borel hierarchy exactly characterize the corresponding senses of *empirical* problem solvability.

Proposition 1 [14] *Table 1.1 matches each sense of problem solvability defined above with the complexity classification that characterizes the problems solvable in that sense, both for empirical problems (empirical hypotheses) and for formal problems (sets of numbers).*

In other words, the arithmetical hierarchy is a classification of formal problems according to the sense in which they are computably solvable and the

[13] For a summary of these issues, cf. (Moschavakis, 1980).

[14] The computational results summarized in this proposition are in (Putnam, 1965), (Gold, 1965) and (Hàjek, 1978). For an expanded presentation, cf. (Kelly, 1995).

Borel hierarchy is a parallel classification of hypothesis assessment problems according to the sense in which they are empirically solvable. That uncomputability and inductive skepticism should generate such similar mathematical structures provides a further analogy between Church's thesis and Hume's problem.

The concept of computable many-one reducibility is basic to the theory of computability. Generally speaking, a many-one reduction of a set of functions $\mathcal{A}$ to a set of functions $\mathcal{B}$ is an operator that takes elements of $\mathcal{A}$ to elements of $\mathcal{B}$ and nonelements of $\mathcal{A}$ to nonelements of $\mathcal{B}$. Here again there is an analogy. We may think of a reliable inductive assessment method as a continuous many-one reduction among sets of functions (i.e., of the hypothesis to the set of all conjecture streams that converge in the relevant sense[15]). The results summarized in the second column of proposition 1 therefore amount to continuous reducibility completeness theorems for the respective convergence criteria in their corresponding Borel complexity classes. An inductive demon continuously reduces the convergence criterion to the complement of the hypothesis. Since the convergence criterion is complete in the corresponding Borel complexity class, the existence of both a demon and a successful assessment method for a given hypothesis would imply collapse of the Borel hierarchy. So skeptical arguments fit naturally into the computability theorist's standard repertoire of moves.[16]

6. THE RICE-SHAPIRO THEOREM

The Rice-Shapiro theorem is a standard tool for proving that an index set (i.e., the set of all indices of some collection of partial computable functions) is not r.e. From the present point of view, however, it illustrates a strong connection between computability and ideal[17] empirical inquiry.

Assume a fixed, computable encoding of pairs of numbers into nonzero numbers. Let ϕ be a partial computable function. A data stream for ϕ is an enumeration of code numbers of ordered pairs in the graph of ϕ, possibly interspersed with 0s. (The 0s ensure that the everywhere undefined function will have at least one data stream, namely, the everywhere 0 data stream). Say that the empirical hypothesis generated by an index set is the set of all data streams of functions with indices in the set. Then the Rice-Shapiro theorem[18] says:

Proposition 2 (Rice-Shapiro) *If an index set is formally verifiable with certainty, then the hypothesis generated by the index set is empirically ver-*

[15]Assume an encoding of rationals in the unit interval by natural numbers.

[16]Continuous reducibility is a standard topic in descriptive set theory; cf. (Moschavakis, 1980).

[17]By 'ideal' we mean subject to no computational limitations.

[18](Cutland, 1986, Theorem 2.16, p.130)

ifiable with certainty given that the data stream is for a partial computable function.[19]

Proposition 2 permits one to prove a purely computational result (that a set of numbers is not r.e.) by means of a purely empirical argument (a skeptical, demonic argument showing that a certain empirical hypothesis is not verifiable with certainty from empirical data).

7. COMPUTABILITY, EMPIRICAL INQUIRY AND COMPUTABLE EMPIRICAL INQUIRY

We began with a common view among inductive methodologists, according to which formal problems are relatively unproblematic and computational limitations are either ignored or are deferred until after some ideal norms governing inductive inference have been framed. In response, we argued that Hume's problem and Church's thesis share a common motivation, namely locality constraints on perception; whether on an empirical data stream or an internal data structure. It was also argued that virtually the same notions of convergence and reliability apply both in the formal and in the empirical domain, which gives rise to similar complexity hierarchies in the two subjects.

But there is a deeper sense in which computation and empirical inquiry are analogous. Say that a problem is *underdetermined* if there is no reliable method for solving it. *Inductive* underdetermination arises due to a finitely bounded perspective on an unbounded structure. It is immediate that classical sceptical arguments are about inductive underdetermination. What we have seen in our discussions of the Rice-Shapiro theorem and the halting problem, however, is that uncomputability arguments are as well.

