Forum of Mathematics, Sigma (2021), **Vol. 9**:e62 1–7
doi:10.1017/fms.2021.58

CrossMark

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

An incompressibility theorem for automatic complexity

Bjørn Kjos-Hanssen

Department of Mathematics, University of Hawaiʻi at Mānoa, Honolulu, HI 96822, USA;
E-mail: bjoern.kjos-hanssen@hawaii.edu.

Received: 25 February 2021; **Revised:** 1 August 2021; **Accepted:** 19 August 2021

2020 Mathematics Subject Classification: *Primary* – 68Q45; *Secondary* – 68Q30

Abstract
Shallit and Wang showed that the automatic complexity $A(x)$ satisfies $A(x) \geq n/13$ for almost all $x \in \{\mathtt{0}, \mathtt{1}\}^n$. They also stated that Holger Petersen had informed them that the constant 13 can be reduced to 7. Here we show that it can be reduced to $2 + \epsilon$ for any $\epsilon > 0$. The result also applies to nondeterministic automatic complexity $A_N(x)$. In that setting the result is tight inasmuch as $A_N(x) \leq n/2 + 1$ for all x.

1. Introduction

Kolmogorov's structure function for a word x is intended to provide a statistical explanation for x. We focus here on a computable version, the automatic structure function h_x. For definiteness, suppose x is a word over the alphabet $\{\mathtt{0}, \mathtt{1}\}$. By definition, $h_x(m)$ is the minimum number of states of a finite automaton that accepts x and accepts at most 2^m many words of length $|x|$. The *best explanation* for the word x is then an automaton witnessing a value of h_x that is unusually low, compared to values of h_y for most other words y of the same length. To find such explanations we would like to know the distribution of h_x for random x. In the present paper we take a step in this direction by studying the case $h_x(0)$, known as the *automatic complexity* of x.

The automatic complexity of Shallit and Wang [9] is the minimal number of states of an automaton accepting only a given word among its equal-length peers. Finding such an automaton is analogous to the protein-folding problem, where one looks for a minimum-energy configuration. The protein-folding problem may be NP-complete [2], depending on how one formalises it as a mathematical problem. For automatic complexity, the computational complexity is not known, but a certain generalisation to equivalence relations gives an NP-complete decision problem [4].

Here we show (Theorem 18) that automatic complexity has a similar incompressibility phenomenon as that of Kolmogorov complexity for Turing machines, first studied in [6, 7, 11, 12].

1.1. Incompressibility

Let C denote Kolmogorov complexity, so that $C(\sigma)$ is the length of the shortest program, for a fixed universal Turing machine, that outputs σ on empty input. Let $\omega = \{0, 1, 2, \ldots\}$ be the set of nonnegative integers and let $\omega^{<\omega} = \omega^*$ be the set of finite words over ω.

As Solomonoff and Kolmogorov observed, for each n there is a word $\sigma \in \{\mathtt{0}, \mathtt{1}\}^n$ with $C(\sigma) \geq n$. Indeed, each word with $C(\sigma) < n$ uses up a description of length $< n$, and there are at most $\sum_{k=0}^{n-1} 2^k = 2^n - 1 < 2^n = |\{\mathtt{0}, \mathtt{1}\}^n|$ of those.

Similarly, we have the following:

Lemma 1 (Solomonoff, Kolmogorov). *For each nonnegative integer n, there are at least $2^n - \left(2^{n-k} - 1\right)$ binary words σ of length n such that $C(\sigma) \geq n - k$.*

Proof. For each word with $C(\sigma) < n - k$, we use up at least one of the at most $2^{n-k} - 1$ many possible descriptions of length less than $n - k$, leaving at least

$$|\{\mathtt{0}, 1\}^n| - \left(2^{n-k} - 1\right)$$

words σ that must have $C(\sigma) \geq n - k$. □

1.2. Almost all words of a given length

Shallit and Wang connected their automatic complexity $A(x)$ with Kolmogorov complexity in the following theorem:

Theorem 2 (Shallit and Wang [9, proof of Theorem 8]). *For all binary words x,*

$$C(x) \leq 12A(x) + 3\log_2 |x| + O(1).$$

They mention ([9, proof of Theorem 8]), without singling it out as a lemma, the result that is our Lemma 4. Since they used, but did not give a definition of, the notion of *almost all*, we give a definition here. The notion is also known by the phrase *natural density 1*.

