

Theory Comput Syst (2009) 45: 724–739
DOI 10.1007/s00224-009-9171-0

Depth as Randomness Deficiency

Luís Antunes · Armando Matos · André Souto ·
Paul Vitányi

Published online: 5 February 2009
© Springer Science+Business Media, LLC 2009

Abstract Depth of an object concerns a tradeoff between computation time and excess of program length over the shortest program length required to obtain the object. It gives an unconditional lower bound on the computation time from a given program in absence of auxiliary information. Variants known as logical depth and computational depth are expressed in Kolmogorov complexity theory.

We derive quantitative relation between logical depth and computational depth and unify the different *depth* notions by relating them to A. Kolmogorov and L. Levin's fruitful notion of randomness deficiency. Subsequently, we revisit the computational

The authors from University of Porto are partially supported by KCrypt (POSC/EIA/60819/2004) and funds granted to LIACC through the Programa de Financiamento Plurianual, Fundação para a Ciência e Tecnologia and Programa POSI.

L. Antunes (✉) · A. Matos · A. Souto
Faculdade de Ciências, Universidade do Porto, Rua Campo Alegre, 1021/1055, 4169007 Porto,
Portugal
e-mail: lfa@ncc.up.pt
url: <http://www.ncc.up.pt/~lfa>

A. Matos
LIACC, Rua Campo Alegre, 1021, 4169007 Porto, Portugal
e-mail: acm@ncc.up.pt
url: <http://www.ncc.up.pt/~acm>

L. Antunes · A. Souto
Instituto de Telecomunicações, Rua Campo Alegre, 1021, 4169007 Porto, Portugal

A. Souto
e-mail: andresouto@dcc.fc.up.pt
url: <http://www.ncc.up.pt/~andresouto>

P. Vitányi
Computer Science Department, University of Amsterdam, Amsterdam, The Netherlands
e-mail: Paul.Vitanyi@cw.nl
url: <http://homepages.cwi.nl/~paulv/>

depth of infinite strings, study the notion of *super deep* sequences and relate it with other approaches.

Keywords Kolmogorov complexity · Computational depth

1 Introduction

The information contained in an individual finite object (a finite binary string) can be measured by its Kolmogorov complexity—the length of the shortest binary program that computes the object. Such a shortest program contains no redundancy: every bit is information; but is it meaningful information? If we flip a fair coin to obtain a finite binary string, then with overwhelming probability that string constitutes its own shortest description. However, with overwhelming probability also, all the bits in the string are apparently meaningless information, just random noise.

The opposite of randomness is regularity; and the effective regularities in an object can be used to compress it and cause it to have lower Kolmogorov complexity. Regular objects contain laws that govern their existence and have meaning. This meaning may be instantly clear, but it is also possible that this meaning becomes intelligible only as the result of a long computation. For example, let the object in question be a book on number theory. The book will list a number of difficult theorems. However, it has very low Kolmogorov complexity since all theorems are derivable from the initial few definitions. Our estimate of the difficulty of the book is based on the fact that it takes a long time to reproduce the book from part of the information in it. We can transmit all the information in the book by just transmitting the theorems. The receiver will have to spend a long time reconstructing the proofs and the full book. On the other hand, we can send all of the book. Now the receiver has all the useful information without literally, and does not have to spend time to extract information. Hence, there is a tradeoff: in both cases we send the same information in terms of Kolmogorov complexity, but in the former case it takes a long time to reconstruct it from a short message, and in the latter case it takes a short time to reconstruct it from a long message. The existence of such book is itself evidence of some long evolution preceding it. The computational effort to transform the information into ‘usable’ information is called ‘depth’.

We also use a central notion in Kolmogorov complexity: that of ‘randomness deficiency’. The randomness deficiency of an object in a particular distribution quantifies the ‘typicality’ or ‘randomness’ of that object for that distribution. A randomness deficiency of 0 tells us that the object is typical (we believe that the object is randomly drawn from the distribution). A high randomness deficiency tells us that the object is atypical and not likely to be randomly drawn. Finally, we consider the information in one object about another one and vice versa, and since these are approximately equal we call it ‘mutual information.’

Results For finite strings, we derive quantitative relations between the different notions of depth: logical depth and computational depth (Sect. 3). In Sect. 4 we prove

that these two notions of depth are instances of a more general measure, namely, Levin's randomness deficiency, i.e., computational depth is the randomness deficiency with respect to the time bounded universal semimeasure and logical depth is the least time for which the randomness deficiency with respect to the time bounded a priori probability is upper bounded by the significance level.

Next, we study the information contained on infinite sequences. Applying the randomness deficiency with respect to $\mathbf{M} \otimes \mathbf{M}$, where \mathbf{M} is the universal lower semicomputable semimeasure over $\{0, 1\}^\infty$, Levin [7, 8] defined mutual information for infinite sequences. We observe that despite the correctness of the definition, it does not fully achieve the desired characterization of mutual information. For example, if $\alpha = \alpha^1 \alpha^2 \dots$ and $\gamma = \gamma^1 \gamma^2 \dots$ are two Kolmogorov random sequences and we construct the sequence $\beta = \alpha^1 \gamma^1 \alpha^2 \gamma^2 \dots$, then $I(\alpha : \beta) = I(\beta : \alpha) = \infty$. However intuitively β has more information about α than the other way around since from β we can fully reconstruct α but from α we can only recover half of β . In order to fulfill our intuition we propose some definitions of normalized mutual information for infinite sequences. We relate this notion with the constructive Hausdorff dimension, using the result proved by Mayordomo [13]. Namely, we show that the normalized mutual information of α with respect to β is at least the ratio of the constructive Hausdorff dimensions of α and β up to an additive factor that measures the difficulty to recover the initial segments of α from the initial segments of the same size of β . This connection motivates the definition of dimensional mutual information for infinite sequences. This measure, contrarily to the normalized mutual information, is symmetric and it is at most the minimum between the normalized mutual information of α with respect to β and vice versa.

