Theoretical Computer Science 4 (1977) 301–319.

ON THE INFERENCE OF OPTIMAL DESCRIPTIONS*

Robert DALEY
University of Pittsburgh, Department of Computer Science, Pittsburgh, PA 15260, USA

Communicated by Albert Meyer
Received June 1975
Revised January 1977

1. Introduction

We are concerned in this paper with an inference problem which is not an inductive inference or grammatical inference problem, but which does seem related to the scientific investigation of phenomena. In order to see this relationship we consider the following scenario.

A scientist wishing to investigate a certain phenomenon performs an experiment and obtains some data. Now, in a sense this data is itself a description of the phenomenon, but the scientist is more interested in discovering some scientific law or principle which explains the phenomenon (or at least agrees with the experimental data). Let us presume that she is successful in formulating such a law. Eventually the law will be verified and become accepted by the scientific community. Later, perhaps, this law will be incorporated into a broader, simpler principle.

Of interest to us in the foregoing scenario is the apparent concern on the part of the scientific community in finding ever more succinct descriptions (i.e., laws) for phenomena. Now by no means is the shortest such description the most convenient to use. Indeed, the application of very general principles to a concrete problem may require a rather large computational effort. However, these short descriptions or general principles are valuable in that they enable us to absorb and understand the phenomena more easily. This is admittedly a rather empiricist point of view but it does provide a setting and some motivation for what follows. We should also point out that we are not concerned with experimental error (all data are 100% accurate!).

The central theme of this paper is the modelling and investigation of the inference problem implicit in the above discussion (viz., given some data to find its shortest description) in the case where the inference procedure is effective. A phenomenon is represented by an infinite binary sequence. If x is an infinite binary

* The research reported here was supported by NSF Grants GJ-31223 and GJ-43223. Some of these results were originally presented at the conferences cited in [8, 9, 10, 11] of the references.

sequence then we denote the n^{th} digit of x by $x(n)$ and the initial segment of x of length n by x^n, i.e., $x^n = x(1)\cdots x(n)$. The data from an experiment on the phenomenon x is represented by x^n where the domain of the experiment is the intial segment $\{0, 1, \ldots, n\}$ of the natural numbers. While this may seem to be a servere restriction on the type of experiments permitted, we will see later that for certain phenomena we can relax this restriction and permit experiments which are designed by an effective procedure. Descriptions for the data x^n are programs π in some "universal programming system" which compute x^n, i.e., $\forall m \leq n_{\blacksquare}\pi(m) = x(m)$. The minimal description of x^n then is a program of minimal length which computes x^n, which we denote here by $M(x^n)$. The inference problem for x is then solvable if and only if there exists a recursive function ψ such that for each n, $\psi(x^n) = M(x^n)$.

In general, the problem of finding $M(x^n)$ from x^n is certainly not recursively solvable (see Pager [18, 19] and and Schubert [26]). In contrast to Pager [18] our success in constructing sequences with recursively solvable inference problems is due apparently to our not requiring that a program for x^n contain an encoding of the number n. We point out that the idea of associating programs with descriptions of data from a scientific experiment, or the concern in finding minimal programs within such a framework is not new. These notions are explicit in the very early papers on this subject by Chaitin [5] and Solomonoff [29]. It should also be clear that this is not an inductive inference problem. There is no inference in the limit. For a discussion of inductive inference and grammatical inference problems the reader is referred to Bierman and Feldman [2], Blum and Blum [4], and Gold [13]. For other discussions of this inference problem the reader is referred to Simon [27, 28].

In Section 2 of this paper we give a precise formulation of the inference problem and consider those sequences for which the inference problem is solvable by a special type of recursive function called an "inference device." We give a characterization of such sequences and investigate their minimal-program complexity. In the process we will construct a number of sequences with additionally interesting properties. Some alternative formulations of the inference problem are investigated in Section 3.

We conclude with a number of open questions. The remainder of this section is devoted to definitions and notation as well as some basic results from the minimal-program complexity theory which will be used in the remainder of the paper.

Every infinite binary sequence can be regarded as the characteristic sequence of some set. The characteristic sequence x of the set X is defined by $x(n) = 1 \Longleftrightarrow n \in X$. We will use the letters x, y, and z (with or without subscripts) for infinite binary sequences and the letters X, Y, and Z (with the same subscripts) to denote the corresponding sets. We use the letters u, v, and w to denote finite binary sequences (i.e., strings), and $|w|$ to denote the length of w, and for $m \leq |w|$, $w(m)$

to denote the m^{th} digit of w. We identify a natural number i with its binary representation and use $|i|$ to denote the length of that representation, i.e., $|i| = 1 + \log_2 i$. We will regard as a *universal programming system* (u.p.s.) any acceptable gödel numbering $\{\varphi_i\}$ (see Rogers [18]) of the partial recursive functions for which the $S-m-n$ function $\sigma(i,j)$ which satisfies $\varphi_{\sigma(i,j)}(n) = \varphi_i(j,n)$ also satisfies $|\sigma(i,j)| = |j| + \delta(i)$, where δ is a total recursive function. The minimal-program complexity of w is defined by,

$$K(w;|w|) = \min\{|i| \mid \forall m \leq |w| \blacksquare \varphi_i(m) = w(m)\}.$$

We define the set of functions

$$\mathscr{F} = \{f \mid f \text{ is total recursive, unbounded and non-decreasing}\}.$$

Let $\{\Phi_i\}$ be a computational complexity measure for $\{\varphi_i\}$ (see Blum [3]). The value $\Phi_i(n)$ will be referred to generically as the computation time for program i on input n. If $t \in \mathscr{F}$ then we define the t-bounded minimal-program complexity of x^n by

$$K^t(w;|w|) = \min\{|i| \mid \forall m \leq |w| \blacksquare \varphi_i(m) = w(m) \text{ and } \Phi_i(m) \leq t(|w|)\}.$$

Clearly, $K(w;|w|) \leq K^t(w;|w|)$ for any $t \in \mathscr{F}$. The complexity classes are defined by,

$$\mathscr{C}[f] = \{x \mid \overset{\infty}{\forall} n \blacksquare K(x^n;n) \leq f(n)\},$$

$$\mathscr{C}[f \mid t] = \{x \mid \overset{\infty}{\forall} n \blacksquare K^t(x^n;n) \leq f(n)\},$$

$$\mathscr{C}^{bd}[f] = \bigcup_{t \in \mathscr{F}} \mathscr{C}[f \mid t],$$

where "$\forall^\infty n$" and "$\exists^\infty n$" are used to denote the expressions "for all but finitely many n" and "there exist infinitely many n", respectively. It is clear that the definition of $K(w;|w|)$ and $K^t(w;|w|)$ depends on the choice of $\{\varphi_i\}$ and $\{\Phi_i\}$. If $\{\hat{\varphi}_i\}$ is some other u.p.s. and $\{\hat{\Phi}_i\}$ a complexity measure for $\{\hat{\varphi}_i\}$ we use $\hat{K}(w;|w|)$ and $\hat{K}^t(w;|w|)$ to denote the minimal-program complexities based on $\{\hat{\varphi}_i\}$ and $\{\hat{\Phi}_i\}$. If $\{\varphi_i\}$ and $\{\hat{\varphi}_i\}$ are u.p.s.'s then there is a constant c such that $\forall w \blacksquare |K(w;|w|) - \hat{K}(w;|w|)| \leq c$. Similarly,

$$\exists c \blacksquare \forall t \in \mathscr{F} \blacksquare \exists \hat{t} \in \mathscr{F} \blacksquare \forall w \blacksquare |K^t(w;|w|) - \hat{K}^{\hat{t}}(w;|w|)| \leq c.$$

If a set X has a particular property then we will consider x as having the same named property. For example, a sequence x is recursive if and only if X is recursive. Of particular interest in this paper are recursively enumerable sequences and retraceable sequences (see Rogers [21] for definitions of the corresponding sets). If x is retraceable and φ is a partial recursive retracing function for x (i.e., φ retraces the 1's of x) then we call $\{p_1, \ldots, p_k\}$ a *φ-retraced sequence from n* if and only if $p_1 = m$ and $\forall j < k \blacksquare p_{j+1} = \varphi(p_j)$. Without loss of generality we can restrict retracing functions to be decreasing, i.e., $\varphi(n) \leq n$, and we will do so.

