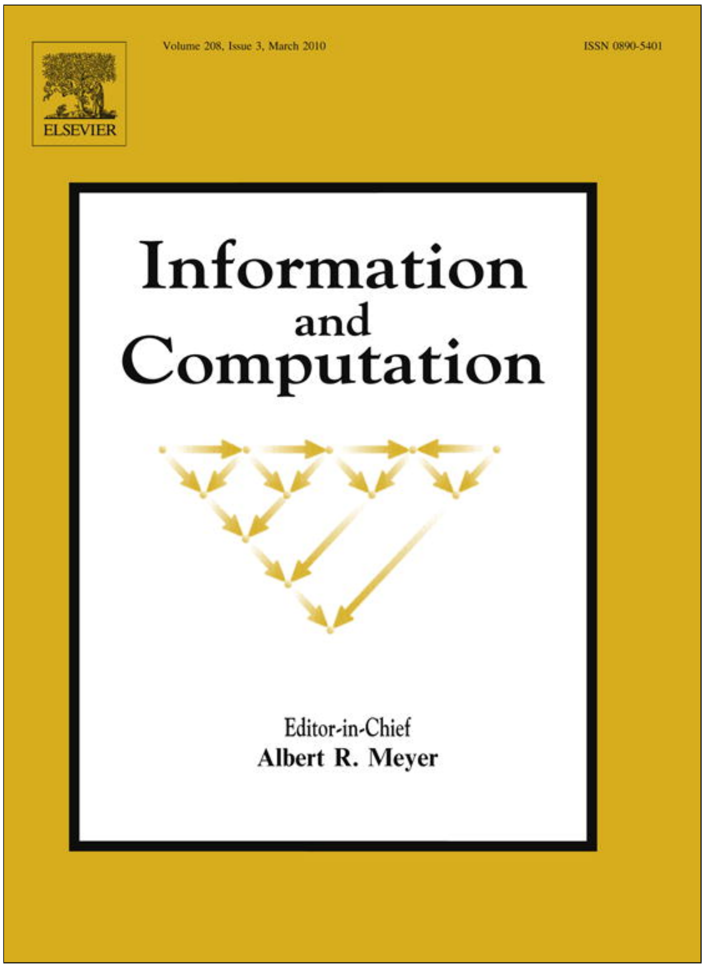
Volume 208, Issue 3, March 2010
ISSN 0890-5401
ELSEVIER
Information
and
Computation
Editor-in-Chief
Albert R. Meyer

Information and Computation 208 (2010) 292–308

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Algorithmically independent sequences ☆

Cristian S. Calude [a,1], Marius Zimand [b,*,2]

[a] Department of Computer Science, University of Auckland, New Zealand
[b] Department of Computer and Information Sciences, Towson University, Baltimore, MD, USA

ARTICLE INFO

Article history:
Received 31 October 2008
Revised 19 May 2009
Available online 18 June 2009

Keywords:
Algorithmic information theory
Independence
Mutual information
Randomness

ABSTRACT

Two objects are independent if they do not affect each other. Independence is well-understood in classical information theory, but less in algorithmic information theory. Working in the framework of algorithmic information theory, the paper proposes two types of independence for arbitrary infinite binary sequences and studies their properties. Our two proposed notions of independence have some of the intuitive properties that one naturally expects. For example, for every sequence x, the set of sequences that are independent with x has measure one. For both notions of independence we investigate to what extent pairs of independent sequences, can be effectively constructed via Turing reductions (from one or more input sequences). In this respect, we prove several impossibility results. For example, it is shown that there is no effective way of producing from an arbitrary sequence with positive constructive Hausdorff dimension two sequences that are independent (even in the weaker type of independence) and have super-logarithmic complexity. Finally, a few conjectures and open questions are discussed.

1. Introduction

Intuitively, two objects are independent if they do not affect each other. The concept is well-understood in classical information theory. There, the objects are random variables, the information in a random variable is its Shannon entropy, and two random variables X and Y are declared to be independent if the information in the join (X, Y) is equal to the sum of the information in X and the information in Y. This is equivalent to saying that the information in X conditioned by Y is equal to the information in X, with the interpretation that, on average, knowing a particular value of Y does not affect the information in X.

The notion of independence has been defined in algorithmic information theory as well, but for finite strings [6]. The approach is very similar. This time the information in a string x is the complexity (plain or prefix-free) of x, and two strings x and y are independent if the information in the join string $\langle x, y\rangle$ is equal to the sum of the information in x and the information in y, up to logarithmic (or, in some cases, constant) precision.

☆ An extended abstract has appeared in M. Ito, M. Toyama (Eds.), Developments in Language Theory (DLT'08), Lectures Notes in Comput. Sci. 5257, Springer-Verlag, Berlin, 2008, pp. 183–195.

* Corresponding author.
E-mail address: mzimand@towson.edu (M. Zimand).
URLs: http://www.cs.auckland.ac.nz/~cristian (C.S. Calude), http://triton.towson.edu/~mzimand (M. Zimand).

[1] Calude was supported in part by UARC Grant 3607894/9343 and CS-PBRF Grant.
[2] Zimand has been partially supported by NSF Grant CCF 0634830. Part of this work was done while visiting the CDMTCS of the University of Auckland, New Zealand.

0890-5401/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.ic.2009.05.004

The case of infinite sequences (in short, sequences) has been less studied. An inspection of the literature reveals that for this setting, independence has been considered to be synonymous with pairwise relative randomness, i.e., two sequences x and y are said to be independent if they are (Martin-Löf) random relative to each other (see [31,7]). The effect of this approach is that the notion of independence is confined to the situation where the sequences are random.

The main objective of this paper is to put forward a concept of independence that applies to *all* sequences, is natural, and is easy to use. One can envision various ways for doing this. One possibility is to use Levin's notion of mutual information for sequences [13] (see also the survey paper [10]) and declare two sequences to be independent if their mutual information is small.[3] We take another approach, which consists in extending in the natural way the notion of independence from finite strings to sequences. This leads us to two concepts: *independence* and *finitary-independence*. We say that (1) two sequences x and y are independent if, for all n, the complexity of $x\upharpoonright n$ (the prefix of x of length n) and the complexity of $x\upharpoonright n$ relativized with y are within $O(\log n)$ (and the same relation holds if we swap the roles of x and y), and (2) two sequences x and y are finitary-independent if, for all n and m, the complexity of $x\upharpoonright n$ and the complexity of $x\upharpoonright n$ given $y\upharpoonright m$ are within $O(\log n + \log m)$ (and the same relation holds if we swap the roles of x and y). We have settled for the additive logarithmical term of precision (rather than some higher accuracy) since this provides robustness with respect to the type of complexity (plain or prefix-free) and other technical advantages.

We establish a series of basic facts regarding the proposed notions of independence. We show that independence is strictly stronger than finitary-independence. The two notions of independence apply to a larger category of sequences than the family of random sequences, as intended. However, they are too rough for being relevant for computable sequences. It is not hard to see that a computable sequence x is independent with any other sequence y, simply because the information in x can be obtained directly. In fact, this type of trivial independence holds for a larger family of sequences, namely for any H-trivial sequence, and trivial finitary-independence holds for any sequence x whose prefixes have logarithmic complexity. It seems that for this type of sequences (computable or with very low complexity) a more refined definition of independence is needed (perhaps, based on resource-bounded complexity). We show that the two proposed notions of independence have some of the intuitive properties that one naturally expects. For example, for every sequence x, the set of sequences that are independent with x has measure one.

