Gilboa, Itzhak

Working Paper

Philosophical Applications of Kolmogorov's Complexity Measure

Discussion Paper, No. 923

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Gilboa, Itzhak (1990) : Philosophical Applications of Kolmogorov's Complexity Measure, Discussion Paper, No. 923, Northwestern University, Kellogg School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston, IL

This Version is available at:
https://hdl.handle.net/10419/221282

Discussion Paper No. 923

PHILOSOPHICAL APPLICATIONS OF KOLMOGOROV'S COMPLEXITY MEASURE*

by

Itzhak Gilboa**

October 1990

---

*I would like to thank many teachers, colleagues, friends and family members for numerous discussions which motivated and refined these notes, for the encouragement to write them down, and for comments and references.

**Department of Managerial Economics and Decision Sciences, J.L. Kellogg Graduate School of Management, Northwestern University, Evanston, Illinois 60208.

Abstract

Kolmogorov has defined the complexity of a sequence of bits to be the minimal size of (the description of) a Turing machine which can regenerate the given sequence.

This paper contains two notes on possible applications of this complexity notion to philosophy in general and the philosophy of science in particular. The first presents simplicism--a theory prescribing that people would tend to choose the simplest theory to explain observations, where "simple" is defined by (a version of) Kolmogorov's measure. The second suggests a reinterpretation of a simple observation, saying that reality is almost surely too complex to understand, terms such as "good" and "evil" almost surely too complex to define, and so forth.

"The process of induction is the process of assuming the simplest law that can be made to harmonize with our experience." (Wittgenstein (1922, Proposition 6.363))

1. Simplicism

1.1 Introduction

The basic question of how people choose theories to explain observations has justifiably drawn much attention and received numerous and various possible answers. In particular, it will certainly not be a shockingly new idea to suggest that people opt for the simplest possible theory, i.e., the simplest theory out of those that are compatible with accumulated evidence. (As an explicitly descriptive theory this idea dates back to Wittgenstein (1922) at the latest, while with a slightly more normative flavor it is often attributed to William of Occam--see, e.g., Russel (1945, pp. 468-473), and Sober (1975) for additional references.)

The main message of this note is that some variations on Kolmogorov's definition of complexity may be used to clarify the notion of a "simplest theory" by defining it in terms of somewhat more primitive notions. (For the complexity measure, see Kolmogorov (1963, 1965), Martin-Lof (1966), and Loveland (1969)).

In very bold strokes, simplicism is a descriptive philosophy of science theory, which says that, for the appropriate choice of a "language," people in general (and scientists in particular) tend to prefer a theory which has the shortest description in this language. Simplicism presupposes a model in which a "scientific theory" is represented by a Turing machine (or a computer program) rather than, say, by a set of axioms. Such a model is presented below, and it may be viewed as a(nother) departure from the Received View (Carnap (1923); see also Suppe (1974) for a survey and

references).

An example may be useful to clarify the idea and the problems we will encounter with a formal definition. Suppose the phenomenon one tries to explain is the rise of the sun. Data is gathered, say, over 10,000 days, and each day provides one observation--1 if the sun rose and 0 if it did not. Suppose the 10,000-bit sequence which we observed consists of 10,000 1's. Further assume that the language we work with is the computer language PASCAL. A scientist is called to develop a theory--i.e., a computer program--that, when run, will produce an infinite sequence of bits, the first 10,000 of which are 1's. Examples of such are:

a.
```
While (0 = 0) do
     write (1); {the sun rises every morning}.
```

b.
```
While (0 = 0) do
begin
     for i = 1..10,000 write (1);
     for i = 1..10,000 write (0);
end; {the sun rises for 10,000 mornings, then stops for 10,000
     mornings, and so forth}.
```

c.
```
While (0 = 0) do
begin
     for i = 1..20,000 write (1);
     for i = 1..30,000 write (0);  {the sun rises for 20,000
                                    mornings, then does not rise
                                    for 30,000 mornings, and so
                                    forth}
end;
```

```
d.   Write (1);
     Write (1);
          .
          .
     Write (1): {10,000 times}
     P:   {P is any program that generates an infinite sequence of
          bits.  "The sun rises on the first day; the sun rises on the
          second day;...;the sun rises on the 10,000th day; P occurs}
     end.
```

Obviously, program (theory) (a) is the shortest and, indeed, seems the intuitive choice in this example. The definition of program "length" should probably be sophisticated enough not to count the number of letters in "while," "do," "begin," and so forth, but to be able to distinguish between programs (b) and (c), probably ranking (b) as "shorter" since it involves smaller constants, with the same constant appearing twice so that it may be further shortened by storing it in memory just once. At any rate, it seems quite obvious that the point-by-point theory (d) is intuitively the most cumbersome and technically the longest.

Notice that should (a) be agreed upon as the "natural" choice in this example, one may use simplicism to explain the "Goodman paradox" (Goodman (1965)), i.e., why people tend to classify emeralds as "green" or "blue" rather than "grue" ("green until the year 2000 and blue thereafter") or "bleen" (the converse). That is, assuming a language in which "green" and "blue" are primitives, "always green" and "always blue" are describable by shorter programs than "grue" and "bleen." (See also Sober (1975, pp. 19-23)

for a simplicity-based resolution of Goodman's paradox in the context of logic systems.) Of course, this conclusion would be reversed were "grue" and "bleen" primitives in the language, by which "green" and "blue" had to be defined.

More generally, the assumptions that (in some way) the future is going to resemble the past and that the phenomenon observed is independent of the observation process--are also derivable from simplicism.

1.2 Model

Simplicism relies on the assumption that all that can ever be observed is faithfully described by a (typically infinite) countable sequence of bits. Although this may seem restrictive, one should recall that all the finite questions one can formulate in English are countable, and so are all the possible finite answers. With the usual encoding techniques, then, all choices of answers to all questions may be thought of as mapped to the infinite bit sequences--alternatively, into the real interval [0,1].