Let φ_e be defined by,

$$\varphi_e(w, m) = \begin{cases} w(m), & \text{if } m \leq |w|, \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Since $\varphi_{\sigma(e,w)}(m) = \varphi_e(w, m)$, $\sigma(e, w)$ is always a program for w and we set $\tau_0(w) = \sigma(e, w)$ and $c_0 = \delta(e)$. Let $t_0(n) = \max\{\Phi_{\tau_0(w)}(m) \mid m \leq n, \ |w| = n\}$. Clearly, $t_0 \in \mathscr{F}$, and we have,

Theorem 1.1. (a) $\forall w \bullet K^{t_0}(w\,; |w|) \leq |w| + c_0$,
(b) $\forall w \bullet K(w\,; |w|) \leq |w| + c_0$.

The following theorems which can be found in Barzdin [1], Daley [6], Kolmogorov [14] and Loveland [16], will be useful in subsequent sections of this paper.

Theorem 1.2. (a) *x is recursive* $\iff \exists c\, \exists t \in \mathscr{F} \bullet \forall n \bullet K^t(x^n; n) \leq c$.
(b) *x is recursive* $\iff \exists c\, \forall n \bullet K(x^n; n) \leq c$.

Theorem 1.3. (a) *If x is recursively enumerable then*

$$\exists c\, \forall n \bullet K(x^n; n) \leq \log_2 n + c.$$

(b) *There exists a recursively enumerable sequence z_1 such that*

$$\forall t \in \mathscr{F} \bullet \overset{\infty}{\exists} n \bullet K^t(z_1^n; n) > \frac{n}{2}.$$

Theorem 1.4. (a) *If x is retraceable then* $\exists c\, \forall n \bullet K(x^n; n) \leq \log_2 n + c$.
(b) *There exists a retraceable sequence z_2 such that*

$$\forall t \in \mathscr{F} \bullet \overset{\infty}{\exists} n \bullet K^t(z_2^n; n) > \frac{n}{2}.$$

We conclude this section with some additional notation. We denote the complement of the set X (sequence x) by $\bar{X}$ ($\bar{x}$). We use $\varphi_i(n)\downarrow$ to denote the fact that $\varphi_i(n)$ is defined, and $\varphi_i(n)\uparrow$ that $\varphi_i(n)$ is undefined. We use σ_1 to denote a program for the sequence of all 1's, i.e., $\forall n \bullet \varphi_{\sigma_1}(n) = 1$.

2. Inferrable sequences and their minimal-program complexity

In this section we give a precise formulation of the inference problem discussed in the preceding section and investigate the properties of sequences whose inference problem is solvable by "inference devices". We call an *inference device* any total recursive function ψ such that for every string w, $\psi(w)$ is a program for w, i.e., $\forall m \leq |w| \, \varphi_{\psi(w)}(m) = w(m)$. Thus, given some data x^n an inference device ψ always formulates some law for x^n, though not necessarily the most succinct one. For example, recalling that $\tau_0(w)$ is a program for w which computes w essentially by table look-up[1], if we let $\psi_0(w) = \tau_0(w)$, then ψ_0 is an inference device albeit a rather trivial one in the sense that given w as input ψ_0 gives as output w in tabular form. Compare this to the situation where a scientist might publish his experimental results without comment in some journal. The set of all inference devices is denoted by Ψ. An inference device ψ is called *frugal* if and only if

$$\forall w \, |\psi(w)| \leq K^{t_\psi}(w; |w|), \tag{2.1}$$

where $t_\psi(n) = \max\{\Phi_{\psi(w)}(m) \mid m \leq n, \ |w| = n\}$. The value $t_\psi(n)$ represents the maximum time required to verify that $\psi(w)$ is always a program for w for any string w of length n. Clearly, $t_\psi \in \mathcal{F}$ for each $\psi \in \Psi$. A frugal inference device is one for which the degree to which it approaches minimal-programs for its inputs is related to how much effort it devotes to the task. Thus ψ_0 above would be regarded as frugal only if ψ_0 spent little effort in determining its outputs.

We say that a partial recursive function φ *infers the optimal description of a sequence* x if and only if for each n, $\varphi(x^n)$ is a program for x^n and

$$\exists c \, \forall n \, |\varphi(x^n)| \leq K(x^n; n) + c. \tag{2.2}$$

We let $\mathcal{O}_\varphi$ denote the set of all sequences x whose optimal descriptions are inferrable by φ, and let $\mathcal{O} = \bigcup_{\psi \in \Psi} \mathcal{O}_\psi$. We remark that $\mathcal{O}$ is clearly closed under complements. The constant c in (2.2) is necessary to insure that whether or not a sequence belongs to $\mathcal{O}$ does not depend on the choice of the u.p.s $\{\varphi_i\}$. It is clear that whether or not a particular total recursive function belongs to Ψ does depend on the choice of $\{\varphi_i\}$. So if $\{\hat{\varphi}_i\}$ is some other u.p.s. we denote by $\hat{\Psi}$ the corresponding set of inference devices. We have the following lemma.

Lemma 2.1. (a) *For every u.p.s.* $\{\varphi_i\}$ *and every* $\psi \in \Psi$ *and every constant* c *there exists a u.p.s.* $\{\hat{\varphi}_i\}$ *and a* $\tilde{\psi} \in \hat{\Psi}$ *such that for every sequence* x, *if* $\forall n \, |\psi(x^n)| \leq K(x^n; n) + c$ *then* $\forall n \, |\tilde{\psi}(x^n)| = \hat{K}(x^n; n)$.

(b) *For every u.p.s.* $\{\varphi_i\}$ *and every* $\psi \in \Psi$ *and every constant* c *there exists a u.p.s.* $\{\hat{\varphi}_i\}$ *such that for every sequence* x *for which* $\forall n \, |\psi(x^n)| = K(x^n; n)$ *the following hold*

(i) $\forall \tilde{\psi} \in \hat{\Psi} \, \overset{\infty}{\exists} n \, |\tilde{\psi}(x^n)| > \hat{K}(x^n; n) + c$, *and*

(ii) $\exists \tilde{\psi} \in \hat{\Psi} \, \exists \tilde{c} \, \forall n \, |\tilde{\psi}(x^n)| \leq \hat{K}(x^n; n) + \tilde{c}$.

Proof. (a) Let $\{\varphi_i\}$, ψ and c be given. Define

[1] This is doubtless an overstatement in as much as τ_0 may convolute the string w in a very complicated way (e.g. by hash coding). What is important here is that every string w can be retrieved from $\tau_0(w)$.

$$\varphi_{0i}(n)=\begin{cases}\varphi_i(n), & \text{if } \psi(\varphi_i)\cdots\varphi_i(n))=i,\\ \text{undefined}, & \text{otherwise},\end{cases}$$

where $0i$ denotes the concatenation of 0 and i, and

$$\hat{\varphi}_{\underbrace{1\cdots 1}_{c+1}0i}(n)=\varphi_i(n).$$

and $\hat{\varphi}_j$ is undefined in all other cases. Define $\tilde{\psi}(w)=0\psi(w)$. The desired conclusion is obvious.