We next investigate to what extent pairs of independent, or finitary-independent sequences, can be effectively constructed via Turing reductions. For example, is there a Turing reduction f that given oracle access to an arbitrary sequence x produces a sequence that is finitary-independent with x? Clearly, if we allow the output of f to be a computable sequence, then the answer is positive by the type of trivial finitary-independence that we have noted above. We show that if we insist that the output of f has super-logarithmic complexity whenever x has positive constructive Hausdorff dimension, then the answer is negative. In the same vein, it is shown that there is no effective way of producing from an arbitrary sequence x with positive constructive Hausdorff dimension two sequences that are finitary-independent and have super-logarithmic complexity.

Similar questions are considered for the situation when we are given two (finitary-) independent sequences. It is shown that there are (finitary-) independent sequences x and y and a Turing reduction g such that x and $g(y)$ are not (finitary-) independent. We consider that this is the only counter-intuitive effect of our definitions. Note that the notion of constructive Hausdorff dimension (or of partial randomness) suffers from the same problem. For example, it is not hard to see that there exist a sequence x with constructive Hausdorff dimension 1 and a computable function g (which can even be a computable permutation of the input bits) such that $g(x)$ has constructive Hausdorff dimension 1/2. It seems that if one wants to extend the notion of independence to sequences that are not random (in particular to sequences that have arbitrary positive constructive Hausdorff dimension) such counter-intuitive effects cannot be avoided. On the other hand, for any independent sequences x and y and for any Turing reduction g, x and $g(y)$ are finitary-independent.

Our results show that partial random sequences can have complex structure: in particular, there are such sequences that cannot be obtained from random sequences by simple dilution operations (such as inserting a 0 between adjacent bits or doubling each bit).

We also raise the question on whether given as input several (finitary-) independent sequences x and y it is possible to effectively build a new sequence that is non-trivially (finitary-) independent with each sequence in the input. It is observed that the answer is positive if the sequences in the input are random, but for other types of sequences the question remains open. The same issue can be raised for finite strings and for this case a positive answer is obtained. Namely, it is shown that given three independent finite strings x, y and z with linear complexity, one can effectively construct a new string that is independent with each of x, y and z, has high complexity and length a constant fraction of the lengths of x, y and z.

1.1. Preliminaries

$\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^+$ denote, respectively, the set of non-negative integers, the set of real numbers, and the set of positive real numbers; the size of a finite set A is denoted $||A||$. Unless stated otherwise, all numbers are in $\mathbb{N}$ and all logs are in base 2. We work over the binary alphabet $\{0, 1\}$. A string is an element of $\{0, 1\}^*$ and a sequence is an element of $\{0, 1\}^\infty$. If x is a string, $|x|$ denotes its length; xy denotes the concatenation of the strings x and y. If x is a string or a sequence, $x(i)$ denotes the ith bit of x and $x\upharpoonright n$ is the substring $x(1)x(2)\cdots x(n)$. For two sequences x and y, $x \oplus y$ denotes the sequence

[3] We note that Levin's definition is technically very complicated and some basic questions remain open. For example, it is not even known whether, in the setting of [13], every sequence (excluding the trivial cases) is dependent with itself (see Problems 8.2 and 8.3 in [22]).

$x(1)y(1)x(2)y(2)x(3)y(3)\cdots$ and x XOR y denotes the sequence $(x(1)\text{ XOR } y(1))(x(2)\text{ XOR } y(2))(x(3)\text{ XOR } y(3))\cdots$, where $(x(i)\text{ XOR } y(i))$ is the sum modulo 2 of the bits $x(i)$ and $y(i)$. We identify a sequence x with the set $\{n \in \mathbb{N} \mid x(n) = 1\}$. We say that a sequence x is computable (computably enumerable, or c.e.) if the corresponding set is computable (respectively, computably enumerable, or c.e.). If x is c.e., then for every $s \in \mathbb{N}$, x_s is the sequence corresponding to the set of elements enumerated within s steps by some machine M that enumerates x (the machine M is given in the context). We also identify a sequence x with the real number in the interval $[0, 1]$ whose binary writing is $0.x(1)x(2)\ldots$ A sequence x is said to be left c.e. if the corresponding real number x is the limit of a computable increasing sequence of rational numbers. The plain and the prefix-free complexities of a string are defined in the standard way (for example see [2]); however we need to provide a few details regarding the computational models. The machines that we consider process information given in three forms: (1) the input, (2) the oracle set, (3) the conditional string. Correspondingly, a universal machine has three tapes:

- one tape for the input and work,
- one tape for storing the conditional string,
- one tape (called the oracle-query tape) for formulating queries to the oracle.

The oracle is a string or a sequence. If the machine enters the query state and the value written in binary on the oracle-query tape is n, then the machine gets the nth bit in the oracle, or if n is larger than the length of the oracle, the machine enters an infinite loop.

We fix such a universal machine U. The notation $U^w(u \mid v)$ means that the input is u, the conditional string v and the oracle is given by w, which is a string or a sequence. The plain complexity of a string x given the oracle w and the conditional string v is $C^w(x \mid v) = \min\{|u| \mid U^w(u \mid v) = x\}$. There exists a constant c such that for every x, v and w, $C^w(x \mid v) < |x| + c$.

A machine is prefix-free (self-delimiting) if its domain (i.e., the set of its input strings) is a prefix-free set. There exist universal prefix-free machines; we fix such a machine U (it will be clear from the context whether U is the universal plain machine or the universal prefix-free machine). The prefix-free complexity of a string x given the oracle w and the conditional string v is $H^w(x \mid v) = \min\{|u| \mid U^w(u \mid v) = x\}$.

In case w or v are the empty strings, we omit them in $C(\cdot)$ and $H(\cdot)$. The standard $O(\cdot)$, $o(\cdot)$, $\Omega(\cdot)$, $\omega(\cdot)$ notations for asymptotic upper and lower bounds are used throughout this paper. The reader should be aware that (a) in statements regarding strings, the $O(\cdot)$ notation hides constants that depend only on the choice of the universal machine underlying the definitions of the complexities C and H, and (b) in statements regarding prefixes of sequences, the hidden constants depend only on the involved sequences and on the underlying universal machines (but not on the lengths of the prefixes). Since the prefix-free universal machine is a particular type of machine, it follows that $C^w(x \mid v) < H^w(x \mid v) + O(1)$, for every x, v and w. The reverse inequality between $C(\cdot)$ and $H(\cdot)$ also holds true, within an additive logarithmic term, and is obtained as follows. For example, a string $x = x(1)x(2)\cdots x(n)$ can be coded in a self-delimiting way by

$$x \mapsto \text{code}(x) = \underbrace{11\cdots1}_{|\text{bin}(n)|}0\text{bin}(n)x(1)x(2)\cdots x(n), \tag{1}$$

where $\text{bin}(n)$ is the binary representation of $n \in \mathbb{N}$. Note that $|\text{code}(x)| = |x| + 2\log|x| + O(1)$. This implies that for every x, v, and w,

$$C^w(x \mid v) > H^w(x \mid v) - 2\log|x| - O(1). \tag{2}$$

The following inequalities hold for all strings x and y:

$$C^y(x) \le C(x \mid y) + 2\log|y| + O(1), \tag{3}$$

$$|C(xy) - (C(x \mid y) + C(y))| \le O(\log C(x) + \log C(y)). \tag{4}$$

The first inequality is easy to derive directly; the second one is called the Symmetry of Information Theorem, see [37].

Let M be a Turing machine whose domain is a prefix-free set. For each string x, let $Q_M(x) = \sum_{p, M(p)=x} 2^{-|p|}$ (the probability that M outputs x). The Coding Theorem (see [2,14,20]) states that if $Q_M(x) \ge 2^{-\ell}$, then $H(x) \le \ell + O(1)$ (the constant in $O(1)$ depends on the machine M).