The set of states of the world is

$$\Omega = \{\omega \mid \omega: \mathbb{N} \rightarrow \{0,1\}\}.$$

Each $\omega \in \Omega$ should be interpreted as providing answers to all questions. Obviously, when all questions and all possible answers are actually encoded into Ω, not every $\omega \in \Omega$ will have a meaningful and consistent interpretation. Hence one may wish to start out the formal model with some subset of Ω rather than Ω itself; but this point is not crucial to the ensuing discussion.

It will prove convenient to identify ω_i with $\omega(i)$ for $\omega \in \Omega$ and $i \in \mathbb{N}$.

Truth is a particular state of the world, which will henceforth be denoted $x = (x_1, x_2, \ldots)$ (where $x_i \in \{0,1\}$ for $i \in \mathbb{N}$). The set of observations at a given point of time is represented by a finite $O \subseteq \mathbb{N}$, to be interpreted as the set of indices $i \in \mathbb{N}$ such that x_i was observed.

In order to define a language, one may use a universal Turing machine, T_U, and add to it a Turing machine T_L to implement statements in the desired language, L. Thus, PASCAL may be modeled as (T_U, T_P) where T_U is a universal Turing machine and T_P is a PASCAL compiler, translating PASCAL programs to the appropriate input for T_U. In general, we will assume that for every language L, T_L halts for every input.

For simplicity of notation, we shall not distinguish between a finite sequence of bits and the corresponding nonnegative integer, i.e., the integer whose binary expansion is the given sequence.

A Turing machine T which halts for every input with a non-empty output sequence thus induces a function $\tilde{T}: \mathbb{N} \to \mathbb{N}$. When no confusion is likely to arise (i.e., always), "T" will also stand for the function $\tilde{T}$.

Given a language L, a sequence of bits P in L is said to be a theory in the language L if for every $i \in \mathbb{N}$, $T_U(T_L(P), i) \in \{0,1\}$. That is, P should be a program in L such that for every input i the universal Turing machine T_U halts for the input $T_L(P)$--a description of a Turing machine--and the given i, and computes a 1-bit output.

If $\omega \in \Omega$ is such that $T_U(T_L(P), i) = \omega_i$ for all $i \in \mathbb{N}$, P will be said to compute ω in L, and ω is computable by P in L. A state $\omega \in \Omega$ is computable in L if it is computable by P for some P. Two programs, P and P' in L, are said to be L-equivalent if they compute the same $\omega \in \Omega$ in L.

The length of the theory P in L is thus well-defined as the number of bits in P.

Thus, simplicism is defined for a given language L, a truth x, and a set of observations O: it prescribes that one of the shortest P's, which satisfy $T_U(T_L(P),i) = x_i$ for $i \in O$, will be chosen as a theory to explain the observations O.

More formally, one may represent the choice of a scientific theory by a choice function c, whose arguments are the language (T_U,T_L) and a non-empty set of possible theories in L, $\mathcal{P} \subseteq \mathbb{N}$. Thus, $c(T_U,T_L,\mathcal{P}) \subseteq \mathcal{P}$ is the set of preferred theories in $\mathcal{P}$. For a truth x and a set of observations $O \subseteq \mathbb{N}$, let $\mathcal{P}_{(x,O,L)}$ be the set of theories P in L such that $T_U(T_L(P),i) = x_i$ for $i \in O$. A choice function c is simplicistic if for every $(x,O,(T_U,T_L))$, $c(T_U,T_L,\mathcal{P}_{(x,O,L)})$ is exactly the set of shortest programs in L explaining O. Obviously, $c(T_U,T_L,\mathcal{P}_{(x,O,L)})$ is finite for all $(x,O,(T_U,T_L))$ and non-empty if (T_U,T_L) is not too restrictive.

Worthy of note is the fact that the identification of a "theory" with a program does not prevent the former from redefining the language. Indeed, a procedure in a computer program may be viewed as extending the language: the procedure name is a new term, while its "body" is this term's definition. Thus, if the introduction of a new term (such as "gravitation," "subconscience," and so forth) makes the rest of the theory extremely simple, there would be a correspondingly short program which includes this term as a procedure, and invokes this procedure in several different statements (while recursion is not precluded).

It is important, however, that the length of the new definitions is part of the program's length, and that the "base" language is a given one,

L. That is, a theory will not be considered "simple" in (with respect to) L if it is simple in L', but L' cannot be easily translated to L.

1.3 The Role of Language

The previous discussion points out the crucial role that the ("programming language") L plays in the choice of the "simplest" theory. Indeed, it may be the case that in one language, say, PASCAL, the orbits of celestial objects turn out to be simpler should the sun, rather than earth, be assumed to define the origin of the solar system, while in another language the converse is true. Moreover, any theory P one may develop can, according to this model, be incorporated into any other language as a "primitive" statement. Hence, any theory is (one of) the simplest in an appropriately chosen language. (Viewed thus, simplicism solves the arbitrariness of the notion of "simplicity" by shifting it one level up (or down), defining simplicity via language, which is, in turn, arbitrarily chosen.) Furthermore, given a finite sequence of theories $\{P_t\}_{t=1}^{T}$, to be thought of as the theories chosen for a corresponding sequence of sets of observations $\{O_t\}_{t=1}^{T}$ (where $O_t \subset O_{t+1}$), there is a language L in which the equivalents of $\{P_t\}_{t=1}^{T}$ are exactly the shortest T programs (in ascending order).

Since there seems to be no theoretically compelling reason to prefer one language to another, simplicism seems a vacuous theory: whatever the choice people make, it may be justified as choosing the simplest theory in the "right" language. However, it is the author's belief that people tend to agree on the primitives of a "natural" language to a high enough degree in order to make simplicism nontrivial. It may well be the case that the

apparently (relatively) common notion of language (and of simplicity itself) heavily depends on a specific culture. Yet, inasmuch as people in a certain culture share the basic language, simplicism predicts they will share their intuition regarding the theory they prefer. (For some qualifications, see subsections 1.4 and 1.5 below.)