All inductive methodologies are responses to inductive underdetermination. If inductive methodologists did not take inductive underdetermination seriously, then they would simply insist that everyone always believe the complete truth. But we have seen that uncomputability also stems from inductive underdetermination arising from similar locality and boundedness assumptions. It is therefore an oddly imbalanced position to ignore computability as an inessential accident in inductive methodology when one's primary focus hinges on the same underlying structure. Such a position resembles that of a physicist interested in pressure who disregards changes in temperature as annoying accidents when in fact the same underlying structure—the kinetic behavior of molecules—gives rise to systematic and essential relations between the two.

Given that uncomputability and inductive skepticism both stem from inductive underdetermination and that computability considerations should therefore not be ignored in the study of inductive methodology, there remains the question of how the two factors should be analyzed. The usual theory of com-

[19] Cf. (Kelly, 1995, Corollary 6.8).

putability requires procedures to eventually produce the right answer. The standard approach to induction is very different, exchanging guaranteed convergence to the truth for other considerations such as probability, confirmation, coherence, simplicity, or social pressure. These distinct conceptualizations of induction and computability do not fit together well. Probabilistic coherence requires logical omniscience, but logical ignorance may be unavoidable when logical relations are uncomputable. And even for effectively solvable problems, logical ignorance persists until the algorithm's output is obtained. Probabilists can say that such constraints are not to be taken literally, but this strategy must stop somewhere because probabilistic coherence is nothing without logical omniscience over some algebra.

If we adopt the logical reliabilist approach, however, then we obtain a uniform perspective on both the inductive and the formal aspects of computable inductive inquiry; a perspective in which the structures of computation and induction interleave without artificial obstacles, as the following proposition illustrates.

Proposition 3 *The characterization of ideal hypothesis assessment presented in proposition 1 holds in the case of hypothesis assessment by a computable method if we substitute arithmetical complexity classes for the corresponding Borel complexity classes.*

The proposition says that the arithmetical hierarchy incorporates both the purely inductive and the purely formal aspects of computable inductive inquiry and that it does so in a manner entirely parallel to the Borel characterization of ideal (uncomputable) inductive inquiry. Arithmetical complexity can be high because a hypothesis is inductively underdetermined by the empirical data, because it forces the scientist to perform impossible computations in order to check it against the data, or because of some complicated combination of the two factors. Moreover, a given amount of formal complexity may either add to the purely empirical complexity of the hypothesis or interleave with it so that the overall arithmetical complexity is not increased, depending on the particular case. Thus the analogy with pressure and temperature is apt: there is an underlying, fundamental structure (arithmetical complexity) against which the interactions between uncomputability and the problem of induction are essential dependencies rather than accidents to be dismissed. In the following sections we provide a systematic study of relations of this sort with some surprising consequences.

8. COMPUTABLE INQUIRY AND UNCOMPUTABLE PREDICTIONS

The traditional distinction between matters of fact and relations of ideas invites the inductive methodologist to separate formal questions from empirical ones. One of the most obvious formal problems facing an empirical scientist is to derive predictions from the hypothesis under test for comparison with the data. Recall that in our present setting, an empirical hypothesis is a set of data streams $\mathcal{H}$. We say that $\mathcal{H}$ predicts observation b at stage n given finite data sequence e just in case each data stream in $\mathcal{H}$ that extends e has b in position n. Then we write $Pred_{\mathcal{H}}(e, n, b)$. As a special case, say that a hypothesis is *empirically complete* just in case it unconditionally predicts a unique datum for each position in the data stream. An empirically complete hypothesis therefore contains a single data stream (namely, the one agreeing with all of the unconditional predictions). Finally, the range of a hypothesis $\mathcal{H}$ is the set of outcomes that may ever arise if $\mathcal{H}$ is true.

The following is perhaps the simplest and most obvious methodological advice imaginable: *To test a hypothesis, derive its predictions, compare them against the data, and reject the theory when a mismatch is found.*

The advice is correct when the hypothesis is empirically complete and there is a computable procedure for deriving the predictions of $\mathcal{H}$ (i.e., when $Pred_{\mathcal{H}}$ is recursively enumerable), for then the advice amounts to a computable method that refutes $\mathcal{H}$ with certainty.

But what happens when the predictions of the hypothesis under test are not computably derivable? Then it seems as though no computable method could refute the hypothesis with certainty. If the predictions are not computable, then every computable method for deriving them must either be unsound (it mistakenly derives the wrong prediction) or incomplete (it hangs in an infinite loop, never yielding the intended prediction). In the former case, the data might conspire to agree with the mistakenly derived prediction so that the hypothesis is false but the method never notices. In the latter case, the method never finds out what the theory predicts about some datum, and the truth of the theory may depend entirely on what this prediction is.