There are various equivalent definitions for (algorithmic) random sequences as defined by Martin-Löf [17] (see [2]). In what follows we will use the (weak) complexity-theoretic one [5] using the prefix-free complexity: A sequence x is Martin-Löf random (in short, random) if there is a constant c such that for every n, $H(x\restriction n) \ge n - c$. The set of random sequences has constructive (Lebesgue) measure one [17].

The sequence x is random relative to the sequence y if there is a constant c such that for every n, $H^y(x\restriction n) \ge n - c$. Note that if x is random, then for every n, $C(x\restriction n) \ge n - 2\log n - O(1)$ (by inequality (2)). A similar inequality also holds for the relativized complexities, i.e. for all x that are random relative to y and for all n, $C^y(x\restriction n) > n - 2\log n - O(1)$. These results will be repeatedly used throughout the paper.

In [31] van Lambalgen proves that $x \oplus y$ is random iff x is random and y is random relative to x. This implies that if x is random and y is random relative to x then x is random relative to y.

The constructive Hausdorff dimension of a sequence x—which is the direct effectivization of "classical Hausdorff dimension"—defined by $\dim(x) = \liminf_{n\to\infty} C(x{\restriction}n)/n$ ($= \liminf_{n\to\infty} H(x{\restriction}n)/n$), measures intermediate levels of randomness (see [24,27,30,16,15,23,28,3,8,4]).

A Turing reduction f is an oracle Turing machine; $f(x)$ is the language computed by f with oracle x, assuming that f halts on all inputs when working with oracle x (otherwise we say that $f(x)$ does not exist). In other words, if $n \in f(x)$ then the machine f on input n and with oracle x halts and outputs 1 and if $n \notin f(x)$ then the machine f on input n and with oracle x halts and outputs 0. The function use is defined as follows: $\mathrm{use}_f^x(n)$ is the index of the rightmost position on the tape of f accessed during the computation of f with oracle x on all input strings of length n. The Turing reduction f is a *wtt-reduction* if there is a computable function q such that $\mathrm{use}_f^x(n) \le q(n)$, for all n. The Turing reduction f is a *truth-table reduction* if f halts on all inputs for every oracle. Every truth-table reduction is a wtt-reduction.

Let (E_n) be a sequence of events in some probability space. The Borel–Cantelli Lemma states that if the sum of the probabilities of the events E_n is finite then the probability that infinitely many of them hold is 0 (see for example [9]).

We use the following standard version of the Chernoff bounds (see for example Appendix A in [33]). Let $X_1, \ldots, X_n$ be independent random variables that take the values 0 and 1, let $X = \sum X_i$, and let μ be the expected value of X. Then for any $0 < d \le 1$, $\mathrm{Prob}(X > (1+d)\mu) \le e^{-d^2\mu/3}$.

2. Defining independence

The basic idea is to declare that two objects are independent if none of them contains significant information about the other one. Thus, if in some formalisation, $I(x)$ denotes the information in x and $I(x \mid y)$ denotes the information in x given y, x and y are independent if $I(x) - I(x \mid y)$ and $I(y) - I(y \mid x)$ are both small. In this paper we work in the framework of algorithmic information theory. In this setting, in case x is a string, $I(x)$ is the complexity of x (where for the "complexity of x" there are several possibilities, the main ones being the plain complexity or the prefix-free complexity).

The independence of strings was studied explicitly in [6][4]: two strings are independent if $I(xy) \approx I(x) + I(y)$. This approach motivates our Definitions 2.1 and 2.2.

The information in an infinite sequence x is characterised by the sequence $(I(x{\restriction}n))_{n\in\mathbb{N}}$ of information in the initial segments of x. The extra information given by an infinite sequence y appearing in $I(x \mid y)$ can be globally available in the form of an oracle or through its initial prefixes of arbitrary length.

Accordingly, we propose two notions of independence.

Definition 2.1 (*The global type of independence*)**.** Two sequences x and y are *independent* if, for every $n \in \mathbb{N}$, $C^x(y{\restriction}n) \ge C(y{\restriction}n) - O(\log n)$ and $C^y(x{\restriction}n) \ge C(x{\restriction}n) - O(\log n)$.

Definition 2.2 (*The finitary type of independence*)**.** Two sequences x, y are *finitary-independent* if for all natural numbers n and m,

$$C(x{\restriction}n\, y{\restriction}m) \ge C(x{\restriction}n) + C(y{\restriction}m) - O(\log(n) + \log(m)).$$

Recall that the constants hidden by the $O(\cdot)$ notation depend on the sequences x and y, but not on the lengths n and m.

Remark 2.3. We will show in Proposition 2.11, that the inequality in Definition 2.2 is equivalent to saying that for all n and m, $C(x{\restriction}n \mid y{\restriction}m) \ge C(x{\restriction}n) - O(\log n + \log m)$, which is the finite analogue of the property in Definition 2.1 and is in line with our discussion above.

Remark 2.4. If x and y are independent, then they are also finitary-independent (Proposition 2.12). The converse is not true (Corollary 4.18).

Remark 2.5. The proposed definitions use the plain complexity $C(\cdot)$, but we could have used the prefix-free complexity as well, because the two types of complexity are within an additive logarithmic term. Also, in Definition 2.2 (and throughout this paper), we use concatenation to represent the joining of two strings. However, since any reasonable pairing function $\langle x, y\rangle$ satisfies the inequality $|\,|\langle x, y\rangle| - |xy|\,| < O(\log|x| + \log|y|)$, it follows that $|C(\langle x, y\rangle) - C(xy)| < O(\log|x| + \log|y|)$, and thus any reasonable pairing function can be used instead.

Remark 2.6. A debatable issue is the quantification of precision. We chose the additive logarithmic term, but there are other natural possibilities. We argue that our choice has certain advantages over other possibilities that come to mind.

Let us focus on the definition of finitary-independence. We want $C(x{\restriction}n\, y{\restriction}m) \ge C(x{\restriction}n) + C(y{\restriction}n) - O(f(x) + f(y))$, for all n, m, where f should be some "small" function. We would like the following two properties to hold:

(A) the sequences x and y are finitary-independent iff $C(x{\restriction}n \mid y{\restriction}m) > C(x{\restriction}n) - O(f(x{\restriction}n) + f(y{\restriction}m))$, for all n and m,
(B) if x is "somehow" random and $y = 0^\omega$, then x and y are finitary-independent.

[4] Related notions such as mutual information and symmetry of information appear much earlier, for example in [37].

The additive logarithmic precision satisfies (A) and (B); robustness properties described in Remark 2.5 are also satisfied. Other natural possibilities for the definition are:
(i) If $f(x) = C(|x|)$, the definition of finitary-independence—(i) becomes:

$$C(x\upharpoonright n\, y\upharpoonright m) \geq C(x\upharpoonright n) + C(y\upharpoonright m) - O(C(n) + C(m)).$$

(ii) If $f(x) = \log C(x)$, the definition of finitary-independence—(ii) becomes:

$$C(x\upharpoonright n\, y\upharpoonright m) \geq C(x\upharpoonright n) + C(y\upharpoonright m) - O(\log C(x\upharpoonright n) + \log C(y\upharpoonright m)).$$

If sequences x and y satisfy (i), or (ii), then they also satisfy Definition 2.2.