(b) Given $\{\varphi_i\}$, ψ, and c, let r be a total recursive function such that $\forall n\,\exists^\infty m\,\blacksquare\, r(m)=n$, and define

$$\hat{\varphi}_{00i}(n)=\begin{cases}\text{undefined}, & \text{if } \varphi_{r(n)}(\varphi_i(1)\cdots\varphi_i(n))=00i\\ \varphi_i(n), & \text{otherwise},\end{cases}$$

$$\hat{\varphi}_{10i}(n)=\begin{cases}\text{undefined}, & \text{if } \varphi_{r(n)}(\varphi_i(1)\cdots\varphi_i(n))=10i\\ \varphi_i(n), & \text{otherwise},\end{cases}$$

$$\hat{\varphi}_{\underbrace{1\cdots 1}_{c+2}0i}(n)=\varphi_i(n).$$

In all other cases φ_j is undefined. For all i and n either $\hat{\varphi}_{00i}(n)=\varphi_i(n)$ or $\hat{\varphi}_{10i}(n)=\varphi_i(n)$ so that for all w, $\hat{K}(w;|w|)\leq K(w;|w|)+1$. Let $\tilde{\psi}\in\hat{\Psi}$ and let $\varphi_i=\tilde{\psi}$. Then for any x and any n such that $r(n)=i$, $|\tilde{\psi}(x^n)|>\hat{K}(x^n;n)+c$. Then (i) is satisfied. To see that (ii) holds choose $\tilde{\psi}$ so that

$$\tilde{\psi}(w)=\underbrace{1\cdots 1}_{c+2}0\psi(w).\quad\square$$

Our first task is to characterize the set $\mathcal{O}$. We say that a sequence x is *practic* if and only if there exists a total recursive function t such that

$$\exists c\,\forall n\,\blacksquare\, K^t(x^n;n)\leq K(x^n;n)+c. \tag{2.3}$$

In case (2.3) holds we also say that x is t-*practic*. In view of Theorem 1.2 clearly all recursive sequences are practic. If x is a non-recursive sequence then the set of lengths of minimal programs for its initial segments is unbounded. In other words, in order to compute the initial segments of x we must change programs infinitely many times, thereby obtaining an infinite sequence of minimal programs for the successively larger initial segments. A non-recursive practic sequence then is one for which there exists a sequence of (to within c) minimal programs for x which represents a reasonable way of computing the initial segments in the sense that there is an a priori total recursive upper bound t on the computation times of these programs. Let $\mathscr{P}_t$ denote the set of t-practic sequences and let $\mathscr{P} = \bigcup_{t \in \mathscr{F}} \mathscr{P}_t$. As in the case of the definition of $\mathscr{O}_\psi$, the constant c in eq. 2.3 is used to insure that whether or not x belongs to $\mathscr{P}$ is independent of the choice of u.p.s., and we have the following lemma which is proved in a manner analogous to the proof of Lemma 2.1.

Lemma 2.2. (a) *For every u.p.s.* $\{\varphi_i\}$ *and every* $t \in \mathscr{F}$ *and every constant* c *there exists a u.p.s.* $\{\hat{\varphi}_i\}$ *and a* $\tilde{t} \in \mathscr{F}$ *such that for every sequence* x, *if*

$$\forall n_{\blacksquare} K^t(x^n; n) \leq K(x^n; n) + c \quad \textit{then} \quad \forall n_{\blacksquare} \hat{K}^{\tilde{t}}(x^n; n) = \hat{K}(x^n; n).$$

(b) *For every u.p.s.* $\{\varphi_i\}$ *and every* $t \in \mathscr{F}$ *and every constant* c *there exists a u.p.s.* $\{\hat{\varphi}_i\}$ *such that for every sequence* x *for which* $\forall n_{\blacksquare} K^t(x^n; n) = K(x^n; n)$ *the following hold*

(i) $\forall \tilde{t} \in \mathscr{F}_{\blacksquare} \exists^\infty n_{\blacksquare} \hat{K}^{\tilde{t}}(x^n; n) > \hat{K}(x^n; n) + c$, *and*
(ii) $\exists \tilde{t} \in \mathscr{F}_{\blacksquare} \exists \tilde{c}\, \forall n_{\blacksquare} K^{\tilde{t}}(x^n; n) \leq \hat{K}(x^n; n) + \tilde{c}$.

We note that $\hat{K}^t(x^n; n)$ depends on both $\{\hat{\varphi}_i\}$ and $\{\hat{\Phi}_i\}$, but given any $\{\varphi_i\}$ and $\{\Phi_i\}$, a complexity measure $\{\hat{\Phi}_i\}$ for $\{\hat{\varphi}_i\}$ is induced by $\{\Phi_i\}$ and the recursive isomorphism between $\{\varphi_i\}$ and $\{\hat{\varphi}_i\}$. The following theorem gives a characterization of the class $\mathscr{O}$.

Theorem 2.1. (a) $\mathscr{O} = \mathscr{P}$.
(b) $\forall t \in \mathscr{F}_{\blacksquare} t \geq t_0 \Longrightarrow \exists$frugal $\psi \in \Psi_{\blacksquare} \mathscr{O}_\psi = \mathscr{P}_t$.
(c) $\forall$frugal $\psi \in \Psi_{\blacksquare} \exists t \in \mathscr{F}_{\blacksquare} \mathscr{O}_\psi = \mathscr{P}_t$.

Proof. (a) *Case* 1: $\mathscr{P} \subseteq \mathscr{O}$. Suppose $x \in \mathscr{P}$. Then $\exists t \in \mathscr{F}$ such that $x \in \mathscr{P}_t$. Define

$$\psi(w) = \begin{cases} \min\{i \mid |i| \leq |w| + c_0 \ \text{ and } \ \forall m \leq |w|_{\blacksquare} \varphi_i(m) = w(m) \ \text{ and} \\ \qquad \Phi_i(m) \leq t(n)\}, \ \text{ if such an } i \text{ exists,} \\ \psi_0(w), \ \text{ otherwise.} \end{cases} \tag{2.4}$$

Clearly ψ is total recursive and $\psi(w)$ is always a program for w so that $\psi \in \Psi$. Also for each w

$$|\psi(w)| \leq K^t(w; |w|), \tag{2.5}$$

and since $x \in \mathscr{P}_t$, combining (2.3) and (2.5) we have $x \in \mathscr{O}_\psi$.
Case 2: $\mathscr{O} \subseteq \mathscr{P}$. Suppose $x \in \mathscr{O}$. Then there is a $\psi \in \Psi$ and a constant c such that

$$\forall n_{\blacksquare} |\psi(x^n)| \leq K(x^n; n) + c. \tag{2.6}$$

Also, $\forall n \forall m \leq n_{\blacksquare} \Phi_{\psi(x^n)}(m) \leq t_\psi(n)$, so that

$$\forall n_{\blacksquare} K^{t_\psi}(x^n; n) \leq |\psi(x^n)|. \tag{2.7}$$

Combining (2.6) and (2.7) we have $x \in \mathscr{P}_{t_\psi}$.

(b) Let $t \in \mathcal{F}$ be such that $t \geqslant t_0$. Considering the function ψ defined by (2.4) we see that $t_\psi \leqslant \max\{t, t_0\} = t$. Thus $K^{t_\psi}(w; |w|) \geqslant K^t(w; |w|)$. Therefore in view of (2.5) we have that ψ is frugal. Clearly $\mathcal{P}_t \subseteq \mathcal{O}_\psi$ by Case 1 above. Also by Case 2 above we have $\mathcal{O}_\psi \subseteq \mathcal{P}_{t_\psi}$. But $\mathcal{P}_{t_\psi} \subseteq \mathcal{P}_t$ since $t \geqslant t_\psi$. Hence $\mathcal{O}_\psi = \mathcal{P}_t$.

(c) By Case 2 above we have $\mathcal{O}_\psi \subseteq \mathcal{P}_{t_\psi}$. If $x \in \mathcal{P}_{t_\psi}$ and ψ is frugal then $\exists c\, \forall n\, |\psi(x^n)| \leqslant K^{t_\psi}(x^n; n) \leqslant K(x^n; n) + c$. Therefore $x \in \mathcal{O}_\psi$ and hence $\mathcal{O}_\psi = \mathcal{P}_{t_\psi}$. □

Theorem 2.1 seems to imply that there is a threshold factor involved in the manner in which a device does its inferring, viz., it runs all programs of an appropriate size for a predetermined amount of time (the threshold) and selects the shortest program which halts within that time and agrees with the input string. In view of Theorem 2.1 and the fact that all recursive sequences are practic we immediately have.