In the last section we revisit the notion of depth for infinite sequences, proposing a new depth measure called dimensional depth. As the name suggests, this measure is related to the constructive Hausdorff dimension. We prove that dimensional depth is at most the difference between time bounded and resource unbounded versions of constructive Hausdorff dimension and finally we fully characterize super deepness using our proposed measures in a similar way as done in [5].

Previous Work Bennett [3] introduced the notion of logical depth of an object as the amount of time required for an algorithm to derive the object from a shorter description.

Antunes et al. [1] consider logical depth as one instantiation of a more general theme, computational depth, and propose several other variants based on the difference between a resource bound Kolmogorov complexity measure and the unbounded Kolmogorov complexity.

For infinite sequences, Bennett identified the classes of *weakly* and *strongly deep* sequences, and showed that the halting problem is strongly deep. Intuitively a sequence is strongly deep if no computable time bound is enough to compress infinitely many of its prefixes to within a constant number of bits of its smallest representation. An interpretation of strongly deep objects is given in [6]; a strongly deep sequence is analogous to a great work of literature for which no number of readings suffices to exhaust its value. Subsequently Judes, Lathrop, and Lutz [5] extended Bennett's work defining the classes of *weakly useful* sequences. The computational usefulness of a

sequence can be measured as the class of computational problems that can be solved efficiently, given access to that sequence. More formally, for infinite sequences, a sequence is weakly useful if every element of a non-negligible set of decidable sequences is reducible to it in recursively bounded time. Lathrop, and Lutz [5] proved that every weakly useful sequence is strongly deep in the sense of Bennett. Later, Fenner et al. [4] proved that there exist sequences that are weakly useful but not strongly useful. Lathrop and Lutz [6] introduced refinements (named *recursive weak depth* and *recursive strong depth*) of Bennett’s notion of weak and strong depth, and studied its fundamental properties, showing that recursively weakly (resp. strongly) deep sequences form a proper subclass of the class of weakly (resp. strongly) deep sequences, and also that every weakly useful sequences is recursive strongly deep.

Levin [7, 8] showed that the randomness deficiency of x with respect to μ is the largest, within an additive constant, randomness μ -test for x . So $\delta(x | \mu)$ is, in a sense, a universal characterization of “non-randomness”, “useful” or “meaningful” information in a string x with respect to a probability distribution μ .

2 Preliminaries

We briefly introduce some notions from Kolmogorov complexity, mainly the standardize notation. We refer to the textbook by Li and Vitányi [9] for more details. Let U be a fixed universal Turing machine. For technical reasons we choose one with a separate read-only input tape, that is scanned from left-to-right without backing up, a separate work tape on which the computation takes place, and a separate output tape. Upon halting, the initial segment p of the input that has been scanned is called the input “program” and the contents of the output tape is called the “output”. By construction, the set of halting programs is prefix free. We call U the reference universal prefix machine. In the rest of this paper we denote the n -length prefix of an infinite sequence α by α_n and the i th bit by α^i .

Definition 2.1 (i) The (prefix) *Kolmogorov complexity* of a finite binary string x is defined as

$$K(x) = \min_p \{|p| : U(p) = x\},$$

where p is a program, and the *Universal a priori probability* of x is

$$Q_U(x) = \sum_{U(p)=x} 2^{-|p|}.$$

(ii) A time-constructible function t from natural numbers to natural numbers is a function with the property that $t(n)$ can be constructed from n by a Turing machine in time of order $O(t(n))$. For every time-constructible t , the *t -time-bounded Kolmogorov complexity* of x is defined as

$$K^t(x) = \min_p \{|p| : U(p) = x \text{ in at most } t(|x|) \text{ steps}\},$$

and the *t-time bounded Universal a priori probability* is defined as

$$Q_U^t(x) = \sum_{U^t(p)=x} 2^{-|p|},$$

and $U^t(p) = x$ means that U computes x in at most $t(|x|)$ steps and halts.

A different universal Turing machine may affect the program size $|p|$ by at most a constant additive term, and the running time t by at most a logarithmic multiplicative factor. The same will hold for all other measures we will introduce.

Levin [7] showed that the Kolmogorov complexity of a string x coincides up to an additive constant term with the logarithm of $1/Q_U(x)$. This result is called the “Coding Theorem” since it shows that the shortest upper semicomputable code is a Shannon-Fano code of the greatest lower semicomputable probability mass function. In order to state formally the Coding theorem we need the following theorem on the existence of a universal lower semicomputable discrete semimeasure (Theorem 4.3.1 in [9]).

Theorem 2.2 *There exists a universal lower semicomputable discrete semimeasure over $\{0, 1\}^*$, denoted by \mathbf{m} .*

Theorem 2.3 (Coding theorem) *For every $x \in \{0, 1\}^n$,*

$$K(x) = -\log Q_U(x) = -\log \mathbf{m}(x)$$

with equality up to an additive constant c .