Variant (i) implies (B), but not (A) (for example, consider sequences x and y with $C(n) \ll \log C(x\upharpoonright n)$ and $C(m) \ll \log C(y\upharpoonright m)$, for infinitely many n and m). Variant (ii) implies (A), but does not imply (B) (for example if for infinitely many n, $C(x\upharpoonright n) = \Omega(\log^3 n)$; take such a value n, let p be a shortest description of $x\upharpoonright n$, and let m be the integer whose binary representation is $1p$. Then $x\upharpoonright n$ and $0^\omega\upharpoonright m$, do not satisfy (B)).

Remark 2.7. If the sequence x is computable, then x is independent with every sequence y. In fact a stronger fact holds. A sequence is called H-trivial if, for all n, $H(x\upharpoonright n) \leq H(n) + O(1)$. This is a notion that has been intensively studied recently (see [8]). Clearly every computable sequence is H-trivial, but the converse does not hold [32,26]. If x is H-trivial, then it is independent with every sequence y. Indeed, $H^y(x\upharpoonright n) \geq H(x\upharpoonright n) - O(\log n)$, because $H(x\upharpoonright n) \leq H(n) + O(1) \leq \log n + O(1)$, and $H^x(y\upharpoonright n) \geq H(y\upharpoonright n) - O(\log n)$, because, in fact, $H^x(y\upharpoonright n)$ and $H(y\upharpoonright n)$ are within a constant of each other [19]. The same inequalities hold if we use the $C(\cdot)$ complexity (see Remark 2.5).

For the case of finitary-independence, a similar phenomenon holds for a (seemingly) even larger class.

Definition 2.8. A sequence x is called C-logarithmic if $C(x\upharpoonright n) = O(\log n)$.

It can be shown (for example using Proposition 2.11(a)) that if x is C-logarithmic, then it is finitary-independent with every sequence y.

Note that every sequence x that is the characteristic sequence of a c.e. set is C-logarithmic. This follows from the observation that, for every n, the initial segment $x\upharpoonright n$ can be constructed given the number of 1's in $x\upharpoonright n$ (an information which can be written with $\log n$ bits) and the finite description of the enumerator of the set represented by x. If a sequence is H-trivial then it is C-logarithmic, but the converse does not hold. Indeed, one can build a C-logarithmic sequence x such that $C(x\upharpoonright 2^{n+1} - 1) = \Theta(n)$. This can be done by taking, for each k that is a power of 2, one string y_k of length k with large complexity (say, $C(y_k) > k/2$). Let $A = \{y_1, y_2, y_4, y_8, \ldots,\}$ and finally x is the characteristic sequence of A. Since $H(2^{n+1} - 1) = O(\log n)$, x is not H-trivial.

In brief, the notions of independence and finitary-independence are relevant for strings having complexity above that of H-trivial sequences, respectively, C-logarithmic sequences. The cases of independent (finitary-independent) pairs (x, y), where at least one of x and y is H-trivial (respectively, C-logarithmic) will be referred to as *trivial independence*.

Remark 2.9. Some desirable properties of the independence relation are:

P1. Symmetry: x is independent with y iff y is independent with x.
P2. Robustness under type of complexity (plain or prefix-free).
P3. If f is a Turing reduction, except for some special cases, x and $f(x)$ are dependent ("independence cannot be algorithmically created").
P4. For every x, the set of sequences that are dependent with x is small (i.e., it has measure zero).

Clearly both the independence and the finitary-independence relations satisfy P1. They also satisfy P2, as we noted in Remark 2.5.

It is easy to see that the independence relation satisfies P3, whenever we require that the initial segments of x and $f(x)$ have plain complexity $\omega(\log n)$ (because $C^x(f(x)\upharpoonright n) = O(\log n)$, while $C(f(x)\upharpoonright n) = \omega(\log n)$). We shall see that the finitary-independence relation satisfies P3 under some stronger assumptions for f (see Section 4.1 and in particular Proposition 4.1).

Theorem 3.3 shows that the (finitary-) independence relation satisfies P4.

The following are parameterized versions of the definitions of independence and finitary-independence. They will be used to present some of our results in a more precise way.

Definition 2.10. Let $f : \mathbb{N} \Rightarrow \mathbb{N}$ be a function.

(a) Two sequences x and y are *f-independent* if, for every $n \in \mathbb{N}$, $C^x(y\upharpoonright n) \geq C(y\upharpoonright n) - O(f(n))$ and $C^y(x\upharpoonright n) \geq C(x\upharpoonright n) - O(f(n))$.
(b) Two sequences x, y are *f-finitary-independent* if for all natural numbers n and m,

$$C(x\upharpoonright n\, y\upharpoonright m) \geq C(x\upharpoonright n) + C(y\upharpoonright m) - O(f(n) + f(m)).$$

2.1. Properties of independent and finitary-independent sequences

The following simple properties of finitary-independent sequences are technically useful in some of the subsequent proofs.

Proposition 2.11

(a) *Two sequences x and y are finitary-independent iff for all n and m,* $C(x\restriction n \mid y\restriction m) \geq C(x\restriction n) - O(\log n + \log m)$.
(b) *Two sequences x and y are finitary-independent iff for all n,* $C(x\restriction n\, y\restriction n) \geq C(x\restriction n) + C(y\restriction n) - O(\log(n))$.
(c) *Two sequences x and y are finitary-independent iff for all n,* $C(x\restriction n \mid y\restriction n) \geq C(x\restriction n) - O(\log(n))$.
(d) *If the sequences x and y are not finitary-independent, then for every constant c there are infinitely many n such that* $C(x\restriction n\, y\restriction n) < C(x\restriction n) + C(y\restriction n) - c\log n$.
(e) *If the sequences x and y are not finitary-independent, then for every constant c there are infinitely many n such that* $C(x\restriction n \mid y\restriction n) < C(x\restriction n) - c\log n$.

Proof. We use the following inequalities which hold for all strings x and y (they follow from the Symmetry of Information Equation (4)):

$$C(xy) \geq C(x) + C(y \mid x) - O(\log|x| + \log|y|), \tag{5}$$

and

$$C(xy) \leq C(x) + C(y \mid x) + O(\log|x| + \log|y|). \tag{6}$$

(a) "⇒"

$$\begin{aligned} C(x\restriction n \mid y\restriction m) &\geq C(x\restriction n\, y\restriction m) - C(y\restriction m) - O(\log n + \log m) \quad \text{(by (6))} \\ &\geq C(x\restriction n) + C(y\restriction m) - C(y\restriction m) - O(\log n + \log m) \quad \text{(by independence)} \\ &= C(x\restriction n) - O(\log n + \log m). \end{aligned}$$

"⇐"

$$\begin{aligned} C(x\restriction n\, y\restriction m) &\geq C(y\restriction m) + C(x\restriction n \mid y\restriction m) - O(\log n + \log m) \quad \text{(by (5))} \\ &\geq C(y\restriction m) + C(x\restriction n) - O(\log n + \log m) \quad \text{(by hypothesis)}. \end{aligned}$$

(b) "⇒" Take $n = m$.

"⇐" Suppose $n \geq m$ (the other case can be handled similarly).

$$\begin{aligned} C(x\restriction n\, y\restriction m) &\geq C(y\restriction m) + C(x\restriction n \mid y\restriction m) - O(\log(n) + \log(m)) \quad \text{(by (5))} \\ &\geq C(y\restriction m) + C(x\restriction n \mid y\restriction n) - O(\log(n) + \log(m)) \\ &\geq C(y\restriction m) + C(x\restriction n) - O(\log(n) + \log(m)) \quad \text{(by (a))}. \end{aligned}$$

(c) This follows from (b) with a similar proof as for (a).

(d) Suppose that for some constant c the inequality holds only for finitely many n. Then one can choose a constant $c' > c$ for which the opposite inequality holds for every n, which by (b) would imply the finitary-independence of x and y.