Theorem 2.2. *If x is recursive then $x \in \mathcal{O}$.*

It is perhaps instructive to establish this result directly in terms of inference devices. Let x be recursive and let $\varphi_{i_x}(n) = x(n)$. We define a ψ_x such that $x \in \mathcal{O}_{\psi_x}$ as follows:

$$\psi_x(w) = \begin{cases} i_x, & \text{if } w = \varphi_{i_x}(1)\cdots\varphi_{i_x}(|w|), \\ \psi_0(w), & \text{otherwise.} \end{cases}$$

We now turn our attention toward the construction of non-recursive sequences belonging to $\mathcal{O}$ and an investigation of their minimal-program complexity. We begin with a particular example of such a sequence which has several interesting properties. First define

$$M_n = \max\{\Phi_i(1) \mid |i| \leqslant n \text{ and } \varphi_i(1)\downarrow\}$$

$$\kappa_n = \text{cardinality of } \{\Phi_i(1) \mid |i| \leqslant n \text{ and } \varphi_i(1)\downarrow\}$$

and choose π_n to satisfy

(i) $|\pi_n| \leqslant n$,

(ii) $\Phi_{\pi_n}(1) = M_n$.

Then define the set X_* by

$$X_* = \{M_n \mid n \in N\}.$$

The set X_* is clearly related to the set of "busy beaver" numbers first studied by Rado [20] (more precisely it is related to the set of shift numbers). The proof of the following theorem may be found in [7] and is similar to several proofs below and so will not be reproduced here.

Theorem 2.3. (a) X_* *is non-recursive.*
(b) X_* *is retraceable.*
(c) $\bar{X}_*$ *is recursively enumerable.*
(d) $x_* \in \mathcal{O}$.
(e) *If* $X \subseteq X_*$ *then* $\exists t \in \mathcal{F}_\blacksquare \forall f \in \mathcal{F}_\blacksquare x \in \mathscr{C}[f \mid t]$.

Referring to (e) of Theorem 2.3 we see that the initial segments of x_* can be computed by arbitrarily (in an effective sense) short programs which run very quickly. Indeed, depending on the complexity measure $\{\Phi_i\}$ (e.g., Turing machine space or time) the function t can be chosen to be linear or near linear. Additional properties of x_* are described in [7] and [12]. Let $\mathscr{C}_{\text{low}}$ denote the complexity class described in (e) of Theorem 2.3 so that $\mathscr{C}_{\text{low}} = \bigcup_{t \in \mathcal{F}} \bigcap_{f \in \mathcal{F}} \mathscr{C}[f \mid t]$. We now show that membership in $\mathscr{C}_{\text{low}}$ is not a sufficient condition for membership in $\mathcal{O}$.

Theorem 2.4. *There exist a recursively enumerable sequence* x *and a retraceable sequence* y *such that* $x, y \in \mathscr{C}_{low} - \mathcal{O}$.

Proof. We first construct a non-recursive sequence z_* which is a variant of the sequence x_*. In fact, z_* will be retraceable with recursively enumerable complement and if $Z \subseteq Z_*$ then $z \in \mathscr{C}_{\text{low}}$. We define the partial recursive function φ_e such that $\varphi_e(i, n) = \varphi_i(n)$ and

$$\Phi_{\sigma(e,i)}(n) \geq 3^{2 \cdot \Phi_i(n)} \cdot 5^{2^{|i|}}. \tag{2.8}$$

Let $d = \delta(e)$, so that

$$|\sigma(e, i)| = |i| + d. \tag{2.9}$$

Define,

$$p_m = M_{m \cdot d},$$

$$q_{m,n} = 3^{p_m} 5^n \quad \text{for } 1 \leq n \leq 2^m,$$

$$Z_* = \{q_{m,n} \mid m \geq 1,\ 1 \leq n \leq 2^m\}.$$

Clearly Z_* is non-recursive. By virtue of (2.8) and (2.9) we have that $q_{m,2^m} < p_{m+1} < q_{m+1,1}$. Now $q_{m,n-1}$ is computable from $q_{m,n}$ and $q_{m,2^m}$ is computable from $q_{m+1,1}$ (since $q_{m,2^m}$ is computable from p_m, which in turn is computable from p_{m+1}, which is computable from $q_{m+1,1}$). In this way we see that Z_* is retraceable. Given $\pi_{m \cdot d}$ we can compute $p_1, \ldots, p_m$ and hence all $q_{i,j}$ for $i \leq m$, $j \leq 2^i$. Furthermore, we can effectively bound in terms of $M_{m \cdot d}$ how long these computations take. Therefore, letting $m_n = \max\{m \mid p_m \leq n\}$, we have

$$\exists c\, \exists t \in \mathcal{F}_\blacksquare \forall n_\blacksquare K^t(z_*^n; n) \leq m_n \cdot d + c. \tag{2.10}$$

From this and the fact that

$$\forall f \in \mathcal{F}_\blacksquare \overset{\infty}{\forall} m_\blacksquare\, p_m > f(m), \tag{2.11}$$

we conclude that $z_* \in \mathscr{C}_{\text{low}}$. We obtain the same conclusion for any subset Z of Z_* since we can specify all those members of Z_* which are $\leq n$ and which belong to Z by a string of length $\leq 2^{m_n+1}$, and by virtue of (2.11) the addition of such a string to the program for z_*^n would not result in a program of appreciably larger size relative to n.

We first construct the sequence x. Let z_1 be the recursively enumerable sequence of Theorem 1.3 b, so that $\forall t \in \mathscr{F}_\blacksquare \exists^\infty n_\blacksquare K^t(z_1^n; n) > n/2$. Define the set X by $n \notin X \iff \exists m_\blacksquare n$ is the m^{th} member of Z_* and $m \notin Z_1$. Clearly x is non-recursive and, since $\bar{X} \subseteq Z_*$ and $\mathscr{C}_{\text{low}}$ is obviously closed under complements, $x \in \mathscr{C}_{\text{low}}$.

Consider x^n. Recall that $m_n = \max\{m \mid p_m \leq n\}$. There are at most 2^{m_n+2} members of Z_* which are $\leq n$, each of which can be computed from p_{m_n}. Thus given Z_*^n and $Z_1^{2^{m_n+1}}$ we can compute x^n. Therefore, it follows from (2.10) and Theorem 1.3(a) that

$$\exists c\, \forall n_\blacksquare K(x^n; n) \leq c \cdot m_n. \tag{2.12}$$

Now suppose $\exists c\, \exists t \in \mathscr{F}_\blacksquare \forall n_\blacksquare K^t(x^n; n) \leq c \cdot m_n$. Let $n = 2^i + j - 2$, where $i \geq 1$, $1 \leq j \leq 2^i$. Then by the definition of z_1, $z_1^n = x(q_{1,1}) \cdots x(q_{i,j})$, so that z_1^n is computable from $x^{p_{i+1}}$ and $z_*^{p_{i+1}}$. But then from (2.10) and our supposition concerning x it follows that $\exists c\, \exists t \in \mathscr{F}_\blacksquare \forall n_\blacksquare K^t(z_1^n; n) \leq c \cdot \log_2 n$, which contradicts Theorem 1.3(b). Hence $\forall c\, \forall t \in \mathscr{F}_\blacksquare \exists^\infty n_\blacksquare K^t(x^n; n) > c \cdot m_n$, which together with (2.12) proves that $x \notin \mathscr{O}$.