Notice that another crucial role of language (which is hidden in the model presented above) is the description of the actual and possible observations. For instance, the very order in which answers to questions are encoded into a sequence of bits may affect the complexity of various theories. It seems, though, that the same argument, namely, the fortuitous universality of language, saves simplicism from being tautologically true. Some comments are still needed, however, to make it at least occasionally true. (See subsection 1.8 below for further discussion of representation and language.)

1.4 Simplicity and Generality

It has been so far assumed that any candidate theory should be defined for the whole domain $\mathbb{N}$. A nice assumption though this is, it is hardly realistic. Indeed, it seems that a much more sensible model would allow each theory P to have a domain $D_P \subseteq \mathbb{N}$ such that, for $i \in D_P$, P computes a "0" or "1" output and, for $i \notin D_P$, P's computation halts with a "no answer" output. (Note that it is required that the theory P will "know" where it is applicable.) D_P will hopefully have a nonempty intersection with the set of observations O and with its complement, $O^C = \mathbb{N} \setminus O$. (When $D_P \cap O \neq \emptyset$ there are at least some predictions of P which may be compared with evidence, i.e., P is falsifiable with respect to O. When $D_P \cap O^C \neq \emptyset$ there are at

least some unresolved questions P has some answer to offer.)

The generality of a theory (program) P may be simply defined as the domain D_P, and the relation "more general" can correspondingly be identified with set inclusion.

With this framework it seems quite obvious that generality and simplicity are two criteria by which theories may be ranked, and that the two may sometimes agree and sometimes not. For instance, the theory "if A then C" is both simpler and more general than "if A and B, then C"; hence, both maximization of generality and simplicism may be evoked to explain why we tend to prefer as few axioms as possible for the explanation of given observations.

Yet for this reason precisely, both criteria prod us to choose one of them rather than let them coexist. Fortunately, the two theories are not observationally equivalent, and it is not difficult to think of examples in which the more general theory is more complex.

Thus, simplicism predicts that "if A then C" will be preferred to "if A or B, then C," assuming the two fit the evidence just as well (see also subsection 1.6 below).

This view seems to be in complete disagreement with Popper (1934, Ch. VII), who equates simplicity with degree of falsifiability. In our model, the "degree of falsifiability" is just a theory's generality, i.e., the domain D_P. As the example above shows, the "simpler" theory ("If A then C") is less general, hence less falsifiable, than the more complex one ("If A or B then C"). Furthermore, this example attempts to convince the reader that, at least in some cases, when the two criteria are not in agreement, it is simplicity which tends to be intuitively preferred.

While Popper strives for "logical or epistemological advantages" of simplicity, such as provided by a higher degree of falsifiability, this paper does not attempt to provide any normative arguments for simplicity or simplicism. Nor does it try to explain why simplicity-seeking behavior is efficient or evolutionary optimal. Its only goal is to present simplicism as a descriptive theory, assuming length-of-program minimization as an axiom.

It should be noted, however, that even if generality is preferred to simplicity, its maximization can hardly replace complexity minimization in our model: whenever two theories, P and P', have the same domain ($D_P = D_{P'}$), they are equally general, hence equally falsifiable. Yet not all such theories, which fit the observations equally well, seem to be preferred to the same extent. Considering the examples of subsection 1.1 again, all theories (a)-(d) are equally falsifiable. Admitting that they are not equally intuitive, nor equally "simple" (according to our definition or any other), would mean that simplicity cannot be equated with falsifiability and, moreover, that the former may help classify theories where the latter fails to do so.

1.5 Simplicity Versus Explanation

Another highly idealized assumption made in the model presented above is that the contestant theories have to perfectly fit the gathered evidence. However, if the smallest discrepancy sufficed to rule out a theory, very few simple theories would remain.

Thus, a more accurate description would include a trade-off between a theory's simplicity and explanatory power. A nice (and very simple) example

is linear regression, in which one variable is assumed to be explained by a linear combination of other variables. The quality of explanation is traditionally measured by "R^2" (the ratio of explained to total "variance"); the complexity of the theory may simply be measured by the number of explanatory variables, which will also correspond to the length of the theory in any reasonable language. What is sometimes called "the adjusted R^2" may be viewed as an attempt to summarize this trade-off in a single number.

In general, there would be many different ways to measure "explanatory power"; the important thing to note is that for simplicism to make sense one has to admit that the trade-off exists, and that simplicity may well be sacrificed for higher quality of explanation.

Yet, probability and statistics provide an insightful example in which the quality of explanation is sacrificed for the sake of simplicity: suppose it rained every Monday, but on no other day of the week. A simple deterministic theory could fit this evidence perfectly well. Unfortunately, this is not the case. Moreover, no obvious deterministic pattern exists in the rainy days. Hence, we resort to probabilistic theories, such as "on a summer day it rains with probability 35 percent." Notice that such a theory is much simpler than deterministic ones which could fit the data perfectly, i.e., theories that would look very much like example (d) in subsection 1.1 above. However, it does not provide nearly as good an explanation as these deterministic theories--it actually does not purport to predict the weather on a single day, it only tries to "fit the data" in a new, weaker, sense of asymptotic frequencies.

1.6 Relative Complexity

In view of this discussion and of examples such as (d) in subsection 1.1, one is tempted to suggest an alternative complexity criterion. Rather than using an absolute measure, a relative one suggests itself: the complexity of a theory P is measured in relation to the complexity (i.e., "length") of the observations it purports to predict (that is, $D_P \cap O$, namely, the size of the intersection of P's domain with the set of available observations, without reference to the accuracy of the prediction). Such a measure would exclude point-by-point theories (such as (d) above) and would admit that a certain trade-off between simplicity and generality may be intuitively appealing.

Obviously, in the idealized model of subsection 1.2, where all theories are as general as could be, minimization of relative complexity and of absolute complexity boil down to the same thing.

Notice that relative complexity, like the absolute one, may be traded off for explanatory power as in subsection 1.5.

1.7 Is Simplicism a Scientific Theory?

The model of subsection 1.2 provided a framework to analyze scientific activities, but also suggested a theory regarding the development of theories. It is only natural to ask whether simplicism itself may be described by this model, and if so, does simplicism prescribe that we choose simplicism (as the simplest theory)?