But remarkably enough, the argument just given is fallacious: there are empirically complete hypotheses that can be reliably assessed by computable methods even though their predictions are not even definable in arithmetic.[20]

Proposition 4 *There is a hypothesis $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is empirically complete*
2. *$Pred_{\mathcal{H}}$ is not arithmetically definable*
3. *but $\mathcal{H}$ is computably refutable with certainty.*

Due to the following proposition, the hypothesis that witnesses proposition 4 has an infinite range of predictions.

Proposition 5 *Suppose that $\mathcal{H}$ is empirically complete, computably verifiable in the limit, and of finite range. Then $Pred_{\mathcal{H}}$ is computably decidable with certainty.*

On the other hand:

[20]Complete proofs of the following propositions may be found in (Kelly and Schulte, 1995).

Proposition 6 *There is a hypothesis $\mathcal{H}$ such that*

1. *$\mathcal{H}$ is empirically complete*
2. *the range of $\mathcal{H}$ is binary,*
3. *$Pred_{\mathcal{H}}$ is not arithmetically definable,*
4. *but $\mathcal{H}$ is computably refutable in the limit.*

No empirically complete hypothesis is open, so by proposition 1, no such hypothesis is verifiable with certainty. Thus we quickly arrive at a complete picture of how uncomputable the predictions of an empirically complete theory can be given that the truth of the theory can be reliably determined by computational means in a specified sense.

In light of the view that science should proceed by deriving predictions and checking them against empirical data, propositions 4 and 6 appear almost magical. Somehow or other, computable inquiry can proceed without being able to derive the predictions of the hypothesis, even though any one of these predictions might be wrong. But it underscores the analogy between computability and inductive methodology that these striking methodological facts are actually restatements of familiar results in the theory of computability. The key to seeing this is that the arithmetical complexity of $\{\varepsilon\}$ corresponds to the number of quantifier alternations in an implicit definition of ε in arithmetic and the arithmetical complexity of ε corresponds to the number of quantifier alternations in an explicit definition of ε in arithmetic. Let ν denote the characteristic function of the Gödel numbered truths of arithmetic. The fact that $\{\nu\} \in \Pi^0_2$ is just the implicit definability of arithmetical truth, which was already shown by Hilbert and Bernays. The fact that ν is not arithmetically definable is just Tarski's celebrated theorem on the undefinability of arithmetical truth. Proposition 6 follows immediately by proposition 3.

The proof of proposition 4 builds on the preceding argument (Hinman, 1978). Let $\langle \_, \_ \rangle$ denote an effective encoding of pairs of natural numbers as natural numbers and let $(\langle n, m \rangle)_1 = n$ and let $(\langle n, m \rangle)_2 = m$. Extend the coding to functions so that $(\varepsilon)_1(n) = (\varepsilon(n))_1$. Since $\{\nu\} \in \Pi^0_2$, there is a recursive relation $\mathcal{G}$ such that for each ε,

$$\varepsilon \in \{\nu\} \iff \forall x \exists y . \mathcal{G}(\varepsilon, x, y).$$

Then define

$$\delta(x) = \langle \nu(x), \mu y . \mathcal{G}(\nu, x, y) \rangle .$$

Hence, $\nu(x) = (\delta(x))_1$, so δ is not arithmetically definable. Finally, $\{\delta\}$ is definable as follows.

$$\varepsilon \in \{\delta\} \iff \forall x . \mathcal{G}((\varepsilon)_1, x, (\varepsilon(x))_2) \text{ and } \forall x, y \text{ if } y < (\varepsilon(x))_2 \text{ then } \neg \mathcal{G}((\varepsilon)_1, x, y).$$

So $\{\delta\} \in \Pi^0_1$. Note that while the range of ν is binary, the minimization in the definition of δ makes the range of δ infinite.

Results relating implicit to explicit definability are standardly grouped together with the recursion theoretic basis theorems. A *basis* for a collection Γ of sets of functions (i.e., empirical hypotheses) is a collection of functions that shares an element with each nonempty set in Γ. Thus, proposition 6 says that the set of all arithmetically definable functions is not a basis for the set of Π^0_2 singletons of binary range, proposition 4 says that the set of all arithmetically definable functions is not a basis for the set of Π^0_1 singletons, and proposition 5 says that the set of all total computable functions is a basis for the Σ^0_2 singletons of finite range.