Hence, if x has high probability because it has many long descriptions then it must have a short description too.

We refer to *mutual information* of two finite strings as

$$I(x : y) = K(x) + K(y) - K(x, y).$$

Notice that the mutual information is symmetric, i.e., $I(x : y) = I(y : x)$.

3 Depth

Bennett [3] defines the *b-significant logical depth* of an object x as the time required by the reference universal Turing machine to generate x by a program that is no more than b bits longer than the shortest descriptions of x . Bennett talks about time as the number of steps; without loss of generality we consider the number of steps $t(|x|)$, where t is a time-constructible function.

Definition 3.1 (Logical depth) *The logical depth of a string x at a significance level b is*

$$\text{ldepth}_b(x) = \min \left\{ t(|x|) : \frac{Q_U^t(x)}{Q_U(x)} \geq 2^{-b} \right\},$$

where the minimum is taken over all time constructible t .

Given a significance level b , the logical depth of a string x is the minimal running time $t(|x|)$, such that programs running in at most $t(|x|)$ steps account for approximately a $1/2^b$ fraction of x 's universal probability. This is Bennett's Tentative Definition 0.3 in [3, p. 240].

In fact, with some probability we can derive the string by simply flipping a coin. But for long strings this probability is exceedingly small. If the string has a short description then we can flip that description with higher probability. Bennett's proposal tries to express the tradeoff between the probability of flipping a short program and the shortest computation time from program to object.

Antunes et al. [1] developed the notion of computational depth in order to capture the tradeoff between the amount of help bits required and the reduced computation time to compute a string. The concept is simple: they consider the difference of two versions of Kolmogorov complexity measures.

Definition 3.2 (Basic computational depth) Let t be a time constructible function. For any finite binary string x we define

$$\text{depth}^t(x) = K^t(x) - K(x).$$

In Definition 1 of [3, p. 241] we find

Definition 3.3 A string x is $(t(|x|), b)$ -deep iff $t(|x|)$ is the least number of steps to compute x from a program of length at most $K(x) + b$.

Then, it is straightforward that $\text{depth}^t(x) = K^t(x) - K(x)$ iff x is $(t(|x|), K^t(x) - K(x))$ -deep. Bennett remarks, [3, p. 241], "The difference between [Definitions 3.3 and 3.1] is rather subtle philosophically and not very great quantitatively." This is followed by [3, Lemma 5, p. 241] which is an informal version of [9, Theorem 7.7.1]. The proof of Item (ii) below uses an idea in the proof of the latter theorem.

Definition 3.4 Let t be a recursive function. Define $K(t)$ as the (prefix) Kolmogorov complexity of t by $K(t) = \min_i \{i : T_i \text{ computes } t(\cdot)\}$, where T_1, T_2, \dots is the standard enumeration of all Turing machines.

Theorem 3.5 Let t be a time-constructible function (hence it is recursive and $K(t)$ is defined in Definition 3.4).

- (i) If b is the minimum value such that $\text{ldepth}_b(x) = t(|x|)$, then $\text{depth}^t(x) \geq b + O(1)$.
- (ii) If $\text{depth}^t(x) = b$, then $\text{ldepth}_{b+\min\{K(b), K(t)\}+O(1)}(x) \geq t(|x|)$.

Proof (i) Assume, $\text{ldepth}_b(x) = t(|x|)$. So

$$\frac{Q_U^t(x)}{Q_U(x)} \geq 2^{-b},$$

with $t(|x|)$ least. Assume furthermore that b is the least integer so that the inequality holds for this $t(|x|)$. We also have

$$\frac{Q_U^t(x)}{Q_U(x)} \geq \frac{2^{-K^t(x)}}{Q_U(x)} = 2^{-(K^t(x)-K(x)-O(1))} = 2^{-b-\Delta},$$

where $b + \Delta = K^t(x) - K(x) - O(1)$. The first inequality holds since the sum $Q_U^t(x)$ comprises a term $2^{-K^t(x)}$ based on a shortest program of length $K^t(x)$ computing x in at most $t(|x|)$ steps. Since b is the least integer, it follows that $\Delta \geq 0$. Since $\text{depth}^t(x) = K^t(x) - K(x)$, we find that $\text{depth}^t(x) \geq b + O(1)$.

(ii) Assume that $\text{depth}^t(x) = b$, that is, x is $(t(|x|), b)$ -deep. We can enumerate the set S of all programs computing x in time at most $t(|x|)$ by simulating all programs of length $l \leq |x| + 2 \log |x|$ for $t(|x|)$ steps. Hence, the shortest such program q enumerating S has length $|q| \leq K(x, t) + O(1)$. But we achieve the same effect if, given x and b we enumerate all programs of length l as above in order of increasing running time and stop when the accumulated algorithmic probability exceeds $2^{-K(x)+b}$. The running time of the last program is $t(|x|)$. (This shows that $K(t, x) \leq K(b, x) + O(1)$, not $K(t) \leq K(b) + O(1)$). The shortest program r doing this has length $|r| \leq K(x, b) + O(1)$. Hence, $K(S) \leq \min\{K(x, t), K(x, b)\} + O(1)$. By definition, $Q_U^t(x) = \sum_{p \in S} 2^{-|p|}$. Assume, by way of contradiction, that