(e) Follows from (c), in a similar way as (d) follows from (b). □

Proposition 2.12. *If the sequences x and y are independent, then they are also finitary-independent.*

Proof. Suppose x and y are not finitary-independent. By Proposition 2.11(e), for every constant c there are infinitely many n such that $C(x\restriction n \mid y\restriction n) < C(x\restriction n) - c \cdot \log n$. Taking into account inequality (3), we obtain $C^y(x\restriction n) < C(x\restriction n) - (c-3)\log n$, for infinitely many n, which contradicts that x and y are independent. □

We show in Corollary 4.18 that the converse of Proposition 2.12 does not hold.

Proposition 2.13. *For any real number* σ *and all sequences x and y, if* $\dim(x) = \sigma$ *and* (x, y) *are finitary-independent, then* $\dim(x \text{ XOR } y) \geq \sigma$.

Proof. Note that $C(x\restriction n \mid y\restriction n) \le C((x \text{ XOR } y)\restriction n) + O(1)$, for all n (this holds for all sequences x and y). Suppose there exists $\varepsilon > 0$ such that $\dim(x \text{ XOR } y) \le \sigma - \varepsilon$. It follows that, for infinitely many n, $C((x \text{ XOR } y)\restriction n) \le (\sigma - \varepsilon)n$. Then

$$\begin{aligned} C(x\restriction n \mid y\restriction n) &< C((x \text{ XOR } y)\restriction n) + O(1) \\ &\le (\sigma - \varepsilon)n + O(1) \qquad \text{for infinitely many } n. \end{aligned}$$

By the finitary-independence of (x, y), $C(x\restriction n) \le C(x\restriction n \mid y\restriction n) + O(\log n) \le (\sigma - \varepsilon)n + O(1)$, i.o. n, which contradicts the fact that $\dim(x) = \sigma$. □

Proposition 2.14

(a) *If the sequence x is random and the sequences (x, y) are finitary-independent, then $(y, x \text{ XOR } y)$ are finitary-independent.*
(b) *If the sequence x is random and the sequences (x, y) are independent, then $(y, x \text{ XOR } y)$ are independent.*

Proof. For (a) suppose that y and x XOR y are not finitary-independent. Then for every constant c, there are infinitely many n, such that $C((x \text{ XOR } y)\restriction n \mid y\restriction n) < C((x \text{ XOR } y)\restriction n) - c \log n$. Note that if a program can produce $(x \text{ XOR } y)\restriction n$ given $y\restriction n$, then by doing an extra bitwise XOR with $y\restriction n$ it will produce $x\restriction n$. Thus, $C(x\restriction n \mid y\restriction n) < C((x \text{ XOR } y)\restriction n \mid y\restriction n) + O(1)$ for all n. Combining with the first inequality, for every constant c and for infinitely many n we have:

$$\begin{aligned} C(x\restriction n \mid y\restriction n) &< C((x \text{ XOR } y)\restriction n) - c \log n + O(1) \\ &< n - c \log n + O(1) \\ &< C(x\restriction n) + 2 \log n - c \log n + O(1) \\ &= C(x\restriction n) - (c - 2) \log n + O(1). \end{aligned}$$

This contradicts the fact that x and y are finitary-independent. The proof for (b) is similar. □

Proposition 2.15. *There are sequences x, y, and z that are pairwise independent, but $(x, y \oplus z)$ are not finitary-independent.*

Proof. Take y and z two sequences that are random relative to each other, and let $x = y$ XOR z. Then (x, y) are independent, and (x, z) are independent, by Proposition 2.14. On the other hand note that $\dim(y \text{ XOR } z) = 1$ (by Proposition 2.13) and $C((y \text{ XOR } z)\restriction n \mid (y \oplus z)\restriction 2n) < O(1)$. Consequently, for every constant c and for almost every n, $C((y \text{ XOR } z)\restriction n \mid (y \oplus z)\restriction 2n) < C((y \text{ XOR } z)\restriction n) - c(\log n + \log 2n)$, and thus, $(y \text{ XOR } z, y \oplus z)$ are not finitary-independent.
□

In Remark 2.7, we have listed several types of sequences, with computability-related properties, that are independent or finitary-independent with any other sequence. The next result goes in the opposite direction: a computability-related property is identified that implies non-finitary-independence (and thus non-independence).

Proposition 2.16 ([29]). *If x and y are left c.e. sequences, $\dim(x) > 0$, and $\dim(y) > 0$, then x and y are not finitary-independent.*

Proof. Let $x(1), \dots, x(n), \dots$ (respectively, $y(1), \dots, y(n), \dots$) be a computational and increasing sequence of rational limits such that $\lim_{n\to\infty} x(n) = x$ ($\lim_{n\to\infty} y(n) = y$). In case x (respectively, y) is rational we let $x(i) = x$ ($y(i) = y$) for every i. For each n, let $\mathrm{cm}_x(n) = \min\{s \mid x(s)\restriction n = x\restriction n\}$ and $\mathrm{cm}_y(n) = \min\{s \mid y(s)\restriction n = y\restriction n\}$ (the convergence moduli of x and, respectively, y). Without loss of generality we can assume that $\mathrm{cm}_x(n) \ge \mathrm{cm}_y(n)$, for infinitely many n. For each n satisfying the inequality, $y\restriction n$ can be computed from $x\restriction n$ as follows. First compute $s = \mathrm{cm}_x(n)$ (which can be done because $x\restriction n$ is known) and output $y(s)\restriction n$. Consequently, for infinitely many n, $C(y\restriction n \mid x\restriction n) < O(1)$. On the other hand, since $\dim(y) > 0$, there exists a constant c such that $C(y\restriction n) \ge c \cdot n$, for almost every n. Consequently, x and y are not finitary-independent. □

3. Examples of independent and finitary-independent sequences

We give more examples of pairs of sequences that are independent or finitary-independent but not trivially independent (see Remark 2.7).

Theorem 3.1. *Let x be a random sequence and let y be a sequence that is random relative to x. Then x and y are independent.*

Proof. Since y is random relative to x, for all n, $C^x(y\restriction n) > n - 2 \log n - O(1) \ge C(y\restriction n) - 2 \log n - O(1)$. The van Lambalgen's Theorem [31] implies that x is random relative to y as well. Therefore, in the same way, $C^y(x\restriction n) > n - 2 \log n - O(1) \ge C(x\restriction n) - O(\log n)$. □

From Theorem 3.1 we can easily derive examples of pairs (x, y) that are independent and which have constructive Hausdorff dimension ε, for every rational $\varepsilon > 0$. For example, if we start with x and y that are random with respect to each other and build $x' = x(1)\,0x(2)\,0\ldots$ (i.e. we insert 0s in the even positions) and similarly build y' from y, then x' and y' have constructive Hausdorff dimension equal to $1/2$ and are independent (because $C^{x'}(y'{\restriction}n)$ and $C^{x}(y{\restriction}(n/2))$ are within a constant of each other, as are $C(y'{\restriction}n)$ and $C(y{\restriction}(n/2))$). The pairs of sequences from Theorem 3.1 (plus those derived from them as above) and the trivially independent sequences from Remark 2.7 are the only examples of independent sequences that we know. Thus, currently, we have examples of independent pairs (x, y) only for the case when x has maximal prefix-free complexity (i.e., x is random) or x is obtained via a simple dilution transformation (similar to inserting 0s in even positions) from a random sequence, and for the case when x has minimal prefix-free complexity (i.e., x is H-trivial). The paucity of examples should be contrasted with Theorem 3.3 which shows that for every sequence x, most sequences y are independent with x. Pairs of sequences that are finitary-independent are easier to find.