Definition 3. A set of strings $S \subseteq \{\mathtt{0}, 1\}^*$ contains almost all $x \in \{\mathtt{0}, 1\}^n$ if

$$\lim_{n\to\infty} \frac{|S \cap \{\mathtt{0}, 1\}^n|}{2^n} = 1.$$

Lemma 4. $C(x) \geq |x| - \log_2 |x|$ *for almost all x.*

Proof. Let $S = \{x \in \{\mathtt{0}, 1\}^* : C(x) \geq |x| - \log_2 |x|\}$. By Lemma 1,

$$\lim_{n\to\infty} \frac{|S \cap \{\mathtt{0}, 1\}^n|}{2^n} \geq \lim_{n\to\infty} \frac{2^n - \left(2^{n-\log_2 n} - 1\right)}{2^n} = \lim_{n\to\infty} 1 - \left(\frac{1}{n} - \frac{1}{2^n}\right) = 1.$$

□

Shallit and Wang then deduced the following:

Theorem 5 ([9, Theorem 8]). *For almost all $x \in \{\mathtt{0}, 1\}^n$, we have $A(x) \geq n/13$.*

Proof. By Lemma 4 and Theorem 2, there is a constant C such that for almost all x,

$$|x| - \log_2 |x| \leq C(x) \leq 12A(x) + 3\log_2 |x| + C.$$

Let $C' = C/12$. By taking n large enough, we have

$$\frac{n}{13} \leq \frac{n}{12} - \frac{1}{3}\log_2 n - C' \leq A(x).$$

□

Our main result (Theorem 18) implies that for all $\epsilon > 0$, $A(x) \geq n/(2 + \epsilon)$ for almost all words $x \in \{\mathtt{0}, 1\}^n$. Analogously, one way of expressing the Solomonoff–Kolmogorov result is as follows:

Proposition 6. *For each $\epsilon > 0$, the following statement holds: $C(x) \geq |x|(1 - \epsilon)$ for almost all $x \in \{\mathtt{0}, 1\}^n$.*

The core idea for Theorem 18 is as follows. Consider an automaton processing a word x of length n over $n+1$ points in time. We show that there exist powers $x_i^{\alpha_i}$ within x with $\alpha_i \geq 2$, and all distinct base lengths $|x_i|$, that in total occupy $\sum 1 + \alpha_i |x_i|$ time, and such that all other states are visited at most twice. Since most words do not contain any long powers, this forces the number of states to be large.

Automatic complexity, introduced by [9], is an automaton-based and length-conditional analogue of CD complexity [10]. CD complexity is in turn a computable analogue of the noncomputable Kolmogorov complexity. CD stands for 'complexity of distinguishing'. Buhrman and Fortnow [1] call it CD, whereas Sipser called it KD. $KD^t(x)$ is the minimum length of a program for a fixed universal Turing machine that accepts x, rejects all other strings and runs in at most $t(|y|)$ steps for all strings y.

The nondeterministic case of automatic complexity was studied in [3]. Among other results, that paper gave a table of the number of words of length n of nondeterministic automatic complexity A_N equal to a given number q for $n \leq 23$, and showed the following:

Theorem 7 (Hyde [9, Theorem 8], [3]). *For all x, $A_N(x) \leq \lfloor n/2 \rfloor + 1$.*

In this article we shall use $\langle a_1, \ldots, a_k \rangle$ to denote a k-tuple and denote concatenation by $\frown$. Thus, for example, $\langle 3, 6 \rangle \frown \langle 4, 4 \rangle = \langle 3, 6, 4, 4 \rangle$. When no confusion is likely, we may also denote concatenation by juxtaposition. For example, instead of $U \frown V \frown U \frown C \frown C \frown V$ we may write simply $UVUCCV$.