But the basis theorems are relevant to our topic only in the case of empirically complete hypotheses. When we move to empirically incomplete hypotheses, basis theorems say that each hypothesis of a given complexity is correct of a sufficiently simple data stream. But that is not our concern. We are interested in seeing how complex the predictions of a computably testable hypothesis can possibly be, which is something quite different. The empirical application therefore suggests a different generalization of the implicit definability issue than the one traditionally examined in the theory of computability.

In the empirically incomplete case, we can find hypotheses that are computably refutable with certainty whose predictions are even more complex than in the preceding examples. The *analytical* hierarchy is defined just like the arithmetical hierarchy, except that existential quantification over numbers is replaced with existential quantification over functions. The analytical hierarchy is indexed with superscript 1 to distinguish it from the arithmetical hierarchy. The relations in Δ^1_1 are said to be *hyperarithmetical.* A relation is *complete* in a complexity class just in case it is a member of the class and every member of the class is computably many-one reducible to it.

Proposition 7 *There is a hypothesis $\mathcal{H}$ such that*

1. *$Pred_{\mathcal{H}}$ is Π^1_1-complete*
2. *and $\mathcal{H}$ is computably refutable with certainty.*

In contrast, $\{\nu\}$ and $\{\delta\}$ are both Δ^1_1 and could not have been chosen outside of Δ^1_1. The construction of $\mathcal{H}$ is as follows. Let S be a Π^1_1-complete set of numbers. Let $\langle\_\rangle$ effectively encode finite sequences of natural numbers into natural numbers. By a standard normal form theorem (Rogers, 1987, Ch.16,Cor.V), there is a recursive G such that for all x:

$$S(x) \iff \forall\tau\exists n.G(\langle\tau|n\rangle\,, x)$$

where $\tau|n$ denotes the initial segment of τ of length n. Our hypothesis is then defined as follows:

$\varepsilon \in \mathcal{H} \iff$

1. $\varepsilon(2)$ encodes a unit sequence,
2. $\forall i \geq 2, \varepsilon(i+1)$ encodes an extension of length i of the sequence encoded by $\varepsilon(i)$,

Given sense in which $\mathcal{H}$ is empirically testable		Maximum complexity of $Pred_{\mathcal{H}}$			
		Empirically complete $\mathcal{H}$		General Case	
		Finite Range	Infinite Range	Finite Range	Infinite Range
decidable with certainty:	Δ^0_1	$\square$	$\square$	Δ^0_1	Π^0_1
verifiable with certainty:	Σ^0_1	$\square$	$\square$	Π^0_1	Π^0_1
refutable with certainty:	Π^0_1	Δ^0_1	Δ^1_1	Σ^0_1	Π^1_1
decidable in the limit:	Δ^0_2	Δ^0_1	Δ^1_1	Π^0_2	Π^1_1
verifiable in the limit:	Σ^0_2	Δ^0_1	Δ^1_1	Π^0_2	Π^1_1
refutable in the limit:	Π^0_2	Δ^1_1	Δ^1_1	Π^1_1	Π^1_1
	Σ^0_{n+3}	Δ^1_1	Δ^1_1	Π^1_1	Π^1_1

TABLE 1.2. The inductive and deductive complexity of empirical hypotheses.

3. and $\forall i \geq 2, \neg G(\varepsilon(i), \varepsilon(0))$.

By the form of the definition, $\mathcal{H} \in \Pi^0_1$. It turns out that each arithmetically definable hypothesis has Π^1_1 predictions. It remains to see that $Pred_{\mathcal{H}}$ reduces S. Suppose $x \in S$. Then $\forall \tau \exists n. G(\langle \tau | n \rangle, x)$. Hence, no data stream in $\mathcal{H}$ begins with x. So the unit data sequence (x) refutes $\mathcal{H}$ and $\mathcal{H}$ trivially predicts every possible outcome at position 1 given that (x) has been observed. On the other hand, suppose $x \notin S$. Then $\exists \tau \forall n. \neg G(\langle \tau | n \rangle, x)$. Hence, (x) is consistent with $\mathcal{H}$. Since $\mathcal{H}$ imposes no constraints on what happens at stage 1, any outcome may arise at stage 1 given only that (x) is observed. Therefore, $x \in S \iff Pred_{\mathcal{H}}((x), 1, 0)$.