Since simplicism does not purport to explain the choice of language, one must assume the latter given. Thus, an instance of the problem simplicism attempts to solve is characterized by a language, (T_U, T_L), and a

set of observations, $\{x_i\}_{i\in O}$. The fact that for such L and $\{x_i\}_{i\in O}$ a certain theory P was chosen by scientists is a single observation simplicism should try to explain.

Therefore, the questions are: (i) Is simplicism a scientific theory, i.e., can it be formulated by a program?; (ii) If so, does it fit given observations reasonably well?: (iii) If this is the case, is it the simplest theory (i.e. the shortest program) doing so? (The choice of language here is, again, arbitrary.)

Unfortunately, the answer to the first question is negative: simplicism is not a scientific theory according to our definition. For it to be one, there should be a Turing machine T such that, given a set of observations and a language, computes a minimal program (in the given language) which fits the observations. The difficulty with actual enumeration and trial of all programs in the given language (in ascending complexity order) is, of course, that they may not halt.

More precisely, let us define the minimal complexity problem: Given a description of a language (T_U, T_L), a set of observations $\{x_i\}_{i\in O}$ and a program P in L, satisfying $T_U(T_L(P), i) = x_i$ for $i \in O$, is there a shorter program P' (in L) such that $T_U(T_L(P'), i) = x_i$ for $i \in O$?

Proposition 1: The minimal complexity problem is undecidable.[1] (See Pager (1969) for related results.)

Thus, the other two questions (namely, "Does simplicism explain the observed data?" and "Is simplicism the simplest theory for the philosophy of

---

[1] All proofs are relegated to the Appendix.

science?") remain what, in view of subsection 1.3, they are doomed to be--a matter of taste.

1.8 Representation and Language

It was pointed out in subsection 1.3 above that language plays an important role not only in the computer language in which theories are given (which is explicit in the model), but also in the language of the questions and possible answers (which is implicit in the discussion so far). More specifically, one may, for example, interchange bits 2i and (2i - 1) for all $i \geq 1$ in every state of the world, resulting in a different representation with respect to which different theories may be the "simplest."

Yet it is obvious that in this example no loss of generality is involved in assuming a given representation: the freedom in the choice of the langauge L may compensate for the specificity of the representation. In particular, for every language L there is a language L', such that for every theory P in L, P's computation in the transformed representation is equivalent to its computation as a program in L' in the original representation: L' has to translate P (for T_U) in such a way that, given the question 2i (alternatively, (2i - 1)), it simulates P on (2i - 1) (respectively, 2i).

To be both more general and more precise one should model the process of representation of questions and answers. Starting with some abstract set Ω_0 of states of the world, a representation is a 1-1 function $R: \Omega_0 \rightarrow \Omega$. For simplicity, let us consider representations which are also onto. Given two such representations, R_1 and R_2, $(R_2 \circ R_1^{-1})$ is a bijection on Ω.

Let us call a bijection $B: \Omega \rightarrow \Omega$ computable if there is a 2-input-tape

Turing matching, T_B, such that for every $\omega \in \Omega$ and $i \in \mathbb{N}$, if T_B accepts the (infinite) string $(\omega_1,\omega_2,\omega_3,\ldots)$ on one tape and i on the other--it computes $B(\omega)(i)$ in finite time. That is, for every $\omega \in \Omega$, $T_B(\omega,i) \simeq B(\omega)(i)$ for all $i \in \mathbb{N}$.

Proposition 2: If $B: \Omega \to \Omega$ is a computable bijection, so is B^{-1}.

Proposition 3: If $B_1,B_2: \Omega \to \Omega$ are computable bijections, so is $B_1 \circ B_2$.

Notice that computable bijections thus form a group (with respect to function composition). Furthermore, defining two representations, R_1 and R_2, to be computationally equivalent if $(R_2 \circ R_1^{-1})$ is computable, one concludes that computational equivalence is indeed an equivalence relation.

To verify that the freedom in the choice of language L may compensate for the arbitrariness of the representation within an equivalence class of this relation, we note that:

Proposition 4: For every language (T_U,T_L) and every computable bijection $B: \Omega \to \Omega$ there is a language (T_U,T_L') such that for every theory P, if P computes ω in L, P also computes $B(\omega)$ in L'.

This proposition shows that the model of subsection 1.2 above is not as arbitrary as it may first seem: as long as two representations are computationally equivalent--that is, as long as there is an algorithmic way to translate one into the other (hence, also vice versa)--the same programs (theories) will have the same predictions (up to this translation) provided they are interpreted in the appropriate language.

1.9 Learning as a Normative Argument

Although simplicism is presented in this paper as a purely descriptive theory, it is natural to ask whether it can also be justified on normative grounds. One possible such argument for simplicism (to which this subsection is devoted) is that following simplicism may lead to knowledge of the true state of the world.

Consider a dynamic process of evidence gathering. Formally, let $\{O_t\}_{t\geq 1}$ be an increasing sequence of sets of observations converging to $\mathbb{N}$ (denoted $O_t \uparrow \mathbb{N}$), i.e., $O_t \subseteq O_{t-1}$ and $\bigcup_{t\geq 1} O_t = \mathbb{N}$ (where $\cdot O_t < \infty$ for all $t \geq 1$). For a given language (T_U, T_L) and a truth $x \in \Omega$, recall that $\mathcal{P}_{(x,O_t,L)}$ denotes the set of theories in L which predict x_i for $i \in O_t$.

Obviously, there is no hope to "learn" the truth x unless it is computable in L. However, it is easy to see that if this happens to be the case, every simplicistic choice function c will "learn" x. A closer inspection, though, will easily convince the reader that a much wider class of choice functions share the same property.