To show that x is recursively enumerable we describe a procedure which enumerates the members of X in stages. At successive stages progressively larger portions of X will be enumerated. The procedure uses a list of natural numbers and infinitely many markers which are moved down the list of integers. There are two types of markers: p-markers $\boxed{m}$ and q-markers $\boxed{m \mid n}$, for $1 \leq n \leq 2^m$. Marker $\boxed{m}$ is associated with the set of computations $\{\varphi_i(1) \mid |i| \leq m \cdot d\}$ and its final resting place will be p_m. The final resting place of marker $\boxed{m \mid n}$ is $q_{m,n}$. At any particular stage, if $\boxed{m}$ is adjacent to integer k then marker $\boxed{m \mid n}$ will be adjacent to integer $3^k \cdot 5^n$. In addition, for each marker there are two colors possible: white and black. Initially, all q-markers are white and all p-markers are black, and once a marker is colored black it remains black. The final resting place of a black marker will be put into X while that of a white marker will not. The markers are ordered according to the ordering of their corresponding members of Z_*, i.e.,

$$\cdots \boxed{m} < \boxed{m \mid 1} < \boxed{m \mid 2} < \cdots < \boxed{m \mid 2^m} < \boxed{m+1} \cdots$$

Let h be a total recursive function which enumerates the members of Z_1 in a one-to-one fashion. We now give the description of stage n of the procedure.

Stage n. (a) Place marker $\boxed{n}$ next to integer n and markers $\boxed{n \mid i}$, $1 \leq i \leq 2^n$ next to integer $3^n \cdot 5^i$.

(b) Find the program j of least size $\leq n$ such that $\Phi_j(1) = n$. If no such j exists go to (c). Otherwise, let $m = |j|$. Move all markers $\boxed{m}$, $\boxed{m+1}, \ldots, \boxed{n-1}$ down to integer n on the list and markers $\boxed{k \mid i}$, $m \leq k \leq n-1$, $1 \leq i \leq 2^k$, down to integer $3^n \cdot 5^i$.

(c) Color the $h(n)^{\text{th}}$ q-marker black.

(d) Enumerate all integers $\leq n$ which are not adjacent to a white marker, i.e., put them into X. Go to stage $n+1$. (Note that some integers $\leq n$ may be adjacent to both a white marker and a black marker. Such integers are not put into X at this stage.)

An examination of the above procedure reveals that in fact the final resting places of all the q-markers are the members of Z_* and that of the white markers are the members of $\bar{X}$.

We now construct the sequence y. Let z_2 be the retraceable sequence of Theorem 1.4(b), so that $\forall t \in \mathscr{F}_\blacksquare \exists^\infty n_\blacksquare K'(z_2^n; n) > n/2$. We define the set Y by,

$$n \in Y \iff \exists m_\blacksquare n \text{ is the } m^{\text{th}} \text{ member of } Z_* \quad \text{and} \quad m \in Z_2.$$

Clearly Y is non-recursive and since $Y \subseteq Z_*$, $y \in \mathscr{C}_{\text{low}}$. The proof that $y \notin \mathcal{O}$ is the same as that for $x \notin \mathcal{O}$. The retraceability of Y follows from the retraceability of Z_* and Z_2. $\square$

The next theorem shows that membership in $\mathscr{C}_{\text{low}}$ is not necessary for membership in $\mathcal{O}$. Let $\mathscr{C}_{\log} = \{x \mid \forall c\, \forall^\infty n_\blacksquare K(x^n; n) \leq \log_2 n - c\}$.

Theorem 2.5. *There exists a recursively enumerable sequence x and a retraceable sequence y such that $x, y \in \mathcal{O} - \mathscr{C}_{\log}$.*

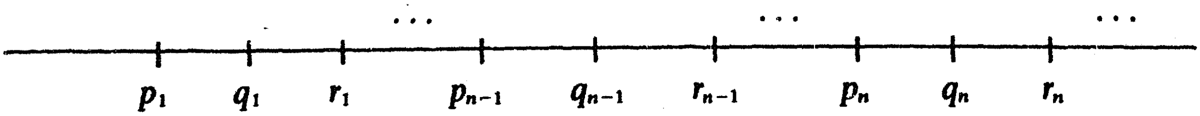
Proof. Let p_1 be so large that for each number $p > p_1$, $M_{\log_2(p)} > 4 \cdot p$. This will insure that $s_n > r_n$ below. Let $s_1 = p_1$ and define

$$\begin{aligned}
p_n &= 4^{s_{n-1}} \\
s_n &= M_{\log_2(p_n)} \\
k_n &= \kappa_{\log_2(p_n)} \\
q_n &= p_n + k_n, \\
r_n &= 3\ p_n.
\end{aligned}$$

Since $k_n \leq 2^{\log_2 p_n + 1} = 2 \cdot p_n$, then $q_n \leq r_n$. Observe that p_n is computable from s_n. We define the sequence x by

$$x(m) = \begin{cases} 1 & \text{if } \exists n_\blacksquare r_{n-1} < m < q_n, \\ 0 & \text{if } \exists n_\blacksquare q_n \leq m \leq r_n. \end{cases}$$

The basic idea in the construction of x is fairly simple although the necessary details may tend to obscure it. We therefore first give a sketch of the proof that $x \in \mathcal{O}$. Referring to the diagram below we see that x is defined in segments:



Let us consider the n^{th} segment (i.e., the interval $(r_{n-1}, r_n]$). Notice that p_n and q_n are computable from k_n, and that k_{n-1} is computable from s_{n-1} and hence from k_n. Thus x^{r_n} and hence $x^{q_{n+1}-1}$ can be computed from k_n. Since $p_n > s_{n-1}$ we can effectively bound how long such a computation will take. Thus for some $t \in \mathcal{F}$ and some constant c_1,

$$K^t(x^{r_n}; r_n) \le \log_2(k_n) + c_1 \le \log_2(p_n) + c_1.$$

To show that $x \in \mathcal{O}$ it suffices to show that for some constant c_2,

$$K(x^{q_n}; q_n) > \log_2(p_n) - c_2.$$

Suppose to the contrary that there is a program i for x^{q_n} such that $|i| \le \log_2(p_n) - c$ for a sufficiently large value of c. Then it is possible to transform i into a program j (see $\varphi(e, i, m)$ below) such that $|j| \le \log_2(p_n)$ and that j refutes $k_n = q_n - p_n$ in the sense that $\varphi_j(i)\downarrow \iff j$ is not among the first $q_n - p_n$ programs of size $\le \log_2(p_n)$ which halt on input 1. It is essential here that p_n be computable from q_n.

The function φ is defined by,

$\varphi(e, i, 1)$: Compute $\varphi_i(1), \varphi_i(2), \ldots$ until the least q is found satisfying

(i) $\forall n \le q \cdot \varphi_i(n)\downarrow$,

(ii) $\varphi_i(q) = 0$ and $\varphi_i(q-1) = 1$,

(iii) $|\sigma(e, i)| \le \log_2 p$, where p is the unique number of the form 4^s such that $4^s \le q < 4^{s+1}$. This completes the definition of φ.

Let $k = q - p$. Compute the finite set $J = \{$first k programs of length $\le \log_2(p)$ which halt on input 1$\}$. If $\sigma(e, i) \in J$ then enter an infinite loop. Otherwise output the value 0 and halt.

Choose an e_0 such that $\varphi_{e_0}(i, m) = \varphi(e_0, i, m)$. If $K(x^{q_n}; q_n) < \log_2(p_n) - \delta(e_0)$ then there exists a program i such that $|i| < \log_2(p_n) - \delta(e_0)$ and $\forall m \le q_n \cdot \varphi_i(m) = x(m)$. From the definition of $\varphi(e_0, i, m)$ we see that $q = q_j$, $p = p_j$ and $k = k_j$ where j is the least number $\le n$ with $|\sigma(e_0, i)| \le |i| + \delta(e_0) \le \log(p_j)$. From this it is also clear that $\varphi_{\sigma(e_0, i)}(1)\downarrow \iff \sigma(e_0, i) \notin J$, which implies that $q_j - p_j \ne k_j$, which is an immediate contradiction. Thus we have $\forall n \cdot K(x^{q_n}; q_n) \ge \log_2(p_n) - \delta(e_0)$, and thus

$$\forall n\, \forall m \ge q_n \cdot K(x^n; m) \ge \log_2(p_n) - \delta(e_0) \ge \log_2(r_n) - \delta(e_0) - 2. \tag{2.13}$$

Consequently, $x \notin \mathscr{C}_{\log}$. Given k_n and using the respective defining equations we can compute p_j, q_j, r_j and k_j for all $j \le n$. Moreover, we can recursively bound the amount of time required to compute these numbers in terms of k_n, since $\Phi_i(1) \le p_n$ for all i such that $|i| \le \log(p_{n-1})$. Thus we conclude

$$\exists c\, \exists t \in \mathscr{F}_{\blacksquare} \forall n\, \forall m < q_{n+1} \blacksquare\, K^t(x^m; m) < |k_n| + c \leq \log_2(r_n) + c. \tag{2.14}$$

Combining (2.13) and (2.14) we have that $x \in \mathcal{O}$.