Definition 8. Let Σ be finite a set called the *alphabet* and let Q be a finite set whose elements are called *states*. A *nondeterministic finite automaton* (NFA) is a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$. The *transition function* $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ maps each $(q, b) \in Q \times \Sigma$ to a subset of Q. Within Q we find the *initial state* $q_0 \in Q$ and the set of *final states* $F \subseteq Q$. As usual, δ is extended to a function $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ by

$$\delta^*(q, \sigma \frown i) = \bigcup_{s \in \delta^*(q,\sigma)} \delta(s, i).$$

Overloading notation, we also write $\delta = \delta^*$. The set of words accepted by M is

$$L(M) = \{x \in \Sigma^* : \delta(q, x) \cap F \neq \emptyset\}.$$

A *deterministic finite automaton* is also a 5-tuple $M = (Q, \Sigma, \delta, q_0, F)$. In this case, $\delta : Q \times \Sigma \to Q$ is a total function and is extended to δ^* by $\delta^*(q, \sigma \frown i) = \delta(\delta^*(q, \sigma), i)$. Finally, the set of words accepted by M is

$$L(M) = \{x \in \Sigma^* : \delta(q, x) \in F\}.$$

We now formally recall our basic notions.

Definition 9 ([3, 9]). The *nondeterministic automatic complexity* $A_N(x)$ of a word $x \in \Sigma^n$ is the minimal number of states of an NFA M accepting x such that there is only one accepting walk in M of length n.

The *automatic complexity* $A(x)$ of a word $x \in \Sigma^n$ is the minimal number of states of a deterministic finite automaton M accepting x such that $L(M) \cap \Sigma^n = \{x\}$.

Insisting that there be only one accepting walk enforces a kind of unambiguity at a fixed length. This appears to reduce the computational complexity of $A_N(x)$, compared to requiring that there be only one accepted word, since one can use matrix exponentiation. It is not known whether these are equivalent definitions [5].

Clearly, $A_N(x) \leq A(x)$. Thus our lower bounds in this paper for $A_N(x)$ apply to $A(x)$ as well.

2. The power–complexity connection

The reader may note that in the context of automatic complexity, Definition 8 can without loss of generality be simplified as follows:

https://doi.org/10.1017/fms.2021.58 Published online by Cambridge University Press

1. We may assume that the set of final states is a singleton.
2. We may assume that whenever $q, r \in Q$ and $b_1, b_2 \in \Sigma$, if $r \in \delta(q, b_1) \cap \delta(q, b_2)$, then $b_1 = b_2$. Indeed, having multiple edges from q to r in an automaton witnessing the automatic complexity of a word would would violate uniqueness.
3. Each automaton M may be assumed to be *generated by a witnessing walk*. That is, only edges used by a walk taken when processing x along the unique accepting walk need to be included in M.

Let us call an NFA M *witness generated* if there is some $x \in \Sigma^*$ such that x is the only word of length $|x|$ that is accepted by M, and M accepts x along only one walk and every state and transition of M is visited during this one walk. In this case we also say that M is witness generated by x. When studying nondeterministic automatic complexity, we may without loss of generality restrict attention to witness-generated NFAs.

Definition 10. Two occurrences of words a (starting at position i) and b (starting at position j) in a word x are *disjoint* if $x = uavbw$, where u, v, w are words and $|u| = i$, $|uav| = j$.

Definition 11. A *digraph* $D = (V, E)$ consists of a set of *vertices* V and a set of *edges* $E \subseteq V^2$. Set $s, t \in V$. Set $n \geq 0$, $n \in \mathbb{Z}$. A *walk* of length n from s to t is a function $\Delta : \{0, 1, \ldots, n\} \to V$ such that $\Delta(0) = s$, $\Delta(n) = t$ and $(\Delta(k), \Delta(k+1)) \in E$ for each $0 \leq k < n$.

A *cycle* of length $n = |\Delta| \geq 1$ in D is a walk from s to s, for some $s \in V$, such that $\Delta(t_1) = \Delta(t_2), t_1 \neq t_2, \implies \{t_1, t_2\} = \{0, n\}$. Two cycles are *disjoint* if their ranges are disjoint.