Let a choice function $c: (T_U, T_L, \mathcal{P}) \rightarrowtail \mathcal{P}$ be <u>exhaustive</u> if for every language L, state of the world $\omega \in \Omega$ and sequence of sets of observations $O_t \uparrow \mathbb{N}$, every program P in L satisfies at least one of the two:

(i) some P' which is L-equivalent to P is chosen at some point, i.e., $P' \in c(T_U, T_L, \mathcal{P}_{(\omega,O_t,L)})$ for some $t \geq 1$;

(ii) P is contradicted by evidence, that is, $T_U(T_L(P), i) \neq \omega_i$ for some $i \in O_t$ and some $t \geq 1$.

Obviously, a choice function is exhaustive if and only if

$$[\cup_{t\geq 1} c(T_U, T_L, \mathcal{P}_{(\omega, O_t, L)})] \cap [\cap_{t\geq 1} \mathcal{P}_{(\omega, O_t, L)}] \neq \emptyset$$

for every computable $\omega \in \Omega$ and every $\{O_t\}_{t\geq 1}$ with $O_t \uparrow \mathbb{N}$.

Next define a choice function c to be independent of irrelevant alternatives (IIA) if for every $\omega \in \Omega$ and language L, $O_1 \subseteq O_2$ implies $c(T_U, T_L, \mathcal{P}_{(\omega, O_2, L)}) = c(T_U, T_L, \mathcal{P}_{(\omega, O_1, L)}) \cap \mathcal{P}_{(\omega, O_2, L)}$ whenever the latter is nonempty.

Proposition 5: A simplicistic choice function, c_s, is exhaustive and IIA.

Proposition 6: Let c be some exhaustive and IIA choice function, and let x by computable in a language L. Let $O_t \uparrow \mathbb{N}$. Then there exists $T < \infty$ such that for $t \geq T$, $c(T_U, T_L, \mathcal{P}_{(x, O_t, L)})$ is constant and consists only of programs P which compute x in L.

Hence simplicistic choice functions are bound to learn the truth whenever the latter is learnable, but they are by no means the unique ones doing so, and these results can hardly be considered a normative justification of simplicism.

However, one may wonder whether the arbitrariness of language can be invoked to show that simplicism is, indeed, the unique exhaustive and IIA choice function. That is: Given such a choice function c and a language L, is there a language L' with respect to which c is the simplicistic one? Namely, does c happen to choose programs in L which are, in fact, the shortest ones when considered as programs in L'?

More formally, let a choice function c be pseudo-simplicistic if for every L there exists L' such that $c(T_U, T_L, \mathcal{P}_{(x,O,L)}) = c_s(T_U, T_{L'}, \mathcal{P}_{(x,O,L')})$

for every $x \in \Omega$ and $O \subseteq \mathbb{N}$, where c_s denotes some simplicistic choice function. (Note that all simplicistic choice functions coincide on arguments of the form given here.) With this notation we note that:

Proposition 7: There are exhaustive and IIA choice functions which are not pseudo-simplicistic.

We therefore conclude that "learning" the computable truth x cannot single out simplicism as a desirable rule. In Section 2 below we will contend that truth is very unlikely to be computable to begin with (based on cardinality arguments), an argument that will further undermine the learning property as a normative argument for simplicism.

1.10 Final Remarks

a. The extent to which simplicism makes sense in specific examples may depend on the scope of observations considered. Considering the weather of last week, for instance, could hardly make a probabilistic model "simple," as it requires a non-trivial apparatus. However, people may still resort to this model because, when a longer period of time is taken into account, it is the "simplest" one with some sort of explanatory power.

One may try to solve the domain-specification problem by assuming that all "meaningful" (finite) questions in English are enumerated and encoded, where theories of the various fields of science are all embedded in one model with appropriate domains, D_p. Despite some obvious problems (such as the meaning of "meaningful"), this solution may be theoretically valid. From a practical viewpoint, however, it is of little help: since scientific

theories are not given as programs with a specific domain, D_p (outside which a "no answer" output is produced), their formulation as such entails the specification of their domain.

Thus, we are left with the observation that simplicism, verbally described, should be qualified by "for the appropriate domain and representation of observations" as well as by "for the appropriate choice of language."

b. The model presented above required that a scientific theory have a specific answer ("0", "1", or "no answer") for every question ($i \in \mathbb{N}$). Thus, a rule such as "the atoms of every element have a fixed number of protons in their nucleus" will not qualify as a theory without the specification of the periodical table; nor will "y is a function of x" (for some observable variables x and y) without the specification of this function. This definition seems to avoid an artificial distinction between the "theory" and its "parameters," and does not allow for the complexity to be hidden in the latter.

c. It is worthy of note that simplicism may apply to less traditional fields of the philosophy of science. For instance, an every-day term such as "to understand a movie" may be modeled as saying "to come up with a simple theory that would explain the observation 'movie'." That is, 120 minutes of pictures and sounds may be considered as the data to be explained, and the shorter the explanation provided, the better one's understanding. Of course, the simplicity-explanation trade-off and relative simplicity would apply here as well.

d. Assume that a black and white movie turns into color in its 39th minute and then switches back to black and white. A short program that explains (i.e., regenerates) every other aspect reasonably well could be added a one-line command:

```
If minute = 39 then color
                else black_and_white;
```

and thus explain all there is to explain with low complexity. Yet, this condition seems inferior to, say:

```
If hero_understands_meaning_of_life = true
     then color
     else black_and_white;
```

That is to say, we may prefer qualitative descriptions to quantitative ones. This may be captured by simplicism since the binary code of "39" is longer than that of "true." Furthermore, the use of many arbitrary constants will make a program longer while few constants (such as 0, 1, π, and e) can be computed by procedures which appear once in the program, though are possibly repeatedly invoked in the code.

Finally, note that an appropriate choice of language may provide an intuitive balance between qualitative but long explanations and quantitative but "short" ones. For instance, it may be more appropriate to use unary (rather than binary) numerical representation to make a large (arbitrary)

constant such as 39 longer than "understanding the meaning of life."

2. Kolmogorov's Impossibility Theorem

We first present a simple observation, which is trivial given the framework and will be nicknamed "Kolmogorov's Impossibilty theorem." In an explicit (and stronger) form it appears in Martin-Lof (1966). Next we suggest some interpretations, and conclude with a brief discussion of the extent to which they make sense.