To see that x is recursively enumerable we give the following procedure using markers which enumerates the members of X. There is an infinite number of pairs of markers $\boxed{m|1}$, $\boxed{m|2}$, and the final resting-places of $\boxed{m|1}$ and $\boxed{m|2}$ are p_m and q_m respectively. We now describe stage n of the procedure. We use $\langle m, 1\rangle$ and $\langle m, 2\rangle$ to denote the current positions of $\boxed{m|1}$ and $\boxed{m|2}$ respectively.

Stage n. Find the program j of least size $m \leq n$ such that $\Phi_j(1) = n$. Place integer $\langle m, 2\rangle$ into X and move marker $\boxed{m|2}$ to integer $\langle m, 2\rangle + 1$. Move all markers $\boxed{m+1|1}$, $\boxed{m+1|2}, \ldots, \boxed{n|1}$, $\boxed{n|2}$ next to integer 4^n, and place into X all integers k such that $3 \cdot \langle m, 1\rangle \leq k < 4^n$.

That this procedure enumerates the members of X is easily seen. We define the sequence y by the condition $Y = \{q_n \mid n \geq 1\}$. Clearly, y is retraceable and using the same proofs used for x we obtain $y \in \mathcal{O}$ and $y \notin \mathscr{C}_{\log}$. □

The results in Theorem 2.5 are the best possible for recursively enumerable and retraceable sequences in view of Theorem 1.3a and Theorem 1.4a. Thus we see that although there is a characterization of sequences in $\mathcal{O}$ in terms of minimal-program complexity notions, there does not seem to be any relationship between sequences in $\mathcal{O}$ and their actual minimal-program complexity.

We now take up the question of whether the order of the experimentation (and by this we understand an effective and complete enumeration of experimental data points) can affect our ability to infer the optimal description of a phenomenon. In other words is the set $\mathcal{O}$ closed under recursive permutations. Certainly every recursive permutation of a recursive sequence (being itself recursive) belongs to $\mathcal{O}$. Moreover, as we shall see below there are many non-recursive sequences with this property as well. However, it does seem plausible that there is some phenomenon, although none has been constructed here, whose inferrability is sensitive to order of presentation.

Theorem 2.6. *Every recursive permutation of x_* belongs to $\mathcal{O}$.*

Proof. This proof is similar to the proof given in [7] that x_* belongs to $\mathcal{O}$. Let f be a recursive permutation and let y be such that $y(f(n)) = x(n)$. We show that

(i) $\exists c_1 \blacksquare \exists t \in \mathscr{F}_{\blacksquare} \forall n_{\blacksquare} K^t(y^n; n) \leq m_n + c_1$,

(ii) $\exists c_2 \forall n_{\blacksquare} K(y^n; n) > m_n - c_2$,

where $m_n = \max\{k \mid f(M_k) \leq n\}$.

Given π_{m_n} (recall that $|\pi_k| \leq k$ and $\Phi_{\pi_k}(1) = M_k$), we can compute M_{m_n} by running π_{m_n} until it halts, and this takes time relative to n of the order of $\max\{f^{-1}(k) \mid k \leq n\}$. From M_{m_n} we can compute all M_k for $k \leq m_n$ in similar amounts of time relative to n and hence we can determine all $k \leq n$ for which $y(k)=1$. In other words we can compute y^n from π_{m_n} in time $t(n)$, for an appropriately chosen recursive t. Thus (i) is satisfied.

We can transform any (such) program π into a program π' such that $|\pi'| \leq |\pi| + c^*$ (where c^* is independent of π) which first uses π to find the least p such that $\varphi_\pi(p)=1$ and $\min\{|k| \mid \Phi_k(1)=f^{-1}(p)\} \geq |\pi'|$, and then loops until $\Phi_{\pi'}(1) > f^{-1}(p)$. Such a program transformation clearly involves a use of the Recursion Theorem. Suppose now there exists an n and a program π for y^n such that $|\pi| \leq m_n - c^*$. Then using the above transformation we construct a program π' with $|\pi'| \leq m_n$ such that $\Phi_{\pi'}(1) > M_{|\pi'|}$ which contradicts the fact that π computes y^n. Thus (ii) must hold and therefore $y \in \mathscr{P}$ and so $y \in \mathcal{O}$. □

In [12] an infinite retraceable subset X_0 of X_* is constructed such that if $Y \subseteq X_0$ and $\bar{Y}$ is recursively enumerable, then Y is retraceable (in fact, Y is retraced by a total recursive retracing function). By means analagous to those used in Theorem 2.6 we have,

Theorem 2.7. *There exists an infinite subset X_0 of X_* such that if $Y \subseteq X_0$ and $\bar{Y}$ is recursively enumerable then every recursive permutation of y belongs to $\mathcal{O}$.*

Moreover, it is shown in [12] that $\bar{X}_0$ is an atomless recursively enumerable set so that every infinite non-recursive corecursively-enumerable subset Y of X_0 has an infinite non-recursive corecursively-enumerable subset Z, such that $Y-Z$ is infinite.

3. Alternative definitions

In this section we consider some notions which are related to the inference problem described in Section 1 but which represent an alternative approach to that of Section 2. We first consider the situation where arbitrary partial recursive functions instead of inference devices are used to infer optimal descriptions of sequences. To this end we define $\mathcal{O}^* = \bigcup_{\varphi \in \{\varphi_i\}} \mathcal{O}_\varphi$. Clearly, $\mathcal{O} \subseteq \mathcal{O}^*$. We begin by giving sufficient conditions for membership in $\mathcal{O}^*$.

Theorem 3.1. *If $f \in \mathscr{F}$ and x is a sequence satisfying*

$$\exists c\, \forall n\, f(n) \leq K(x^n; n) \leq f(n) + c, \quad \textit{then } x \in \mathcal{O}^*.$$

Proof. Define φ by,

$$\varphi(w) = \min\{i \mid |i| \leq f(|w|) + c \quad \text{and} \quad \forall m \leq |w|\, \varphi_i(m) = w(m)\}.$$

Clearly, $x \in \mathcal{O}_\varphi$. Note that $\varphi(w)\downarrow \iff K(w; |w|) \leq f(|w|) + c$. □

In [6] it was shown that for every $f \in \mathscr{F}$ such that $\forall n_{\blacksquare} f(n) \leq \log_2(n)$ and $\forall n_{\blacksquare} f(n+1) - f(n) \leq 1$ there exists a sequence x such that $\exists c \forall n_{\blacksquare} f(n) \leq K(x^n; n) \leq f(n) + c$. We remark that the condition $\forall n_{\blacksquare} f(n+1) - f(n) \leq 1$ is not a severe restriction in view of the fact that any $f \in \mathscr{F}$ which serves as a non-trivial upper bound for program complexity will satisfy $\forall n_{\blacksquare} f(n) \leq n$. We now give an improvement of the above mentioned result. Let $\mathscr{F}_1$ be the set of all $f \in \mathscr{F}$ satisfying

(i) $\forall n_{\blacksquare}$ cardinality $\{m \mid f(m) = n\} \geq n+1$.

For example, $\lambda n(\lfloor \sqrt{n} \rfloor) \in \mathscr{F}_1$. Condition (i) can be viewed as a smoothness condition. Note that since $\mathscr{F}_1 \subseteq \mathscr{F}$, every $f \in \mathscr{F}_1$ is non decreasing.