Theorem 12. *Let n be a positive integer. Let $D = (V, E)$ be a digraph and set $s, t \in V$. Suppose that there is a unique walk Δ on D from s to t of length n, and that for each $e \in E$ there is a t with $(\Delta(t), \Delta(t+1)) = e$. Then there is a set of disjoint cycles $\mathcal{C}$ such that*

$$v \in V \setminus \bigcup_{C \in \mathcal{C}} \operatorname{range}(C) \implies |\{t : \Delta(t) = v\}| \leq 2,$$

and such that for each $C \in \mathcal{C}$ there exist $\mu_C \geq 2|C|$ and t_C such that

$$\begin{aligned} \{t : \Delta(t) \in \operatorname{range}(C)\} &= [t_C, t_C + \mu_C], \\ \Delta(t_C + k) &= C(k \bmod |C|) \quad \text{for all } 0 \leq k \leq \mu_C. \end{aligned} \tag{1}$$

Proof. Suppose $v \in V$ with $\{t : \Delta(t_j) = v\} = \{t_1 < t_2 < \cdots < t_k\}$ and $k \geq 3$. Let us write $\Delta_{[a,b]}$ for the sequence $(\Delta(a), \ldots, \Delta(b))$ for any a, b.

Claim. *The vertex sequence $S = \Delta_{[t_j, t_{j+1}]}$ does not depend on j.*

Proof of claim. For $k = 3$, $v \in V$ with $\Delta(t_1) = \Delta(t_2) = \Delta(t_3)$ for some $t_1 < t_2 < t_3$. Then the same vertex sequence must have appeared in $[t_1, t_2]$ and $[t_2, t_3]$,

$$\Delta_{[t_1,t_3]} = \Delta_{[t_2,t_3]}{}^\frown\Delta_{[t_1+1,t_2]},$$

or else the uniqueness of the path would be violated, since

$$\Delta_{[0,t_1-1]}{}^\frown\Delta_{[t_2,t_3]}{}^\frown\Delta_{[t_1+1,t_2]}{}^\frown\Delta_{[t_3+1,n]}$$

would be a second walk on D from s to t of length n. For $k > 3$, the only difference in the argument is notational. □

By definition of the t_js, S is a cycle except for reindexing. Thus, let $C(r) = S(t_1 + r)$ for all r, let $t_C = t_1$ and let $\mu = \mu_C$ be defined by equation (1). We have

$$t_C + \mu_C \geq t_k = t_1 + \sum_{j=1}^{k-1} t_{j+1} - t_j = t_1 + (k-1)|C|,$$

and hence $\mu_C \geq (k-1)|C| \geq 2|C|$. □

https://doi.org/10.1017/fms.2021.58 Published online by Cambridge University Press

3. Main theorem from power–complexity connection

Definition 13. Let $\mathbf{w}$ be an infinite word over the alphabet Σ, and let x be a finite word over Σ. Let $\alpha > 0$ be a rational number. The word x is said to occur in $\mathbf{w}$ with exponent α if there is a subword y of $\mathbf{w}$ with $y = x^a x_0$, where x_0 is a prefix of x, a is the integer part of α and $|y| = \alpha\, |x|$. We say that y is an α-power. The word $\mathbf{w}$ is α-power-free if it contains no subwords which are α-powers.

Here in Section 3 we show how to establish our main theorem (Theorem 18).

Definition 14. Let M be an NFA. The directed graph $D(M)$ has the set of states Q as its set of vertices and has edges (s,t) whenever $t \in \delta(s,b)$ for some $b \in \Sigma$.

Theorem 15. *Set $q \geq 1$ and $n \geq 0$, and let x be a word of length n such that $A_N(x) = q$. Then x contains a set of powers $x_i^{\alpha_i}$, $\alpha_i \geq 2$, $1 \leq i \leq m$, satisfying the following equations with $\beta_i = \lfloor \alpha_i \rfloor$:*

$$\sum_{i=1}^{m} \beta_i\, |x_i| = \sum_{i=1}^{m} \gamma_i\, |x_i|, \quad \gamma_i \in \mathbb{Z}, \gamma_i \geq 0 \implies \gamma_i = \beta_i \text{ for each } i, \tag{2}$$

$$n + 1 - m - \sum_{i=1}^{m} (\alpha_i - 2)\, |x_i| \leq 2q. \tag{3}$$

Proof. Let M be an NFA witnessing that $A_N(x) \leq q$. Let D be the digraph $D(M)$. Let $\mathcal{C}$ be a set of disjoint cycles in D as guaranteed by Theorem 12. Let $m = |\mathcal{C}|$ and write $\mathcal{C} = \{C_1, \ldots, C_m\}$. Let x_i be the word read by M while traversing C_i and let $\alpha_i = \mu_{C_i}$ from Theorem 12.