$$\frac{Q_U^t(x)}{Q_U(x)} < 2^{-b-\min\{K(b), K(t)\}-O(1)}$$

Since $Q_U(x) = 2^{-K(x)-O(1)}$, we have

$$Q_U^t(x) < 2^{-K(x)-b-\min\{K(b), K(t)\}-O(1)}$$

Denote $m = K(x) + b + \min\{K(b), K(t)\} + O(1)$. Therefore, $\sum_{p \in S} 2^{-|p|} < 2^{-m}$. Now every string in S can be effectively compressed by at least $m - K(S) - O(1)$ bits. Namely,

$$\sum_{p \in S} 2^{-|p|+m} < 1$$

The latter inequality is a Kraft inequality, and hence the elements of S can be coded by a prefix code with the code word length for p at most $|p| - m$. In order to make this coding effective, we use a program of length $K(S)$ to enumerate exactly the strings of S . This takes an additional $K(S) + O(1)$ bits in the code for each $p \in S$. This way, each $p \in S$ is effectively compressed by $m - K(S) - O(1)$ bits. Therefore, each $p \in S$ can be compressed by at least $K(x) + b + \min\{K(b), K(t)\} - \min\{K(x, t), K(x, b)\}$ bits, up to an additive constant we can set freely, and hence by more than b bits which is a contradiction. Hence,

$$\frac{Q_U^t(x)}{Q_U(x)} \geq 2^{-b-\min\{K(t), K(b)\}-O(1)}$$

which proves (ii). □

4 A Unifying Approach

Logical depth and computational depth are all instances of a more general measure, namely the randomness deficiency of a string x with respect to a probability distribution, Levin [7, 8]. In the rest of this paper, with some abuse of notation (see [9]), a function $\mu : \{0, 1\}^* \rightarrow \mathbb{R}$ defines a *probability measure*, or *measure* for short, if

$$\begin{aligned}\mu(\epsilon) &= 1, \\ \mu(x) &= \sum_{a \in \{0,1\}} \mu(xa).\end{aligned}$$

Definition 4.1 Let μ be a computable measure. The value

$$\delta(x \mid \mu) = \left\lceil \log \frac{Q_U(x)}{\mu(x)} \right\rceil$$

is the *randomness deficiency*¹ of x with respect to μ . Here Q_U is the universal a priori probability of Definition 2.1.

Note that $Q_U(x)$ is of exact order of magnitude of $2^{-K(x)}$ by the Coding Theorem 2.3, i.e., up to multiplicative terms $Q_U(x)$ and $2^{-K(x)}$ are equal. (In the literature, see for example [9], $\mathbf{m}(x) = 2^{-K(x)}$ is used instead of $Q_U(x)$, and it is straightforward that this is equivalent up to a multiplicative independent constant by the Coding Theorem.)

We now observe that logical depth and computational depth of a string x equals the randomness deficiency of x with respect to the measures $Q^t(x) = \sum_{U^t(p)=x} 2^{-|p|}$ and $2^{-K^t(x)}$ respectively. The proofs follow directly from the definitions.

Lemma 4.2 Let x be a finite binary string and let t be a time-constructible function.

- (i) $\text{ldepth}_b(x) = \min\{t : \delta(x \mid Q^t) \leq b\}$.
- (ii) $\text{depth}^t(x) = \delta(x \mid \mathbf{m}^t)$ where $\mathbf{m}^t(z) = 2^{-K^t(z)}$.

5 On the Information of Infinite Strings

Based on the unification of depth concepts for finite strings, in this section we extend those ideas for infinite sequences. In order to motivate our approach we start by introducing Levin's notion of randomness deficiency for infinite sequences. Let \mathbf{M} be the universal lower semicomputable (continuous) semimeasure over $\{0, 1\}^\infty$ as defined, and proved to exist, by [8] (see also [9]). If $\alpha \in \{0, 1\}^\infty$, then with $\alpha = \alpha^1 \alpha^2 \dots$ with $\alpha^i \in \{0, 1\}$, we write $\alpha_n = \alpha^1 \alpha^2 \dots \alpha^n$. Finally, we write ' $\mathbf{M}(x)$ ' and ' $\mu(x)$ ' as notational shorthand for ' $\mathbf{M}(\Gamma_x)$ ' and ' $\mu(\Gamma_x)$ ', with $x \in \{0, 1\}^*$ and Γ_x is the cylinder $\{\omega : \omega \in \{x\}\{0, 1\}^\infty\}$. Strictly speaking, $\mathbf{M}(x)$ is not over $\{0, 1\}^\infty$ but over $\{0, 1\}^\infty \cup \{0, 1\}^*$, see also [9], and $\mathbf{M}(x)$ is the probability concentrated on the set of finite and infinite sequences starting with x .

¹ $\lfloor r \rfloor$ denotes the integer part of r and $\lceil \alpha \rceil$ denotes the smallest integer bigger than α .

Definition 5.1 (Levin) The value $D(\alpha \mid \mu) = \lfloor \log(\sup_n \frac{\mathbf{M}(\alpha_n)}{\mu(\alpha_n)}) \rfloor$ is called the randomness deficiency of α with respect to the semimeasure μ . Here $\mathbf{M}(\alpha_n)$ is the probability density function of $\mathbf{M}(\alpha_n)$.

Let α and β be two sequences and $\mathbf{M} \otimes \mathbf{M}$ be defined by $\mathbf{M} \otimes \mathbf{M}(\alpha, \beta) = \mathbf{M}(\alpha)\mathbf{M}(\beta)$.