Similar arguments lead to a complete table mapping out how intractable the predictions of a computably testable theory can be for each notion of inductive success (Table 1.2). The results differ sharply depending on whether or not the hypothesis is empirically complete and on whether or not the range of the hypothesis is finite. The $\square$ symbol indicates that the case in question is impossible.

9. IDEAL NORMS AND COMPUTATIONAL DISASTERS

So what, intuitively, is the magic that enables computable science to reliably test theories with uncomputable predictions? Say that a method that rejects a hypothesis with certainty as soon as it is refuted by the data is *consistent*. Consistency is recommended by almost all ideal methodologists, including hypothetico-deductivists, Bayesians and, more recently, belief revision theorists (Gärdenfors, 1988). But it turns out that:

Proposition 8 [21] *Suppose that $\mathcal{H}$ is assessable (in any of the senses defined above) by a* consistent *method α. Then we have the following consequences.*

1. *If α is in Σ^0_n, then $Pred_{\mathcal{H}} \in \Pi^0_{n+1}$.*
2. *If $\mathcal{H}$ is also empirically complete, then $Pred_{\mathcal{H}} \in \Sigma^0_n$.*
3. *If α is analytical, then $Pred_{\mathcal{H}}$ is of analytic complexity no greater than α's.*

It follows that hypotheses whose predictions are not arithmetically definable cannot be assessed in any of our senses by an arithmetically definable consistent method. Since the predictions of the hypothesis $\mathcal{H}$ exhibited in proposition 7 are not even hyperarithmetical, no consistent, hyperarithmetically definable method could even gradually refute or verify $\mathcal{H}$; a situation that contrasts sharply with the fact that $\mathcal{H}$ is refutable *with certainty* by a nonconsistent, *computable* method. Consistency imposes a severe restriction on the power of computable science.

Nonconsistent methods allow some time lag before noticing that the hypothesis under test has been refuted. This might suggest that the formal method deriving the predictions merely needs some extra time to do its work while further data are being read. But there is more to it than that, since nonarithmetical predictions cannot be derived by computable means no matter how much time delay is provided, or even in the limit. In each of our examples, the successful computable method succeeds by actually *using* future empirical data to detect the inconsistency of the hypothesis under test with earlier data. Let us return to the hypothesis $\mathcal{H}$ defined in the proof of proposition 7. To refute $\mathcal{H}$ with certainty, one begins by reading and saving the datum x in position 0. Then one ignores the datum in position 1 and checks whether the datum i in position 2 is the code number of a unit sequence and whether $\neg G(i, x)$. Thereafter, one checks whether the current datum j encodes a unit extension of the sequence encoded by the preceding datum and whether $\neg G(j, x)$. The important point is this: if the first datum encountered is in S, then $\mathcal{H}$ is already refuted by this datum, but our procedure won't notice that until some later datum is read and decoded. Moreover, it is clear that no hyperarithmetically definable formal method that reads only the first datum could ever determine, without looking at future data, whether this first datum already refutes $\mathcal{H}$. The same may be said of any arithmetically definable method that succeeds in any of our senses when faced with the hypotheses $\{\nu\}$ and $\{\delta\}$.

We have just seen that purely formal problems arising in the course of computable empirical inquiry may require future empirical data for their so-

[21](Gaifman and Snir, 1990) contains a related result concerning the definability of Bayesian probability measures that maintain unit priors on aritmetical truths and nonextremal priors on contingent sentences in a first order language including arithmetic.

lution, so that the traditional distinction between the formal and inductive components of inquiry is blurred. This situation stands in sharp contrast with Hume's distinction between empirical matters of fact and a priori relations of ideas—the distinction with which this paper began.

10. CONCLUSION

The paper began with the traditional view that formal problems are to be bracketed or deferred in fundamental discussions on the logic of induction. In response, we have argued that Church's thesis is analogous to Hume's problem. Both are founded on material assumptions concerning bounded perception. Both arise out of similar reliabilist criteria of success. Both generate similar hierarchies of complexity. Both give rise to natural notions of reducibility and completetness. And most importantly, arguments for both uncomputability and inductive skepticism share a common structure; the structure of inductive underdetermination. Therefore it is unnatural for inductive methodology to abstract from computability considerations. Logical reliabilism provides a balanced perspective on uncomputability and inductive skepticism in the study of computable empirical inquiry.