Theorem 3.2. *Let x be an arbitrary sequence and let y be a sequence that is random relative to x. Then x and y are finitary-independent.*

Proof. Suppose x and y are not finitary-independent. Then there are infinitely many n with $C(y{\restriction}n \mid x{\restriction}n) < C(y{\restriction}n) - 5\log n$. Consider a constant c_1 satisfying $C(y{\restriction}n) < n + c_1$, for all n. We get (under our assumption) that, for infinitely many n. $C(y{\restriction}n \mid x{\restriction}n) < n - 5\log n + c_1$. Then, by inequality (3), for infinitely many n, $C^{x{\restriction}n}(y{\restriction}n) < n - 3\log n + c + c_1$. Note that for every n and every $m \geq n$, $C^{x{\restriction}m}(y{\restriction}n) < C^{x{\restriction}n}(y{\restriction}n)$ (here we use the fact that the oracle query mechanism does not allow the machine to find the length of the oracle string). Thus, for infinitely many n and for all $m \geq n$,

$$C^{x{\restriction}m}(y{\restriction}n) < n - 3\log n + (c + c_1). \tag{7}$$

On the other hand, y is random relative to x. Therefore, for all n, $H^x(y{\restriction}n) > n - O(1)$. Let U be the universal machine underlying the complexity $H(\cdot)$ and let p^* be the shortest program such that $U^x(p^*) = y{\restriction}n$ (if there are ties, take p^* to be the lexicographically smallest among the tying programs). Let $m(n) = \max(n, \mathrm{use}(U^x(p^*)))$. Note that, for all n, $H^x(y{\restriction}n) = H^{x{\restriction}m(n)}(y{\restriction}n)$. It follows that, for every n, $H^{x{\restriction}m(n)}(y{\restriction}n) = H^x(y{\restriction}n) > n - O(1)$. Recall that for every pair of strings u and v, $C^v(u) > H^v(u) - 2\log|u| - O(1)$. Thus, for every n,

$$C^{x{\restriction}m(n)}(y{\restriction}n) > n - 2\log n - O(1). \tag{8}$$

Inequalities (7) and (8) are contradictory. □

Theorem 3.3. (a) *For every sequence x, the set $\{y \mid y$ independent with $x\}$ has measure one.*
(b) *For every sequence x, the set $\{y \mid y$ finitary-independent with $x\}$ has measure one.*

Proof. Clearly, (a) implies (b) (because the set in (a) is a subset of the set in (b)). Thus, we only have to prove (a). We show that the sets

$$\begin{aligned} C_1 &= \{y \in \{0,1\}^\infty \mid \forall n, H^x(y{\restriction}n) \geq H(y{\restriction}n) - O(\log n)\},\\ C_2 &= \{y \in \{0,1\}^\infty \mid \forall n, H^y(x{\restriction}n) \geq H(x{\restriction}n) - O(\log n)\} \end{aligned}$$

have both measure one, from which the conclusion follows because the set of sequences independent with x is $C_1 \cap C_2$.

The set C_1 has measure one because it includes the set of sequences y that are random relative to x.

We next focus on C_2. It is enough to show that the set

$$C_3 = \{y \in \{0,1\}^\infty \mid H^y(x{\restriction}n) \leq H(x{\restriction}n) - 4\log n \text{ i.o. } n\}$$

has measure 0, because C_3 contains the complement of C_2. The following claim (whose proof we postpone for the moment) holds:

Claim 3.4. *For every $v \in \{0,1\}^*$ and every $k, \ell \in \mathbb{N}$, if the set $A = \{y \in \{0,1\}^\infty \mid H^y(v) \leq k\}$ has measure at least $2^{-\ell}$, then $H(v) \leq k + \ell + 2\log k + O(1)$.*

Note that **Claim** 3.4 implies that if n is sufficiently large, then the measure of the set $\{y \in \{0,1\}^\infty \mid H^y(x{\restriction}n) < H(x{\restriction}n) - 4\log n\}$ is less than $2^{-2\log n}$, which, by the Borel–Cantelli Lemma (see [9]), implies that C_3 has measure zero.

We continue with the *proof* of **Claim** 3.4. For a string $\sigma \in \{0,1\}^*$, let $[\sigma]$ denote the set of sequences that have σ as initial prefix. We denote the Lebesgue measure by μ (in particular, $\mu[\sigma] = 2^{-|\sigma|}$; see more in [2]) and U is the fixed universal prefix-free Turing machine underlying the relativized complexity $H^y(v)$.

The intuition is that since $\mu(A) \geq 2^{-\ell}$, there should be some string σ of length at most ℓ such that $[\sigma] \subseteq A$ and therefore v can be described by σ and a string p of length at most k such that $U^{\sigma}(p) = v$. This intuition does not hold because it can happen that $\mu(A) \geq 2^{-\ell}$ due to the fact that there are many σ as above (i.e., whose extensions are all in A) of length $> \ell$. To fix this problem, we need to use the Coding Theorem, and this necessitates some preparations.

For each string p, we define a c.e. set A_p (essentially, A_p is a prefix-free set of oracle prefixes σ such that $U^{\sigma}(p) \downarrow$ and such that any sequence y with $U^{y}(p) \downarrow$ has a prefix in A_p; $U^{\sigma}(p) \downarrow$ and $U^{y}(p) \downarrow$ denote that the two computations halt).

Construction of A_p. Initially, at Step 0, $A_p = \emptyset$. At Step s (with $s = 1, 2, \ldots,$) we enumerate some strings in A_p as follows:

Step s. Let Candidates be the set of all strings σ with $|\sigma| \leq s$. If some prefix or some extension of σ has been already enumerated in A_p, then σ is removed from the set Candidates. Next, for all $\sigma \in$ Candidates, in lexicographical order, run $U^{\sigma}(p)$ for s steps. If $U^{\sigma}(p)$ stops in s steps and no prefix of σ has been enumerated already in A_p, then enumerate σ in A_p.

End of construction of A_p.

By construction, A_p is c.e. and prefix-free. We next define a Turing machine M. For any strings p and σ, we denote $\langle p, \sigma\rangle = \text{code}(p)\sigma$ (code(p) was defined in Eq. (1)). The machine M on input $\langle p, \sigma\rangle$ outputs $U^{\sigma}(p)$, provided $\sigma \in A_p$ (on all other inputs M is not defined). Note that the domain of M is a prefix-free set. For any string v, we define

$$Q_M(v) = \sum_{\{\langle p,\sigma\rangle | M(\langle p,\sigma\rangle)=v\}} 2^{-|\langle p,\sigma\rangle|}.$$

Then,

$$\begin{aligned} Q_M(v) &= \textstyle\sum_p \sum_{\{\sigma | M(\langle p,\sigma\rangle)=v\}} 2^{-(|\text{code}(p)+|\sigma|)} \\ &\geq \textstyle\sum_{p,|p|\leq k} \sum_{\{\sigma | M(\langle p,\sigma\rangle)=v\}} 2^{-(|\text{code}(p)+|\sigma|)} \\ &= \textstyle\sum_{p,|p|\leq k} 2^{-|\text{code}(p)|} \sum_{\{\sigma | M(\langle p,\sigma\rangle)=v\}} 2^{-|\sigma|} \\ &\geq 2^{-(k+2\log k+1)} \textstyle\sum_{p,|p|\leq k} \sum_{\{\sigma | M(\langle p,\sigma\rangle)=v\}} 2^{-|\sigma|}. \end{aligned}$$