Theorem 3.2. *For every $f \in \mathscr{F}_1$ there exists a sequence x such that*

$$\forall n_{\blacksquare} f(n) \leq K(x^n; n) \leq f(n) + c.$$

Proof. Let f be a function satisfying the conditions of the hypothesis of the theorem and let $g \in \mathscr{F}$ be defined by $g(n) = \min\{m \mid f(m) = n\}$. Since $f \in \mathscr{F}_1$, $g(n) - g(n-1) > n+1$. Let w_n be the binary representation of length $g(n) - g(n-1)$ (using the necessary number of high order 0's) of the number κ_n. Such a w_n always exists since there are less than 2^{n+1} programs of length $\leq n$ and there are exactly $2^{g(n)-g(n-1)}$ strings of length $g(n) - g(n-1)$. Define the sequence x by $x(g(n-1)+1) \cdots x(g(n)) = w_n$. By an argument analogous to that used to establish (2.13) we can show that

$$\exists c \forall n \forall m \geq g(n)_{\blacksquare} K(x^m; m) > n - c. \tag{3.1}$$

The number κ_n can be represented by a string of length at most $n+1$ (without the high order 0's), and since κ_{n-1} can be computed from κ_n we have,

$$\exists c \forall n \forall m \leq g(n)_{\blacksquare} K(x^m; m) \leq n + c. \tag{3.2}$$

Combining (3.1) and (3.2) we have $\exists c \forall n \forall m_{\blacksquare} g(n-1) \leq m < g(n) \implies n - c \leq K(x^m; m) \leq n + c$, and since $\forall n \forall m_{\blacksquare} g(n) \leq m < g(n+1) \implies f(m) = n$, we arrive at the desired conclusion $\exists c \forall n_{\blacksquare} f(n) \leq K(x^n; n) \leq f(n) + c$. $\square$

We are now prepared to show that $\mathcal{O} \neq \mathcal{O}^*$. We modify the preceding construction as follows. Let $g(1) = 1$ and $g(n) = g(n-1) + 3n + 2$, and define f by the condition $f(n) = \max\{m \mid g(m) \leq n\}$. Let δ_n be the least string in lexicographical order of length $2n+1$ such that $\delta_n \not\in \{\varphi_i(g(n-1)+n+2) \cdots \varphi_i(g(n)) \mid |i| \leq 2n$ and $\Phi_i(g(n)) \leq M_n\}$. Such a δ_n exists since there are less than 2^{2n+1} programs of length $\leq 2n$. Define x by $x(g(n-1)+1) \cdots x(g(n-1)+n+1) = w_n$ and $x(g(n-1)+n+2) \cdots x(g(n)) = \delta_n$. Since δ_n is computable from κ_n we conclude in the same manner as above that

$$\exists c \forall n_{\blacksquare} f(n) - c \leq K(x^n; n) \leq f(n) + c \tag{3.3}$$

and so $x \in \mathcal{O}^*$. Given any c, n, i and $t \in \mathcal{F}$ such that $c \leq f(n)$ and $t(n) \leq M_{f(n)}$ and $|i| \leq f(n)+c$ and $\Phi_i(m) \leq t(n)$ for all $m \leq n$, from the definition of δ_n we see that $\varphi_i(g(n-1)+n+2)\cdots\varphi_i(g(n)) \neq \delta_n$. Therefore, $\forall c\, \forall t \in \mathcal{F}_{\blacksquare}\, \exists^\infty n_{\blacksquare}\, K'(x^n; n) > f(n)+c$, so that in view of (3.3) we have $x \notin \mathcal{O}$. Hence we have the following

Theorem 3.3. (a) $\mathcal{O} \subsetneq \mathcal{O}^*$.

(b) *There exists a recursively enumerable sequence x such that* $x \in \mathcal{O}^*$ *and* $x \notin \mathcal{O}$.

(c) *There exists a retraceable sequence y such that* $y \in \mathcal{O}^*$ *and* $y \notin \mathcal{O}$.

The sequences x and y in parts (b) and (c) above can be constructed in a straightforward manner by using a combination of several of the preceding constructions.

We now consider two formulations of the inference problem which involve a slightly different notion of program complexity. Let a non-recursive sequence x be fixed for the remainder of this paragraph, and let m and n be integers with n much greater than m, and let p_n, p_m be minimal programs for x^n, x^m, respectively. With respect to the complexity measure K' which we have thus far been considering for p_m we require that $\Phi_{p_m}(m) \leq t(m)$, but for p_n we require only that $\Phi_{p_n}(m) \leq t(n)$. But $\varphi_{p_m}(m) = \varphi_{p_n}(m) = x(m)$. Suppose now that we want $t(m)$ to be an upper bound for the computation of $x(m)$ by a program p regardless of how large an initial segment of x p may compute, i.e. replace the condition $\Phi_i(m) \leq t(|w|)$ in the definition of $K'(w; |w|)$ by the condition $\Phi_i(m) \leq t(m)$. As things stand this would constitute an impossible situation for many computational complexity measures. Indeed, any program for any arbitrarily large initial segment of x (or what amounts to the same thing, any arbitrarily large program) would be required to compute $x(1)$ within time $t(1)$. This would be untenable, for example, in the real life situation where the amount of memory used for a computation includes the memory used to store the program. The way out of this dilemma is to observe that the program p_n contains more information than p_m since it can compute the initial segments of x of length between m and n in addition to x^m. If we take the view that p_n is obtained from p_m by "adding" information to p_m, then it should also be possible to obtain p_m from p_n by "deleting" information from p_n. Combining these notions we arrive at what is in essence the monotonic operator complexity of Levin [15] (see also Zvonkin and Levin [30]) and the process complexity of Schnorr [22].

A u.p.s. $\{\varphi_i\}$ is called *monotonic* if and only if

$$\forall i\, \forall j \sqsupseteq i_{\blacksquare}\, \forall n_{\blacksquare} [\varphi_i(n)\downarrow \quad \text{and} \quad \varphi_j(n)\downarrow] \Longrightarrow \varphi_i(n) = \varphi_j(n),$$

where $v \sqsubseteq w$ or $w \sqsupseteq v$ is used to denote that the string v is an initial segment of the string w. By a computational complexity measure for a monotonic u.p.s. $\{\varphi_i\}$ we will understand any complexity measure $\{\Phi_i\}$ which in addition to satisfying the Blum axioms also satisfies (i) $\Phi_i(n)$ is strictly increasing with respect to i and n and (ii) $\forall i\, \forall n_{\blacksquare}\, \Phi_i(n) \geq |i|$. For a given monotonic u.p.s. $\{\Phi_i\}$ and a computational complexity measure $\{\Phi_i\}$ for $\{\varphi_i\}$ we define

$$Km(w;|w|) = \min\{|j| \mid \forall m \leq |w| \centerdot \exists i \sqsubseteq j \centerdot \varphi_i(m) = w(m)\},$$

$$Km'(w;|w|) = \min\{|j| \mid \forall m \leq |w| \centerdot \exists i \sqsubseteq j \centerdot \varphi_i(m) = w(m) \text{ and } \Phi_i(m) \leq t(m)\}.$$

Let Ψ^m denote the set of inference devices relative to the monotonic complexity, i.e. the set of all total recursive functions ψ such that

$$\forall x\, \forall n\, \forall m \leq n \centerdot \exists i \sqsubseteq \psi(x^n) \centerdot \varphi_i(m) = x(m).$$

We next define

$$\mathcal{O}^m = \{x \mid \exists \psi \in \Psi^m \centerdot \exists c\, \forall n \centerdot |\psi(x^n)| \leq Km(x^n; n) + c\},$$

$$\mathcal{P}^m = \{m \mid \exists t \in \mathcal{F} \centerdot \exists c\, \forall n \centerdot Km'(x^n; n) \leq Km(x^n; n) + c\}.$$

In a manner analogous to the proof of Theorem 2.1 we can show

Theorem 3.4. $\mathcal{P}^m \subseteq \mathcal{O}^m$.