As an illustration of this approach, we analyzed what seems to be the most elementary methodological recommendation conceivable: that to test a theory one should formally derive its predictions and then check them against the data. A striking result of this investigation is that for some hypotheses a computable inductive method must use future empirical data to determine the consistency of past data with the hypothesis, thereby further breaking down the traditional distinction between empirical and formal methodology. Another is that for empirically complete theories, inductive and deductive complexity are literally related as implicit and explicit definability in arithmetic. Still another is that inductive methodology suggests a natural generalization of implicit and explicit definability that differs from the standard basis theorems.

REFERENCES

Addison, J. (1955). "Analogies in the Borel, Lusin, and Kleene Hierarchies," *Bulletin of the American Mathematics Society* 61: 171-172.

Angluin, D. and Smith, C. "A survey of inductive inference: Theory and methods." *Computing Surveys* 15:237–289.

Case, J. and Smith, C. (1983)"Comparison of Identification Criteria for Machine Inductive Inference," *Theoretical Computer Science* 25: 193–220.

Cutland, N.J. (1986). *Computability: An introduction to recursive function theory.* Cambridge: Cambridge University Press.

Gaifman, H. and M. Snir (1982). "Probabilities Over Rich Languages, Testing and Randomness," *Journal of Symbolic Logic* 47: 495– 548.

Gärdenfors, P. (1988). *Knowledge In Flux: modeling the dynamics of epistemic states.* Cambridge: MIT Press.

Garber, D. (1983) "Old Evidence and Logical Omniscience in Bayesian Confirmation Theory," in *Testing Scientific Theories: Minnesota Studies in the Philosophy of Science*, vol. 10, ed. J. Earman. Minneapolis: University of Minnesota Press.

Gold, E.M. (1965). "Limiting Recursion," *Journal of Symbolic Logic* 30: 27–48.

Hàjek, P. (1978). "Experimental Logics and Π^0_3 Theories," *Journal of Symbolic Logic* 42: 515–522.

Hinman, P.G. (1978). *Recursion-Theoretic Hierarchies.* New York: Prentice-Hall.

Hume, D. (1984). *An Inquiry Concerning Human Understanding*, ed. C. Hendell. New York: Bobbs-Merrill.

James, W. (1948) "The Will To Believe," *Essays in Pragmatism.* ed. A. Castell. New York: Collier.

Kelly, K. (1995). *The Logic of Reliable Inquiry.* Oxford: Oxford University Press.

Kelly, K. and Schulte, O. (1995) "The Computable Testability of Theories Making Uncomputable Predictions," *Erkenntnis.* 43:29–66.

Kugel, P. (1977). "Induction, Pure and Simple," *Information and Control* 33:276–336.

Levi, I. (1990). "Rationality Unbound," in *Acting and Reflecting,* ed. W. Sieg. Dordrecht: Kluwer.

Moschavakis, Y. (1980). *Descriptive Set Theory.* Amsterdam: North Holland.

Osherson, D., Stob, M. and Weinstein, S (1986). *Systems That Learn.* Cambridge, Mass: MIT Press.

Peirce, C.S. (1958). *Charles S. Peirce: Selected Writings*, ed. P. Wiener. New York: Dover.

Popper, K. (1968). *The Logic Of Scientific Discovery.* New York: Harper.

Putnam, H. (1963). " 'Degree of Confirmation' and Inductive Logic," In *The Philosophy of Rudolph Carnap*, ed. A. Schilpp. La Sall, Ill.: Open Court.

Putnam, H. (1965). "Trial and Error Predicates and a Solution to a Problem of Mostowski," *Journal of Symbolic Logic* 30: 49–57.

Reichenbach,H.(1949). *The Theory of Probability.* London: Cambridge University Press.

Rogers, H. (1987). *The Theory of Recursive Functions and Effective Computability.* Cambridge: MIT Press.

Sextus Empiricus (1985). *Selections from the Major Writings on Scepticism, Man and God,* ed. P. Hallie, trans. S. Etheridge. Indianapolis: Hackett.

Sieg, W.(1994). "Mechanical Procedures and Mathematical Experience," in *Mathematics and Mind*, ed. A. George. Oxford: Oxford University Press.

Turing, A. (1936). "On Computable Numbers, with an Application to the Entscheidungsproblem," *London Mathematics Society* 42: 230–265.

ACKNOWLEDGEMENTS

Oliver Schulte was supported by a grant from Siemens AG during the course of this research. We are indebted to our colleagues Clark Glymour and Wilfried Sieg for comments on a draft of this paper.

Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA
15213 USA
E-mail: kevin.kelly@andrew.cmu.edu, oliver.schulte@andrew.cmu.edu