Since the C_i are disjoint, there are $\Omega := q - \sum_{i=1}^{m} |x_i|$ vertices not in $\bigcup_i C_i$. Let $P = |\{t : \Delta(t) \in C_i, \text{ for some } i\}|$ and let $N = n + 1 - P$. By Theorem 12, $N \leq 2\Omega$, and so $P = n + 1 - N \geq n + 1 - 2\Omega$. On the other hand, $P = \sum_{i=1}^{m} (1 + \alpha_i\, |x_i|)$, since a walk of length k is the range of a function with domain of cardinality $k+1$. Substituting back into the inequality $P \geq n + 1 - 2\Omega$ now yields

$$\sum_{i=1}^{m} (1 + \alpha_i\, |x_i|) \geq n + 1 - 2\left(q - \sum_{i=1}^{m} |x_i|\right)$$

and hence formula (3). □

Theorem 16. *Set $q \geq 1$ and let x be a word such that $A_N(x) \leq q$. Then x contains a set of powers $x_i^{\alpha_i}$, $\alpha_i \geq 2$, $1 \leq i \leq m$, such that all the $|x_i|$, $1 \leq i \leq m$, are distinct and nonzero, and satisfying formula (3).*

Proof. This follows from Theorem 15 once we note that unique solvability of equation (2) implies that all the lengths are distinct.

The unique solution is $\beta_k = \lfloor \alpha_k \rfloor \geq 1$. Suppose $|x_i| = \left|x_j\right|$, $i \neq j$. Then another solution is $\gamma_k = \beta_k$ for $k \notin \{i,j\}$, $\gamma_i = \beta_i - 1$, $\gamma_j = \beta_j + 1$. □

For a word $x = x_1 \cdots x_n$ with each $x_i \in \{0, 1\}$, we write $x_{[a,b]} = x_a x_{a+1} \cdots x_b$.

Definition 17. Let $x = x_1 \cdots x_n$ with each $x_i \in \{0, 1\}$. $\mathrm{Lookback}(m,k,t,x)$ is the statement that $x_{m+1+u} = x_{m+1+u-k}$ for each $0 \leq u < t$ – that is,

$$\mathrm{Lookback}(m,k,t,x) \iff x_{[m+1:m+t]} = x_{[m+1-k:m+t-k]}.$$

We can read $\mathrm{Lookback}(m,k,t,x)$ as 'position m starts a continued run with lookback amount k of length t in x'.

Theorem 18. *Let $\mathbb{P}_n$ denote the uniform probability measure on words $x \in \Gamma^n$, where Γ is a finite alphabet of cardinality at least 2. For all $\epsilon > 0$,*

$$\lim_{n \to \infty} \mathbb{P}_n \left(\left| \frac{A_N(x)}{n/2} - 1 \right| < \epsilon \right) = 1.$$

https://doi.org/10.1017/fms.2021.58 Published online by Cambridge University Press

Proof. Let us write $\log = \log_{|\Gamma|}$ in this proof. Let $d = 3$, although any fixed real number $d > 2$ will do for the proof. For $1 \le m \le n$ and $1 \le k \le m$, let $R_{m,k} = \{x \in \Gamma^n : \mathrm{Lookback}(m, k, \lceil d \log n \rceil, x)\}$. By the union bound,[1]

$$\mathbb{P}_n\left(\bigcup_{m=1}^{n}\bigcup_{k=1}^{m} R_{m,k}\right) \le \sum_{m=1}^{n}\sum_{k=1}^{m} |\Gamma|^{-d\log n} = n^{-d}\sum_{m=1}^{n} m = \frac{n(n+1)}{2}\cdot n^{-d} =: \epsilon_{n,d}. \tag{4}$$