2.1 Observation

Let $\Delta \subseteq [0,1]$ be the Cantor set, i.e., $\Delta = \{\sum_{i=1}^{\infty} 2x_i 3^{-i} \; x_i \in \{0,1\}\}$. Let $(\Delta,\mathcal{B},\mu)$ be some probability space where $\mathcal{B}$ is a σ-algebra containing $\{x\}$ for all $x \in \Delta$, and μ is a nonatomic σ-additive measure. (In particular, this implies $\mu(\{x\}) = 0$ for all $x \in \Delta$.) Let $\Delta_0 \subseteq \Delta$ denote the set of finite complexity points in Δ, i.e., $x = \sum_{i=1}^{\infty} 2x_i 3^{-i} \in \Delta_0$ if and only if there is a Turing machine T_x such that for every $i \in \mathbb{N}$, when T_x gets i as an input it halts and outputs x_i. (Note that $\{x_i\}_i$ is uniquely defined by $x \in \Delta$ and vice versa.) Then $\mu(\Delta_0) = 0$. (In this formulation, a proof is not called for since Δ_0 is countable.)

2.2 Suggested Interpretation

In the framework of subsection 1.2 above, Δ may be identified with the set of all states of the world Ω, where Δ_0 corresponds to the set of computable ω in a certain language L. Not knowing what truth really is, we (as modelers, scientists, and so forth) may have beliefs given by μ. Should μ be nonatomic, with probability 1 truth will never be discovered. Hence,

the process suggested by simplicism--i.e., finding the simplest theory which matches observations, and recomputing it once contradictory evidence was detected--is doomed to continue forever.

Let us consider a slightly different interpretation. Finding a set of moral laws a person (or a society) would like to abide by may be viewed as an attempt to formalize one's intuition. For various reasons, one may have a strong intuition that murder is bad. So one may decide to have "do not kill" in one's codex. Yet this is only an approximation to intuition, and it is not unlikely that one may find oneself in a situation of conflict between the intuition and its formal approximation and, say, decide to qualify this rule by allowing self-defense killing. But, then again, this qualified rule may still be too crude and contradict the moral intuition in another instance. Will the process stop? Are we likely to formalize our intuition precisely?

Obviously, the problem may be viewed as a scientific one: all moral decision situations that may be described in finitely long English statements are countable, and so are the describable actions. Thus, one's intuition may be modeled as "truth," i.e., a point $x \in \Delta$. The set of decision problems that have occurred to a person by a given point of time is equivalent to the set of observations. Indeed, a decision problem one has thought about is one observation on one's moral intuition. The set of rules we want to formulate is an algorithm that, given a specific moral decision problem, should halt and compute a "moral" choice. The question of existence of such an algorithm that will always fit the intuition, i.e., of existence of a true scientific theory, reduces to the complexity of "truth" --that is, of one's moral intuition.

At this point the reader is asked to suppose that a nonatomic measure on $(\Delta, \mathcal{B})$ is a reasonable assumption. Kolmogorov's impossibility theorem may then be invoked to say that "truth," i.e., one's true moral intuition, is infinitely complex with probability 1. Hence, we are probably doomed to keep facing moral dilemmas in which our formal laws seem to fail; therefore, even if our codex were algorithmic, there would be room for human judgment in its implementation.

The moral codex problem may be considered as a problem of definition; in this case, defining "moral" or, alternatively, "immoral." However, the same arguments would apply (to a larger or smaller extent) to other cases of definitions. In general, the process of a definition of a concept (the pragmatic motivation for which will not be discussed here) can be modeled as a formalization of given intuition. For instance, the definition of a "work of art" or (worse still) "good art" starts with intuition, which again may be modeled as truth, some $x \in \Delta$, which specifies for each (finitely describable) object whether it is art work and/or whether it is "good art." Finding the definition means spelling out an algorithm that can compute this intuition, and this is done based on finitely many "observations," i.e., cases which were already encountered and studied. Hence, if μ is nonatomic, a perfect definition exists with probability zero.

2.3 Discussion

How realistic are the assumptions made in the interpretations suggested above? To what extent do they fit our intuition?

Some readers may certainly not like the focus on those things one may describe in finitely many words. It should be mentioned, however, that the

crucial point is that the algorithm--the codex, the definition of "art," and so forth--is finitely describable. That is, if the set of possible dilemmas is uncountable, the likelihood of a finitely complex intuition certainly does not increase.

Followers of de Finetti will probably object to the assumption that μ is σ-additive (see de Finetti (1949, 1950), Savage (1954) and Dubins and Savage (1965)). Without delving into this discussion we will only note that σ-additivity of a probability measure is by far more commonly assumed than not.

The weakest point in the last two interpretations suggested above seems to be the nonatomicity of μ. While this assumption makes sense in the context of an objective "truth" (chosen by God or Nature), it is certainly arguable when applied to one's intuition (regarding "good" and "evil," "art" and "junk," and so forth). One may follow Turing (1956) and contend that the human mind is precisely the machine that implements the required (finite) algorithm. Hence, μ is a priori concentrated on finite complexity intuitions.

The writer of these lines finds this argument quite convincing from a theoretical viewpoint. Yet, for slightly more practical situations it seems more precise to model human intuition as if it could be of infinite complexity. Should one try to draw conclusions from this analysis regarding the actual likelihood of finding a definition that captures intuition perfectly well, it would be misleading to use the Turing argument. <u>For practical purposes</u> it seems that the gap between human intuition and feasible definitions makes a model, in which intuition may be infinitely complex, a better qualitative description of reality.

This discussion seems to be a perfect point to conclude this paper. The two notes presented here give a somewhat naive, certainly oversimplified, mathematical models of human thinking activities.

As with any model in any field, they should not be absolutely faithful descriptions of reality, nor should they be taken too seriously in general. Their main goal is to provide an additional point of view on and hopefully some insight into the phenomenon under consideration, and it is the author's hope that even if this goal was not achieved, this paper may be of help in clarifying some concepts and opinions.