For each $y \in A$, there exists a string τ, minimal under taking prefixes, and a string p with $|p| \leq k$, such that $y \in [\tau]$ and $U^{\tau}(p) = v$. We call τ the root of y. If B is the set of strings τ that are roots for at least a $y \in A$, $\mu(A) = \sum_{\tau \in B} 2^{-|\tau|}$. It can be seen that for each $\tau \in B$, there must be some p with $|p| \leq k$ such that A_p contains a subset $A_{p,\tau}$ of extensions of τ such that $2^{-|\tau|} = \sum_{\sigma \in A_{p,\tau}} 2^{-|\sigma|}$ (The reason for this is the following: Let s be the number of steps in which $U^{\tau}(p) \downarrow$; then, at Step s, all the suffixes of τ of length s are in the set Candidates; therefore, by the end of Step s, all these suffixes are in A_p because the computation of U on input p with oracle any of these suffixes will stop in exactly s steps.). Note that for any two roots $\tau \neq \tau'$, $A_{p,\tau}$ and $A_{p,\tau'}$ are disjoint. Therefore,

$$\sum_{p,|p|\leq k} \sum_{\{\sigma | M(\langle p,\sigma\rangle)=v\}} 2^{-|\sigma|} \geq \sum_{\tau \in B} 2^{-|\tau|} = \mu(A) \geq 2^{-\ell}.$$

It follows that $Q_M(x) \geq 2^{-(k+\ell+2\log k+1)}$. By the Coding Theorem, it follows that $H(x) \leq k + \ell + 2\log k + O(1)$. □

Thus there are many (in measure-theoretical sense) pairs of sequences that are independent. But is it possible to have such pairs satisfying a given constraint? We consider one instance of this general question.

Proposition 3.5. *If x is a random sequence then there are sequences y and z such that (y, z) are independent and $x = y$ XOR z.*

Proof. Take a sequence y independent with x. Then, by Proposition 2.14, y and (x XOR y) are independent. By taking $z = x$ XOR y, it follows that $x = y$ XOR z, with y and z independent. □

4. Effective constructions of finitary-independent sequences

In this section we investigate to what extent it is possible to effectively construct sequences that are independent or finitary-independent. We show some impossibility results and therefore we focus on the weaker type of independence, finitary-independence (clearly, if it is not possible to produce a pair of sequences that are finitary-independent, then it is also not possible to produce a pair of sequences that are independent). Since a C-logarithmic sequence is finitary-independent with any other sequence, the issue of constructibility is interesting if we also require that the sequences have complexity above that of C-logarithmic sequences (see Remark 2.7). Such sequences are of course non-computable, and therefore the whole issue of constructibility appears to be a moot point. However this is not so if we assume that we already have in hand one (or several) non-computable sequence(s), and we want to build additional sequences that are finitary-independent. Informally speaking, we investigate the following questions:

Question (a). Is it possible to effectively construct from a sequence x another sequence y finitary-independent with x, where the finitary-independence is not trivial (recall Remark 2.7)? This question has two variants depending on whether we seek a uniform procedure (i.e., one procedure that works for all x), or whether we allow the procedure to depend on x.

Question (b). Is it possible to effectively construct from a sequence x two sequences y and z that are finitary-independent (not in the trivial way)? Again, there are uniform and non-uniform variants of this question.

We analyse these questions in Section 4.1. Similar questions for the case when the input consists of two sequences x_1 and x_2 are tackled in Section 4.2.

4.1. Producing independence with one source

We first consider the uniform variant of Question (a): Is there a Turing reduction f such that for all $x \in \{0,1\}^*$, $(x, f(x))$ are finitary-independent? We even relax the requirement and demand that f should achieve this objective only if x has positive constructive Hausdorff dimension (this only makes the following impossibility results stronger).

As discussed above, we first eliminate some trivial instances of this question. Without any requirement on the algorithmic complexity of the desired $f(x)$, the answer is trivially YES because we can take $f(x) = 0^\omega$ (or any other computable sequence). Even if we only require that $f(x)$ is not computable, then the answer is still trivially YES because we can make $f(x)$ to be C-logarithmic. For example, consider

$$f(x) = x(1)\, x(2)0\, x(3)000 \ldots\; x(k)\, \underbrace{0\ldots0}_{2^{k-1}-1} \ldots .$$

Then $f(x)$ is C-logarithmic, but not computable provided x is not computable, and $(x, f(x))$ are finitary-independent simply because $f(x)$ is C-logarithmic.

As noted above, the question is interesting if we require $f(x)$ to have some "significant" amount of randomness whenever x has some "significant" amount of randomness. We expect that in this case the answer should be negative, because, intuitively, one should not be able to effectively produce independence (this is property P3 in Remark 2.9).

We consider two situations depending on two different meanings of the concept of "significant" amount of randomness.

Case 1: We require that $f(x)$ is not C-logarithmic (i.e., for any constant c, $f(x) > c\log(x)$, for infinitely many x). We do not solve the uniform version of Question (a) in this case, but we show that every reduction f that potentially does the job must have non-polynomial use.

Proposition 4.1. *Let f be a Turing reduction. For every sequence x, if the function $\mathrm{use}_f^x(n)$ is polynomially bounded, then x and $f(x)$ are not finitary-independent, unless one of them is C-logarithmic.*

Proof. Let y be $f(x)$. Then for every n, let $m(n) = \max_{k\le n}(\mathrm{use}_f^x(k))$. Then $y\restriction n$ depends only on $x\restriction m(n)$ and $m(n)$ is polynomial in n. Then $C(y\restriction n \mid x\restriction m(n)) \le O(\log n)$. If x and y were finitary-independent, then $C(y\restriction n) \le C(y\restriction (n) \mid x\restriction m(n)) + O(\log n + \log m(n)) \le O(\log(n)) + \log(m(n)) \le O(\log n)$, for all n, i.e., y would be C-logarithmic. □

Case 2: We require that $f(x)$ has complexity just above that of C-logarithmic sequences (in the sense below). We show that in this case, the answer to the uniform variant of Question (a) is negative: there is no such f. The following definition introduces a class of sequences having complexity just above that of C-logarithmic sequences.

Definition 4.2. A sequence x is C-superlogarithmic if for every constant $c > 0$, $C(x\restriction n) > c\log n$, for almost every n.

The next proofs use the following facts.

Fact 4.3 (Variant of Theorem 3.1 in [21]). *For all rationals $0 \le \alpha < \beta < 1$, and for every infinite and computable set S, there exists a sequence x such that $\dim(x) = \alpha$ and for all wtt-reductions f, either $f(x)$ does not exist or $C(f(x)\restriction n) \le \beta n$, for infinitely many n in S.*

Fact 4.4 (Variant of Theorem 3.1 in [1]). *For every Turing reduction h, for all rationals $0 < \alpha < \beta < 1$, and for every infinite and computable set S, there is a sequence x with $\dim(x) \ge \alpha$ such that either $h(x)$ does not exist or $C(h(x)\restriction n) < \beta n$, for infinitely many n in S.*

Fact 4.5 ([18]). *For every rational $0 \le \alpha \le 1$, there exists a sequence x such that $\dim(x) = \alpha$ and for all Turing-reductions f,* $\dim(f(x)) \le \alpha$.