Definition 5.2 (Levin) The value $I(\alpha : \beta) = D((\alpha, \beta) \mid \mathbf{M} \otimes \mathbf{M})$ is called the amount of information in α about β or the deficiency of their independence.

This definition is equivalent to the mutual information $I(\alpha : \beta) = \sup_n I(\alpha_n : \beta_n)$.

Example 5.3 Let α and γ be two random infinite and independent strings (in the sense that their prefixes are independent). Consider the following sequence

$$\beta = \alpha^1 \gamma^1 \alpha^2 \gamma^2 \dots$$

By Definition 5.2 we have

$$\begin{aligned} I(\alpha : \beta) &= \sup_n I(\alpha_n : \beta_n) \\ &= \sup_n (K(\alpha_n) + K(\beta_n) - K(\alpha_n, \beta_n)) \\ &\geq \sup_n \left(n + n - \left(n + \frac{n}{2} \right) \right) = \infty. \end{aligned}$$

As $I(\beta : \alpha) = I(\alpha : \beta)$ then $I(\beta : \alpha) = \infty$.

However, intuitively β contains more information about α than the other way around, since from the sequence β we can totally reconstruct α but from α we can only recover half of β , namely, the bits with odd indexes.

This seems to be a lacuna in Definition 5.2. The definition says more when the information is finite but that is precisely when we do not need an accurate result. Notice that if the sequences are finite we can argue that they are independent. In the infinite case, one should be able to classify the cases where the mutual information is infinite. Two infinite sequences may have infinite mutual information and yet infinite information may be still lacking to reconstruct one of them out of the other one. In the previous example α fails to provide all the information of β related to γ , which has infinite information. In this section we will present two approaches to reformulate the definition of “mutual information” in order to fulfill our intuition. In order to have a proportion of information as the prefixes grow we need to do some normalization in the process.

5.1 The Mutual Information Point of View

We are looking for a normalized mutual information measure I_m that applied to Example 5.3 gives

$$I_m(\alpha : \alpha) = 1$$

$$I_m(\alpha : \beta) = 1/2$$

$$I_m(\beta : \alpha) = 1$$

$$I_m(\beta : \beta) = 1$$

Contrarily to Levin's definition of mutual information for infinite sequences, and accordingly to our intuition, the above conditions imply that the normalized version must be non-symmetric.

Definition 5.4 (First attempt) Given two infinite sequences α and β the normalized mutual information that β has about α is defined as

$$I_m(\beta : \alpha) = \lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{I(\beta_m : \alpha_n)}{I(\alpha_n : \alpha_n)}$$

The major drawback of this definition is the fact that the limit does not always exist.² However, it does exist for the Example 5.3 with the desired properties. Furthermore, we obtain for the same α and β

$$I_m(\alpha : \alpha) = 1;$$

$$I_m(\beta : \beta) = 1;$$

$$I_m(\alpha : \beta) = \lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{m + n - (m + n - n/2)}{n} = \frac{1}{2};$$

$$I_m(\beta : \alpha) = \lim_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{m + n - m}{n} = 1.$$

Definition 5.5 (Normalized mutual information for infinite sequences) Given two infinite sequences α and β we define the lower normalized mutual information that β has about α as

$$I_{m*}(\beta : \alpha) = \liminf_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{I(\beta_m : \alpha_n)}{I(\alpha_n : \alpha_n)}$$

and the upper normalized mutual information that β has about α as

$$I_m^*(\beta : \alpha) = \limsup_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{I(\beta_m : \alpha_n)}{I(\alpha_n : \alpha_n)}$$

Notice that these definitions also fulfill the requirements presented in the beginning of this section with respect to Example 5.3.

We now can define independence with respect to normalized mutual information:

Definition 5.6 Two sequences, α and β , are independent if $I_m^*(\alpha : \beta) = I_m^*(\beta : \alpha) = 0$.

In [11, 12], the author developed a constructive version of Hausdorff dimension. That dimension assigns to every binary sequence α a real number $\dim(\alpha)$ in the interval $[0, 1]$. Lutz claims that the dimension of a sequence is a measure of its in-

²Notice that there are sequences α for which $\lim_n \frac{n}{K(\alpha_n)}$ does not exist.

formation density. The idea is to differentiate sequences by non-randomness degrees, namely by their dimension. Our approach is precisely to introduce a measure of density of information that one sequence has about the other, in the total amount of the other's information. So we differentiate non-independent sequences, by their normalized mutual information.

Mayordomo [13] redefined constructive Hausdorff dimension in terms of Kolmogorov complexity.

Theorem 5.7 (Mayordomo) *For every sequence α ,*

$$\dim(\alpha) = \liminf_{n \rightarrow \infty} \frac{K(\alpha_n)}{n}$$

So, now the connection between constructive dimension and normalized information measure introduced here is clear. It is only natural to accomplish results about the Hausdorff constructive dimension of a sequence, knowing the dimension of another, and their normalized information.