References

Dubins, L. E. and L. J. Savage (1965), How to Gamble if You Must, New York: McGraw Hill.

Carnap, R. (1923), "Uber die Aufgabe der Physik und die Andwednung des Grundsatze der Einfachstheit," Kant-Studien, 28, 90-107.

de Finetti, B. (1949), "Sull Impostazione Assiomatica del Calcolo delle Probabilita," Annali Triestini, 19, 29-81.

de Finetti, B. (1950), "Aggiunta alla Nota sull' Assiomatica della Probabilita," Annali Triestini, 20, 5-22.

Fine, T. L. (1973), Theories of Probability, New York: Academic Press.

Gilboa, I. and D. Schmeidler (1989), "Infinite Histories and Steady Orbits in Repeated Games," mimeo.

Goodman, N. (1965), Fact, Fiction and Forecast, 2nd edition, Indianapolis: Bobbs-Merrill.

Kolmogorov, A. (1963), "On Tables of Random Numbers," Sankhya Ser. A, 369-376.

Kolmogorov, A. (1965), "Three Approaches to the Quantitative Definition of Information," Problems of Information Transmission (translated from Problemy Peredaci Informacii), 1, 1-7.

Loveland, D. W. (1969), "A Variant of the Kolmogorov Concept of Complexity," Information and Control, 15, 510-526.

Martin-Lof, P. (1966), "The Definition of Random Sequences," Information and Control, 9, 602-619.

Pager, D. (1969), "On a Problem of Finding Minimal Programs for Tables," Information and Control, 14, 550-554.

Popper, K. R. (1934), Logik der Forschung; English edition (1958), The Logic of Scientific Discovery, London: Hutchinson and Co. Reprinted (1961), New York: Science Editions.

Russel, B. (1945), A History of Western Philosophy, New York: Simon and Schuster.

Sober, E. (1975), Simplicity, Oxford: Clarendon Press.

Suppe, F. (1974), The Structure of Scientific Theories (edited with a critical introduction by F. Suppe), Urbana, Chicago, London: University of Illinois Press.

Turing, A. M. (1956), "Can a Machine Think?", in The World of Mathematics, Vol. IV, 2099-2123, New York: Simon and Schuster.

Wittgenstein, L. (1922), Tractatus Logico Philosophicus, London: Routledge and Kegan Paul; fifth impression, 1951.

Appendix: Proofs of Propositions

Proposition 1: By reduction of the halting problem: let there be given a (description of a) Turing machine M and input j for M. Define the set of observations O to be {1} and let the truth observed be $x_1 = 1$. Next define a language L as follows: if the input string P is (M,j), L writes for T_U the commands: "Simulate the run of M on j and then (regardless of M's output and P's input) write 1." Otherwise, if the input string is (the concatenation of) (M,j,M,j,s) for some string s, L writes the string s (as a command for T_U). Finally, if the input string is none of the above, L outputs (for T_U) the command "write 0." Obviously, P = (M,j) is the shortest program in L (which computes $x_1 = 1$) if and only if M halts on j. //

Proposition 2: Let $B: \Omega \to \Omega$ be computable, and let T_B be a Turing machine which computes it, i.e., $T_B(\omega,i) = B(\omega)(i)$ for all $\omega \in \Omega$ and all $i \in \mathbb{N}$. We first note the following:

Lemma: For every $i \in \mathbb{N}$ there is an $n_i \in \mathbb{N}$ such that for every $\omega \in \Omega$, T_B does not consult $\omega(j)$ for $j > n_i$ in the computation $T_B(\omega,i)$. Furthermore, given (the description of) T_B and $i \in \mathbb{N}$, such an integer n_i can be computed by a Turing machine in finite time.

Proof: The existence of such an integer n_i is an application of Konig's lemma and has appeared in an almost identical framework in Gilboa and Schmeidler (1989, Proposition 3.1). For completeness' sake, we provide a

sketch of the proof: let $i \in \mathbb{N}$ be given. Consider the (infinite) binary tree in which every edge determines the value of $\omega(j)$ for some $j \in \mathbb{N}$, and every node corresponds to a finite sequence of bits $\omega(1),\ldots,\omega(j)$ for some $j \in \mathbb{N}$. Obviously, a state of the world ω corresponds to an (infinite) path in this tree.

Next consider the computation of T_B given i and all possible states ω. For every ω, T_B is known to halt. Hence, along every path in the tree, T_B's computation may reach only finitely many nodes. Assume, contrary to the claim, that a uniform bound n_i does not exist, and consider the root of the tree. It must be the case that at least one of its two subtrees does not have a uniform bound on the length paths entering it. Continuing with the root of this subtree, one generates an infinite path. But then T_B will not halt for the state ω, defined by the path, and the input i, which is a contradiction. Hence, such integers n_i do exist.

Next we have to show that such an integer, say, the minimal one, can be computed by a Turing machine. Yet this is straightforward: for every $n \geq 1$ one may enumerate all the 2^n possible prefixes of ω, and simulate the run of T_B on (ω,i) for each prefix. Should one of these computations try to read $\omega(j)$ for $j > n$, n should be increased to $(n + 1)$ and the process starts again. The first n for which all 2^n prefixes do not induce reading $\omega(j)$ for $j > n$ is the minimal n_i. By the existence proof, this algorithm is bound to halt. //

We now turn to the proof of Proposition 2. We will describe a Turing machine, $T_{B^{-1}}$ such that, given $\omega' \in \Omega$ and $j \in \mathbb{N}$, $T_{B^{-1}}(\omega',j) = \omega(j)$ for $\omega = B^{-1}(\omega')$.