By Theorem 16, if $A_N(x) \le q$ then x contains powers $x_i^{\alpha_i}$ with all $\alpha_i \ge 2$ and all $|x_i|$ distinct and nonzero such that formula (3) holds:

$$n + 1 - m - \sum_{i=1}^{m}(\alpha_i - 2)\,|x_i| \le 2q.$$

Applying this with $q = A_N(x)$,

$$n + 1 - m - \sum_{i=1}^{m}(\alpha_i - 2)\,|x_i| \le 2A_N(x). \tag{5}$$

Let $S_i = (\alpha_i - 1)\,|x_i|$ and $S = \sum_{i=1}^{m} S_i$. Using $|x_i| \ge 1$ and formula (5), we have

$$n + 1 - S \le n + 1 - S - m + \sum_{i=1}^{m} |x_i| \le 2A_N(x). \tag{6}$$

Using $\alpha_i \ge 2$, and the observation that *if m many distinct positive integers $|x_i|$ are all bounded by $\lceil d\log n\rceil$, then it follows that $m \le \lceil d \log n\rceil$*, we have

$$\left\{x : \max_{i=1}^{m} S_i \le \lceil d\log n\rceil\right\} \subseteq \left\{x : \max_{i=1}^{m} |x_i| \le \lceil d\log n\rceil)\right\} \subseteq \{x : m \le \lfloor d\log n\rfloor\}. \tag{7}$$

By equation (4) (since S_i is the length of a continued run in x), we have

$$\mathbb{P}_n\left(\max_{i=1}^{m} S_i \le \lceil d\log n\rceil\right) \ge 1 - \epsilon_{n,d}. \tag{8}$$

Using $S \le m \max_{i=1}^{m} S_i$ and formulas (7) and (8),

$$\begin{aligned}\mathbb{P}_n\left(S \le (\lceil d\log n\rceil)^2\right) &\ge \mathbb{P}_n\left(m \max_i S_i \le (\lceil d\log n\rceil)^2\right)\\ &\ge \mathbb{P}_n\left(\max_{i=1}^{m} S_i \le \lceil d\log n\rceil\right) \ge 1 - \epsilon_{n,d}.\end{aligned}$$

So by formula (6),

$$\mathbb{P}_n\left(A_N(x) \ge \frac{n+1}{2} - \frac{1}{2}(\lceil d\log n\rceil)^2\right) \ge 1 - \epsilon_{n,d}. \tag{9}$$

Letting $n \to \infty$ completes the proof. □

Acknowledgments. This work was partially supported a grant from the Simons Foundation (#704836).

Conflicts of Interest: None.

[1]This part is inspired by an argument in [8].

https://doi.org/10.1017/fms.2021.58 Published online by Cambridge University Press

References

[1] H. Buhrman, L. Fortnow and S. Laplante, ‘Resource-bounded Kolmogorov complexity revisited’, *SIAM J. Comput* **31**(3) (2002), 887–905.

[2] A. S. Fraenkel, ‘Complexity of protein folding’, *Bull. Math. Biol.* **55**(6) (1993), 1199–1210.

[3] K. K. Hyde and B. Kjos-Hanssen, ‘Nondeterministic automatic complexity of overlap-free and almost square-free words’, *Electron. J. Combin.* **22**(3) (2015), P3.22.

[4] B. Kjos-Hanssen, ‘On the complexity of automatic complexity’, *Theory Comput. Syst.* **61**(4) (2017), 1427–1439.

[5] B. Kjos-Hanssen, ‘Few paths, fewer words: Model selection with automatic structure functions’, *Exp. Math.* **28**(1) (2019), 121–127.

[6] A. N. Kolmogorov, ‘Three approaches to the definition of the concept “quantity of information”’, *Problemy Peredachi Informatsii* **1**(1) (1965), 3–11.

[7] A. N. Kolmogorov, ‘Three approaches to the quantitative definition of information’, *Int. J. Comput. Math.* **2** (1968), 157–168.

[8] A. Quas, ‘Longest runs and concentration of measure’, comment on MathOverflow (2016). URL: https://mathoverflow.net/q/247929.

[9] J. Shallit and M.-W. Wang, ‘Automatic complexity of strings’, *J. Autom. Lang. Comb.* **6**(4) (2001), 537–554.

[10] M. Sipser, ‘A complexity theoretic approach to randomness’, in *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, STOC’83* (Association for Computing Machinery, New York, NY, 1983), 330–335.

[11] R. J. Solomonoff, ‘A formal theory of inductive inference, I’, *Inf. Control* **7** (1964), 1–22.

[12] R. J. Solomonoff, ‘A formal theory of inductive inference, II’, *Inf. Control* **7** (1964), 224–254.