Lemma 5.8 *Let α and β be two infinite sequences. Then*

$$I_m^*(\alpha : \beta) \cdot \dim(\beta) \geq \dim(\alpha) + \liminf_{n \rightarrow \infty} \frac{K(\alpha_n | \beta_n)}{n}$$

Proof

$$\begin{aligned} I_m^*(\alpha : \beta) \cdot \dim(\beta) &= \limsup_n \lim_m \frac{I(\alpha_m : \beta_n)}{I(\beta_n : \beta_n)} \cdot \liminf_n \frac{K(\beta_n)}{n} \\ &\geq \liminf_n \liminf_m \frac{I(\alpha_m : \beta_n)}{n} \\ &\geq \liminf_n \liminf_m \frac{I(\alpha_m : \beta_n)}{m} \\ &\geq \liminf_n \liminf_m \frac{K(\alpha_m) - K(\alpha_m | \beta_m)}{m} \\ &\geq \liminf_m \frac{K(\alpha_m)}{m} + \liminf_m \frac{-K(\alpha_m | \beta_m)}{m} \\ &= \dim(\alpha) + \liminf_m \frac{K(\alpha_m | \beta_m)}{m} \quad \square \end{aligned}$$

Note that, in the previous lemma the (unexpected) additive term $\liminf_m \frac{K(\alpha_m | \beta_m)}{m}$ is necessary to express the hardness of recover α given β .

We present now the time bounded version of $\dim(\alpha)$. This definition will be important later on this paper.

Definition 5.9 Let t be a time-constructible function. The t -bounded dimension of an infinite sequence α is defined as

$$\dim^t(\alpha) = \liminf_{n \rightarrow \infty} \frac{K^t(\alpha_n)}{n}$$

5.2 The Hausdorff Constructive Dimension Point of View

In this subsection we define a version of mutual information between two sequences based on Hausdorff constructive dimension and establish a connection to it.

Definition 5.10 The dimensional mutual information of the sequences α and β is defined as

$$I_{dim}(\alpha : \beta) = \dim(\alpha) + \dim(\beta) - 2 \dim\langle \alpha, \beta \rangle$$

This measure of mutual information is symmetric. The definition considers twice $\dim\langle \alpha, \beta \rangle$ because when encoding the prefixes α_n and β_n the result is a $2n$ -length string. Notice that,

$$\begin{aligned} I_{dim}(\alpha : \beta) &= \dim(\alpha) + \dim(\beta) - 2 \dim\langle \alpha, \beta \rangle \\ &= \liminf_{n \rightarrow \infty} \frac{K(\alpha_{n/2})}{n/2} + \liminf_{n \rightarrow \infty} \frac{K(\beta_{n/2})}{n/2} - 2 \liminf_{n \rightarrow \infty} \frac{K(\langle \alpha, \beta \rangle_n)}{n} \\ &\leq \liminf_{n \rightarrow \infty} \frac{K(\alpha_{n/2}) + K(\beta_{n/2}) - K(\alpha_{n/2}, \beta_{n/2})}{n/2} \\ &= \liminf_{n \rightarrow \infty} \frac{I(\alpha_n : \beta_n)}{n} \\ &\leq \liminf_{n \rightarrow \infty} \frac{I(\alpha_n : \beta_n)}{K(\beta_n)} \\ &\leq \liminf_{n \rightarrow \infty} \lim_{m \rightarrow \infty} \frac{I(\alpha_m : \beta_n)}{K(\beta_n)} \\ &= I_{m*}(\alpha : \beta) \end{aligned}$$

The third inequality is true due to the following fact:

$$I(\beta_n : \alpha_m) = K(\beta_n) - K(\beta_n | \alpha_m) \geq K(\beta_n) - K(\beta_n | \alpha_n) = I(\beta_n : \alpha_n).$$

By the symmetry of the definition we also have that $I_{dim}(\alpha : \beta) \leq I_{m*}(\beta : \alpha)$. These two facts prove the following lemma:

Lemma 5.11 *Let α and β be two sequences. Then*

$$I_{dim}(\alpha : \beta) \leq \min(I_{m*}(\alpha : \beta), I_{m*}(\beta : \alpha))$$

One can easily modify the definitions introduced in this section by considering the limits when n goes to the length of the string, or the maximum length of the strings being considered. One should also notice that when x and y are finite strings and $K(y) \geq K(x)$, $I_{m*}(x : y)$ is $1 - d(x, y)$, where $d(x, y)$ is the normalized information distance studied in [10].

6 Depth of Infinite Strings

In this section we revisit depth for infinite sequences. We introduce a new depth measure, prove that it is closely related with constructive Hausdorff dimension and use it to characterize super deepness. To motivate our definitions we recall the definitions of the classes of weakly (vs. strongly deep) sequences and weakly useful (vs. strongly useful) sequences.

Definition 6.1 [3] An infinite binary sequence α is defined as

- weakly deep if it is not computable in recursively bounded time from any algorithmically random infinite sequence.
- strongly deep if at every significance level b , and for every recursive function t , all but finitely many initial segments α_n have logical depth exceeding $t(n)$.

Definition 6.2 [4] An infinite binary sequence α is defined as

- weakly useful if there is a computable time bound within which all the sequences in a non-measure 0 subset of the set of decidable sequences are Turing reducible to α .
- strongly useful if there is a computable time bound within which every decidable sequence is Turing reducible to α .

The relation between logical depth and usefulness was studied by Juedes, Lathrop and Lutz [5] who defined the conditions for weak and strong usefulness and showed that every weakly useful sequence is strongly deep. This result generalizes Bennett's remark that the diagonal halting problem is strongly deep, strengthening the relation between depth and usefulness. Latter Fenner et al. [4] proved the existence of sequences that are weakly useful but not strongly useful.

The Hausdorff constructive dimension has a close connection with the information theories for infinite strings studied before, see for example [4, 11, 12] and [13]. Therefore, in this section we define the dimensional computational depth of a sequence in order to study the nonrandom information on a infinite sequence.