$T_{B^{-1}}$ will perform the following algorithm:

1. Set k to 1.
2. Compute the minimal n_k provided by the lemma. Compute $\bar{n}_k = n_k$ for $k = 1$ and $\bar{n}_k = \max\{n_k, \bar{n}_{k-1}\}$ for $k > 1$.
3. Compute all $2^{\bar{n}_k}$ sequences of length $\bar{n}_k$. For each one of them, simulate T_B on this sequence (as a prefix of some ω) and k. Let A_k be the set of prefixes for which the computation ended with $\omega'(k)$.
4. Let $\bar{A}_k = A_k$ for $k = 1$ and

$$\bar{A}_k = A_k \cap (\bar{A}_{k-1} \circ \{0,1\}^{\bar{n}_k - \bar{n}_{k-1}})$$

where $\circ$ stands for concatenation. (That is, $\bar{A}_k$ is the set of all $\bar{n}_k$-long prefixes in A_k which are also continuations of prefixes in $\bar{A}_{k-1}$).

5. If $\bar{n}_k \geq j$ and all prefixes in $\bar{A}_k$ have the same j-th bit value, output this value and halt. Otherwise, set k to $(k + 1)$ and go to (2).

In words, the algorithm tries to compute the (known) value $\omega'(k)$ based on (the unknown) ω. Since for every k only finitely many (n_k) bits of ω are used to compute $B(\omega)(k)$, all possible prefixes can be tried, where $\bar{A}_k$ contains only those which fit $\omega'(r)$ $1 \leq r \leq k$.

It is obvious (for cardinality reasons) that $\bar{n}_k \to \infty$ as $k \to \infty$. It is also clear that should the algorithm halt, its computation is correct, i.e., $T_{B^{-1}}(\omega', j) = \omega(j) = B^{-1}(\omega')(j)$. All that we have to prove, therefore, is

that for large enough k, all prefixes in $\bar{A}_k$ will have an identical j-th bit value.

Assume this is not the case. Then for every k (such that $\bar{n}_k \geq j$) $\bar{A}_k$ contains at least one prefix $(x_1,\ldots,x_{\bar{n}_k})$ with $x_j = 1$ and at least one prefix $(y_1,\ldots,y_{\bar{n}_k})$ with $y_j = 0$. Since every prefix in $\bar{A}_k$ is the continuation of some prefix in $\bar{A}_{k-1}$, all these prefixes may be identified with paths in the binary tree used in the lemma. Considering the 2^{j-1} subtrees beginning at a node specifying $\omega_j = 1$ (i.e., a node in depth j corresponding to the value of 1), at least one of them has to have unboundedly long paths, and applying Konig's lemma again, at least one of them has to contain an infinite path. Let ω be the state of the world defined by this path. Then $T_B(\omega,k) = \omega'(k)$ for all k, which means that $B(\omega) = \omega'$.

However, the same argument for the subtrees with $\omega_j = 0$ yields another state $\bar{\omega}$ such that $B(\bar{\omega}) = \omega'$ as well, a contradiction. Hence, at some point $\bar{A}_k$ will contain prefixes with identical j-th bit, and the algorithm halts. //

Proposition 3: Let B_1 and B_2 be computable by T_{B_1} and T_{B_2}, respectively. The machine $T_{B_1 o B_2}$ that would compute $B_1 o B_2$ will operate as follows: given (ω,i), $T_{B_1 o B_2}$ computes $B_1(B_2(\omega))(i)$ by simulating T_{B_1} on the input $(B_2(\omega),i)$. Whenever T_{B_1} tries to read a bit from its first input string, say, $B_2(\omega)(j)$, it invokes T_{B_2} to compute it with the input (ω,j). Obviously, $T_{B_1 o B_2}$ halts since both T_{B_1} and T_{B_2} always halt. //

Proposition 4: Given a language (T_U,T_L) and a bijection $B: \Omega \rightarrow \Omega$ computable

by T_{B_1}, define a language L' by $(T_U, T_{L'})$ where $T_{L'}$ operates as follows: given a program P, write (as input for T_U) the description of a machine M which, given input i, tries to simulate T_B on the input (ω, i). Whenever T_B tries to read some bit value $\omega(j)$, M simulates $T_U(T_L(P), j)$. Since P computes ω in L, $T_U(T_L(P), j) = \omega(j)$ and P thus computes $B(\omega)$ in L'.

Proposition 5: Obviously c_s is IIA. To show exhaustiveness, note that only computable $\omega \in \Omega$ need be considered. Let ω be such, and consider the shortest P computing ω in L. Every P' which is shorter than P has $T_U(T_L(P'), i) \neq \omega_i$ for some i; hence, $P' \notin \mathcal{P}_{(\omega, O_t, L)}$ for some $t \geq 1$. Since there are finitely many such (P')'s, P will eventually be chosen.

Proposition 6: Let x be computable in L and let $\mathcal{P}_{x,L} = \{P \mid P \text{ computes } x \text{ in } L\}$. By exhaustiveness, there is a $t \geq 1$ such that $c(T_U, T_L, \mathcal{P}_{(x, O_t, L)}) \cap \mathcal{P}_{x,L} \neq \emptyset$. Let t_0 be the minimal such t. Since c is IIA, for some $T \geq 1$, $c(T_U, T_L, \mathcal{P}_{(x, O_t, L)}) = c(T_U, T_L, \mathcal{P}_{(x, O_{t_0}, L)}) \cap \mathcal{P}_{x,L}$ for $t \geq T$. //

Proposition 7: Fix a language L_0 and consider a family of choice functions which are equal on $L \neq L_0$, e.g., all choice functions c such that $c(T_U, T_L, \mathcal{P}) = c_s(T_U, T_L, \mathcal{P})$ for all T_U, all $\mathcal{P}$ and all $L \neq L_0$ for some simplicistic c_s. The set of pseudo-simplicistic choice functions in this family is countable. However, for every bijection $\mathbb{N} \to \mathbb{N}$ one can define a choice function c_B which chooses the B-minimal program in $\mathcal{P}$ for $(T_U, T_{L_0}, \mathcal{P})$. For every B, c_B is exhaustive and IIA. Since there are uncountably many bijections, there are functions c_s which are not pseudo-simplicistic.

(This result would hold even if one restricts the domain of choice

functions to include only sets $\mathcal{P}$ of the form $\mathcal{P}_{(x,0,L)}$. Not all pairs of bijections will necessarily result in different choice functions in this case, yet there will be uncountably many different ones.) //