However, the reverse inclusion does not appear to hold, because it seems unlikely that the appropriate time bound t_ψ can be constructed. This could be remedied perhaps by placing more restrictions on Ψ^m. For example we could require that for $\psi \in \Psi^m$ ψ be monotonic, i.e. $u \sqsubseteq v \implies \psi(u) \sqsubseteq \psi(v)$, which would be sufficient to show $\mathcal{O}^m \subseteq \mathcal{P}^m$, but which unfortunately also invalidates the current proof of Theorem 3.4.

A Martin-Löf random sequence (see [17]) is a sequence which in a certain sense satisfies all constructive tests for randomness. In [12] Schnorr showed that a sequence x is Martin–Löf random if and only if $\exists c\, \forall n \centerdot n - c \leq Km(x^n; n) \leq n + c$. Since for sufficiently large $t \in \mathcal{F}$ and every sequence x $\exists c\, \forall n \centerdot Km'(x^n; n) \leq n + c$ and $Km(x^n; n) \leq Km'(x^n; n)$, it immediately follows that every Martin–Löf random sequence belongs to $\mathcal{P}^m$. In more recent work of Levin [16] (see also Schnorr [24]) the notion of generalized Martin–Löf random sequences (i.e., sequences random in the sense of Martin–Löf relative to some computable probability measure) has been developed and it has been shown that a sequence x is generalized Martin–Löf random if and only if $x \in \mathcal{P}^m$. This is a very elegant result. In this setting recursive sequences are random with respect to the trivial probability measure which assigns a 0 probability to a 0. In other words all recursive sequences are recursive transformations of the sequence of all ones.

Our last version of the inference problem represents a weakening or our preceding conditions for the inference of optimal descriptions.
Let

$$\tilde{\mathcal{O}}^m = \{x \mid \exists \psi \in \Psi^m \centerdot \forall f \in \mathcal{F} \centerdot \overset{\infty}{\forall} n \centerdot |\psi(x^n)| \leq Km(x^n; n) + f(n)\}$$

$$\tilde{\mathcal{P}}^m = \{x \mid \exists t \in \mathcal{F} \centerdot \forall f \in \mathcal{F} \centerdot \overset{\infty}{\forall} n \centerdot Km'(x^n; n) \leq Km(x^n; n) + f(n)\}.$$

One has no difficulty in showing that $\tilde{\mathcal{O}}^m \supseteq \tilde{\mathcal{P}}^m$. The interesting feature of the definition of $\tilde{\mathcal{O}}^m$ is that for any x such that $x \in \tilde{\mathcal{O}}^m$ and $x \notin \mathcal{O}^m$ there is an inference device which, while not producing the minimal descriptions of x^n, produces descriptions which one cannot prove to be not minimal. This notion is similar to one considered by Schnorr and Fuchs in [25]. A sequence x is called *learnable* by them if and only if

$$\exists t \in \mathcal{F}_{\blacksquare} \forall f \in \mathcal{F}_{\blacksquare} \forall t' \in \mathcal{F}_{\blacksquare} \overset{\infty}{\forall} n_{\blacksquare} Km^t(x^n; n) \leq Km^{t'}(x^n; n) + f(n).$$

In their paper it was shown that learnable sequences coincide with sequences which are random in a sense due to Schnorr (see [23]) and which is weaker than the above notion of generalized Martin–Löf randomness. It is interesting to note that it was shown in [7] that the sequence x_* is learnable and so is random in Schnorr's sense.

Some open questions in this area are (i) what is the relation between $\tilde{\mathcal{O}}^m$ and Schnorr randomness, or alternately what properties of randomness do sequences in $\tilde{\mathcal{O}}^m$ possess, and (ii) can a characterization of $\mathcal{O}^*$ be found which is analagous to the one given in Theorem 2.1 for $\mathcal{O}$?

Acknowledgements

I wish to thank Gregory Chaitin, Peter Fuchs and Claus Schnorr for their helpful discussions on the subject contained in this paper. Thanks also are due to the referees who pointed out some deficiencies in an earlier version of this paper.

References

[1] J. Barzdin, Complexity of programs to determine whether natural numbers not greater than n belong to a recursively enumerable set, *Soviet Math. Dokl.* **9** (1968) 1251–1254.
[2] A. Bierman and J. Feldman, A Survey of Results in Grammatical Inference, in: *Frontiers of Pattern Recognition* (Academic Press, Inc., NY, 1972) 31–54.
[3] M. Blum, A machine independent theory of the complexity of recursive functions, *J. ACM* **14** (1967) 322–336.
[4] L. Blum and M. Blum, Toward a mathematical theory of inductive inference, *Information and Control* **28** (1975) 125–155.
[5] G. Chaitin, On the length of programs for computing finite binary sequences, *J. ACM* **13** (1966) 547–569.
[6] R. Daley, The extent and density of sequences within the minimal-program complexity hierarchies, *J. Comput. Systems Sci.* **9** (1974) 151–163.
[7] R. Daley, Non-complex sequences: Characterizations and examples, *J. Symbolic Logic* **41** (1976) 626–638.
[8] R. Daley, Equi-informational sequences, Proc. 6th Ann. Princeton Conf. Information Sci. Systems (1972) 585.
[9] R. Daley, On the learning of non-recursive sequences, Proc. 7th Ann. Princeton Conf. Information Sci. Systems (1973) 552–556.

[10] R. Daley, On the inference of optimal descriptions, Proc. Conf. Information Sci. Systems, Baltimore, MD (1975) 121–126.

[11] R. Daley, The Process complexity and the understanding of sequences, Proc. Conf. Math. Found. Comput. Sci., High Tatras, Czechoslovakia (1973) 215–218.

[12] R. Daley, On the simplicity of busy beaver sets, *Zeit. Math. Logik u. Grund. Math.* (to appear).

[13] M. Gold, Language identification in the limit, *Information and Control* **10** (1967) 447–474.

[14] A. Kolmogorov, Three approaches for defining the concept of information quantity, *Information Transmission* **1** (1965) 3–11.

[15] L. Levin, On the Notion of Random Sequence, *Soviet Math. Dokl.* **14** (1973) 1413–1415.

[16] D. Loveland, A variant of the Kolmogorov concept of complexity, *Information and Control* **4** (1969) 510–526.

[17] P. Martin-Löf, The definition of random sequences, *Information and Control* **9** (1966) 603–613.

[18] D. Pager, On the problem of finding minimal-programs for tables, *Information and Control* **14** (1969) 550–554.

[19] D. Pager, Further results on the problem of finding minimal length programs for decision tables, *J. ACM* **21** (1974) 207–212.

[20] T. Rado, On a simple source for non-computable functions, Proc. Symp. Math. Theory Automata, MRI Symposium Series XII, Polytechnic Press, Brooklyn (1962) 75–81.

[21] H. Rogers, *Theory of Recursive Functions and Effective Computability*, (McGraw-Hill, NY, 1967).

[22] C. Schnorr, *Zufälligkeit und Warscheinlichkeit*, Lecture Notes in Mathematics, No. 218 (Springer-Verlag, Berlin, 1971).

[23] C. Schnorr, The process complexity and effective random tests, JCSS **7** (1973) 276–388.

[24] C. Schnorr, A review of the theory of random sequences, Proc. 5th Int. Congress of Logic, Methodology and Philosophy of Science, London, Ontario (August 1975).

[25] C. Schnorr and P. Fuchs, General random sequences and the concept of learnable sequences, preprint, Fachbereich Mathematik, Universitat Frankfurt (1975).

[26] L. Schubert, Iterated limiting recursion and the program minimization problem, *J. ACM* **21** (1974) 436–445.

[27] H. Simon, Does scientific discovery have a logic? *Philosophy of Science* **40** (1973) 471–480.

[28] H. Simon, Complexity and the representation of patterned sequences of symbols, *Psychological Review* **79** (1972) 369–382.

[29] R. Solomonoff, A formal theory of inductive inference, Part I, *Information and Control* **7** (1964) 1–22.

[30] A. Zvonkin and L. Levin, Complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms, *Russian Math. Surveys* **25** (1970) 83–124.