Definition 6.3 The dimensional depth of a sequence α is defined as

$$\text{depth}_{\text{dim}}^t(\alpha) = \liminf_{n \rightarrow \infty} \frac{\delta(\alpha_n \mid 2^{-K^t(\alpha_n)})}{n}.$$

Lemma 6.4

$$\text{depth}_{\text{dim}}^t(\alpha) \leq \text{dim}^t(\alpha) - \text{dim}(\alpha)$$

Proof

$$\begin{aligned} \text{depth}_{\text{dim}}^t(\alpha) &= \liminf_{n \rightarrow \infty} \frac{\delta(\alpha_n \mid 2^{-K^t(\alpha_n)})}{n} \\ &= \liminf_{n \rightarrow \infty} \frac{K^t(\alpha_n) - K(\alpha_n)}{n} \\ &\leq \text{dim}^t(\alpha) - \text{dim}(\alpha). \end{aligned}$$

The last inequality holds since the sequence of values $K(\alpha_n)/n$ is non negative and then $\liminf_n -K(\alpha_n)/n \leq -\liminf_n K(\alpha_n)/n$. \square

Now, in the definition of strongly deep sequences, instead of considering a fixed significance level s we consider a significance level function $s : \mathbb{N} \rightarrow \mathbb{N}$. Naturally, we want $s(n)$ to grow very slowly so we assume for example that $s = o(n)$. With this replacement we obtain a tighter definition as deepness decreases with the increase of the significance level.

Definition 6.5 A sequence is called super deep if for every significance level function $s : \mathbb{N} \rightarrow \mathbb{N}$, such that $s = o(n)$, and for every recursive function $t : \mathbb{N} \rightarrow \mathbb{N}$, all but finitely many initial segments α_n have logical depth exceeding $t(n)$.

We have already characterized super deep sequences using their dimensional depth in Theorem 3.5. In fact we have

$$\text{ldepth}_b(x) = t(|x|), \quad \text{with } b \text{ minimal} \Rightarrow \text{depth}^t(x) \geq b + O(1)$$

Theorem 6.6 A sequence α is super deep if and only if $\text{depth}_{\text{dim}}^t(\alpha) > 0$ for all recursive time bound t .

Proof Let α be a super deep sequence. Then for every significance level function s , such that $s = o(n)$ and every recursive function t we have that for almost all n , $\text{ldepth}_{s(n)}(\alpha_n) > t(n)$. Then

$$\text{depth}^{t(n)}(\alpha_n) > s(n).$$

Now if for some time bound g , $\text{depth}_{\text{dim}}^g(\alpha) = 0$ then there exists a bound S , such that $S = o(n)$, and, infinitely often

$$\text{depth}^{g(n)}(\alpha_n) < S(n).$$

This is absurd and therefore for all recursive time bound t , $\text{depth}_{\text{dim}}^t(\alpha) > 0$.

Conversely if $\text{depth}_{\text{dim}}^t(\alpha) > 0$ then there is some $\epsilon > 0$ such that for almost all n , $\text{depth}_{\text{dim}}^{t(n)}(\alpha_n) > \epsilon n$. This implies that

$$\text{ldepth}_{s(n)}(\alpha_n) > \text{ldepth}_{\epsilon n}(\alpha_n) > t(n)$$

for all significance function $s = o(n)$ and almost all n . So α is super deep. \square

In the next theorem we express other equivalent ways to define super deepness.

Theorem 6.7 For every sequence α the following conditions are equivalent.

1. α is super deep;
2. For every recursive time bound $t : \mathbb{N} \rightarrow \mathbb{N}$ and every significance function $g = o(n)$, $\text{depth}^t(\alpha_n) > g(n)$ for all except finitely many n ;

3. For every recursive time bound $t : \mathbb{N} \rightarrow \mathbb{N}$ and every significance function $g = o(n)$, $Q(\alpha_n) \geq 2^{g(n)} Q^t(\alpha_n)$ for all except finitely many n .

Proof [Sketch] The equivalence $(1 \Leftrightarrow 2)$ was proved in Theorem 6.6. To show that $(2 \Leftrightarrow 3)$ consider the following sets:

$$D_g^t = \{\alpha \in \{0, 1\}^\infty : \text{depth}^t(\alpha_n) \geq g(n) \text{ a.e.}\}$$

$$\tilde{D}_g^t = \{\alpha \in \{0, 1\}^\infty : Q(\alpha_n) \geq 2^{g(n)} Q^t(\alpha_n) \text{ a.e.}\}$$

The proof now is an immediate consequence of the following lemma:

Lemma 6.8 (Lemma 3.5 in [5]) *If t is a recursive time bound then there exists constants c_1 and c_2 and a recursive time bound t_1 such that $D_{g+c_1}^{t_1} \subset \tilde{D}_g^t$ and $\tilde{D}_{g+c_2}^t \subset D_g^t$.* □

Following the ideas in [5] to prove that every weakly useful sequence is strongly deep we can prove that every weakly useful sequence is super deep.

Theorem 6.9 *Every weakly useful sequence is super deep.*

For the proof of this result we need the following lemmas:

Lemma 6.10 (Lemma 5.5 in [5]) *Let $s : \mathbb{N} \rightarrow \mathbb{N}$ be strictly increasing and time-constructible with the constant c_s as witness. For each s -time-bounded Turing machine M , there is a constant c_M that satisfies the following. Given non-decreasing functions $t, g : \mathbb{N} \rightarrow \mathbb{N}$ we define $s^*, \tau, \hat{t}, \hat{g} : \mathbb{N} \rightarrow \mathbb{N}$ by*

$$s^*(n) = 2^{s(\lceil \log n \rceil) + 1},$$

$$\tau(n) = t(s^*(n+1) + 4s^*(n+1) + 2(n+1)c_s s(|w|) + 2ns^*(n+1)s(|w|)),$$

$$\hat{t} = c_M(1 + \tau(n) \lceil \log \tau(n) \rceil),$$

$$\hat{g} = g(s^*(n+1)) + c_M,$$

where w is the binary representation of n . For all sequences α, β , if β is Turing reducible to α in time s by M and $\beta \in D_{\hat{g}}^{\hat{t}}$ then $\alpha \in D_g^t$.

Lemma 6.11 (Corollary 5.9 in [5]) *For every recursive function $t : \mathbb{N} \rightarrow \mathbb{N}$ and every $0 < \gamma < 1$, the set $D_{\gamma n}^t$ has measure 1 in the set of recursive sequences.*

Proof of Theorem 6.9 Let α be a weakly useful sequence. To prove that α is super deep we show that for every recursive time bound t and every any significance level $g = o(n)$, $\alpha \in D_g^t$, where D_g^t is the set defined in proof of Theorem 6.7.

Since α is weakly useful then there exists a recursive time bound s (that without loss of generality we can assume increasing) such that the set $DTIME^\alpha(s)$ of all

sequences that are Turing reducible to α has positive measure in the set of recursive sequences. Using Lemma 6.10, to conclude that $\alpha \in D_g^t$ all that is necessary is to prove that there exists $\beta \in D_{\hat{g}}^{\hat{t}} \cap DTIME^\alpha(s)$, where \hat{t} and \hat{g} are described in same lemma.

Fix $\gamma \in]0, 1[$ and consider $\tilde{t}(n) = n(1 + \gamma(n)\lceil \log n \rceil)$ where γ is obtained from t and s as in Lemma 6.10. Since \tilde{t} is recursive, by Lemma 6.11, $D_{\gamma n}^{\tilde{t}}$ has measure 1 in the set of all recursive sequences. Thus $D_{\gamma n}^{\tilde{t}} \cap DTIME^\alpha(s)$ has measure 1 and in particular is non empty. As $\tilde{t}(n) > \hat{t}(n)$ a.e. and $\gamma n > o(n) = g$ a.e. it follows, directly from the definitions, that $D_{\gamma n}^{\tilde{t}} \subset D_{\hat{g}}^{\hat{t}}$ and then $D_{\gamma n}^{\tilde{t}} \neq \emptyset$, as we wanted to show. \square

Corollary 6.12 *The characteristic sequences of the halting problem and the diagonal halting problem are super deep.*

Proof In [3], the author proved that the characteristic sequences of the halting problem and the diagonal halting problem are weakly useful. Then, it follows from Theorem 6.9 that these two sequences are super deep. \square

Acknowledgements We thank Harry Buhrman, Lance Fortnow and Ming Li for comments and suggestions. This work follows the work started in [2], we thank Armindo Costa for some useful comments, suggestions and ideas.

References

1. Antunes, L., Fortnow, L., van Melkebeek, D., Vinodchandran, N.: Computational depth: concept and applications. *Theor. Comput. Sci.* **354**(3), 391–404 (2006)
2. Antunes, L., Costa, A., Matos, A., Vítányi, P.: Computational depth of infinite strings revisited. In: *Proc. CiE 2007, Computation and Logic in the Real World* (2007)
3. Bennett, C.: Logical depth and physical complexity. In: *The Universal Turing Machine: A Half-Century Survey*, pp. 227–257. Oxford University Press, London (1988)
4. Fenner, S., Lutz, J., Mayordomo, E., Reardon, P.: Weakly useful sequences. *Inf. Comput.* **197**, 41–54 (2005)
5. Juedes, D., Lathrop, J., Lutz, J.: Computational depth and reducibility. *Theor. Comput. Sci.* **132**, 37–70 (1994)
6. Lathrop, J., Lutz, J.: Recursive computational depth. *Inf. Comput.* **153**, 139–172 (1999)
7. Levin, L.: Laws of information conservation (nongrowth) and aspects of the foundation of probability theory. *Probl. Inf. Transm.* **10**, 206–210 (1974)
8. Levin, L.: Randomness conservation inequalities: information and independence in mathematical theories. *Inf. Control* **61**, 15–37 (1984)
9. Li, M., Vítányi, P.: *An Introduction to Kolmogorov Complexity and its Applications*, 2nd edn. Springer, Berlin (1997)
10. Li, M., Chen, X., Li, X., Ma, B., Vítányi, P.: The similarity metric. *IEEE Trans. Inf. Theory* **50**(12), 3250–3264 (2004)
11. Lutz, J.: Dimension in complexity classes. In: *Proceedings of the 15th IEEE Conference of Computational Complexity*. IEEE Comput. Soc., Los Alamitos (2000)
12. Lutz, J.: The dimensions of individual strings and sequences. Technical Report [cs.CC/0203017](https://arxiv.org/abs/cs.CC/0203017), ACM Computing Research Repository (2002)
13. Mayordomo, E.: A Kolmogorov complexity characterization of constructive Hausdorff dimension. *Inf. Process. Lett.* **84**, 1–3 (2002)