Fact 4.6 (Corollary 1 in ([34]). *For any $\alpha > 0$, there is a truth-table reduction f such that if the input sequences x and y are finitary-independent and $\dim(x) \ge \alpha$ and $\dim(y) \ge \alpha$, then $\dim f(x, y) = 1$.*

Fact 4.7 (Theorem 3 in ([34]). *For any* $\delta > 0$, *there exist a constant* c, *an infinite and computable set* S, *and a truth-table reduction* $f : \{0, 1\}^\infty \times \{0, 1\}^\infty \to \{0, 1\}^\infty$ (*i.e.*, f *is a Turing machine with two oracles*) *with the following property*:

If the input sequences x *and* y *are finitary-independent and satisfy* $C(x\upharpoonright n) > c \cdot \log n$ *and* $C(y\upharpoonright n) > c \cdot \log n$, *for almost every* n, *then the output* $z = f(x, y)$ *satisfies* $C(f(x, y)\upharpoonright n) > (1 - \delta) \cdot n$, *for almost every* n *in* S.

Theorem 3.1 in [21] is for $S = \mathbb{N}$ (and is stronger in that $\alpha = \beta$) but its proof can be modified in a straightforward manner to yield Fact 4.3. Theorem 3.1 in [1] is also for $S = \mathbb{N}$ and can also be modified in a simple manner—using Fact 4.3—to yield Fact 4.4.

We can now state the impossibility results related to *Case 2*. To simplify the structure of quantifiers in the statement of the following result, we posit here the following task, depending on a parameter $\alpha \in \mathbb{R}$, for a function f mapping sequences to sequences:

TASK A: Let $0 < \alpha < 1$. For every $x \in \{0, 1\}^\infty$ with $\dim(x) \geq \alpha$, the following should hold:

(a) $f(x)$ exists,
(b) $f(x)$ is C-superlogarithmic,
(c) x and $f(x)$ are finitary-independent.

Theorem 4.8. *For every* $\alpha \in (0, 1)$, *there is no Turing reduction* f *that satisfies TASK A*.

Proof. Let us fix $\alpha \in (0, 1)$. Suppose there exists f satisfying (a), (b) and (c) in TASK A for this parameter α. Let S be the infinite, computable set and let g be the truth-table reduction promised by Fact 4.7 for $\delta = (1 - \alpha)/3$. Let h be the Turing reduction $h(x) = g(x, f(x))$. Let x^* be the sequence promised by Fact 4.4 for α, $\beta = (1 + \alpha)/2$, and the above set S and Turing reduction h. On one hand, by Fact 4.4, $C(h(x^*)\upharpoonright n) < ((1 + \alpha)/2)n$, for infinitely many $n \in S$. On the other hand, by Fact 4.7, $C(h(x^*)\upharpoonright n) > ((2 + \alpha)/3)n$, for almost every $n \in S$. We have reached a contradiction. □

We next consider the uniform variant of *Question (b)*.

First we remark that, by van Lambalgen's Theorem [31], if the sequence x is random, then x_{even} and x_{odd} are random relative to each other (where x_{odd} is $x(1)x(3)x(5)\ldots$ and x_{even} is $x(2)x(4)x(6)\ldots$). Thus, x_{even} and x_{odd} are certainly independent.

Kautz [11] has shown a much more general result by examining the splittings of sequences obtained using bounded Kolmogorov–Loveland selection rules.[5] He showed that if x is a random sequence, x_0 is the subsequence of x obtained by concatenating the bits of x chosen by an arbitrary bounded Kolmogorov–Loveland selection rule, and x_1 consists of the bits of x that were not selected by the selection rule, then x_0 and x_1 are random with respect to each other (and thus independent).

We show that the similar result for sequences with constructive Hausdorff dimension $\sigma \in (0, 1)$ is not valid. In fact, our next result is stronger, and essentially gives a negative answer to the uniform variant of *Question (b)*.

We posit the following task, depending on a parameter $\alpha \in \mathbb{R}$, for two functions f_1 and f_2 mapping sequences to sequences:

TASK B: Let $0 < \alpha < 1$. For every $x \in \{0, 1\}^\infty$ with $\dim(x) \geq \alpha$, the following should hold:

(a) $f_1(x)$ and $f_2(x)$ exist,
(b) $f_1(x)$ and $f_2(x)$ are C-superlogarithmic,
(c) $f_1(x)$ and $f_2(x)$ are finitary-independent.

Theorem 4.9. *For every* $\alpha \in (0, 1)$, *there are no Turing reductions* f_1 *and* f_2 *satisfying TASK B*.

Proof. Similar to the proof of Theorem 4.8. □

The non-uniform variants of *Questions (a)* and *(b)* remain open. In the particular case when (i) f is a wtt-reduction or (ii) the output has positive constructive Hausdorff dimension, we present impossibility results analogous to those in Theorems 4.8 and 4.9. The proofs are similar, with the difference that for (i) we use Fact 4.3 instead of Fact 4.4, and for (ii) we use Fact 4.5 instead of Fact 4.4 and Fact 4.6 instead of Fact 4.7.

Theorem 4.10. *For every rational* $\sigma \in (0, 1)$, *there exists a sequence* x *with* $\dim(x) = \sigma$ *such that for every wtt-reduction* f, *at least one of the following holds true*:

(a) $f(x)$ *does not exist*,
(b) $f(x)$ *is not finitary-independent with* x,
(c) $f(x)$ *is not* C-*superlogarithmic*.

[5] A Kolmogorov–Loveland selection rule is an effective process for selecting bits from a sequence. Informally, it is an iterative process and at each step, based on the bits that have been already read, a new bit from the sequence is chosen to be read and (before that bit is actually read) the decision on whether that bit is selected or not is taken. A *bounded* Kolmogorov–Loveland selection rule satisfies a certain requirement of monotonocity for deciding the selected bits, see [11].

Theorem 4.11. *For every rational* $\sigma \in (0, 1)$, *there exists a sequence* x *with* $\dim(x) = \sigma$ *such that for every wtt-reductions* f_1 *and* f_2, *at least one of the following holds true*:

(a) $f_1(x)$ *does not exist or* $f_2(x)$ *does not exist*,
(b) $f_1(x)$ *and* $f_2(x)$ *are not finitary-independent*,
(c) $f_1(x)$ *is not C-superlogarithmic or* $f_2(x)$ *is not C-superlogarithmic*.

Theorem 4.12. *For every rational* $\sigma \in (0, 1)$, *there exists a sequence* x *with* $\dim(x) = \sigma$ *such that for every Turing-reduction* f, *at least one of the following holds true*:

(a) $f(x)$ *does not exist*,
(b) $f(x)$ *is not finitary-independent with* x,
(c) $\dim(f(x)) = 0$.

Theorem 4.13. *For every rational* $\sigma \in (0, 1)$, *there exists a sequence* x *with* $\dim(x) = \sigma$ *such that for every Turing-reduction* f_1 *and* f_2, *at least one of the following holds true*:

(a) $f_1(x)$ *does not exist or* $f_2(x)$ *does not exist*,
(b) $f_1(x)$ *and* $f_2(x)$ *are not finitary-independent*,
(c) $\dim(f_1(x)) = 0$ *or* $\dim(f_2(x)) = 0$.

Theorem 4.11 has an interesting implication regarding sequences with constructive Hausdorff dimension in the interval $(0, 1)$. Suppose, for example, that we want to construct a sequence with constructive Hausdorff dimension $1/2$. The first idea that comes to mind is to take a random sequence $x = x(1)x(2)\cdots$ and either consider the sequence $y = x(1)0x(2)0\ldots$ (we insert 0s in all even positions) or the sequence $z = x(1)x(1)x(2)x(2)\cdots$ (we double every bit). The sequences y and z have constructive Hausdorff dimension $1/2$ and they have been obtained by diluting a random sequence. In a similar way, once we have sequences with positive constructive Hausdorff dimension, we can combine them in reversible ways to obtain new such sequences ("reversible" means that from the output of the procedure it is possible to effectively obtain the inputs). Theorem 4.11 shows, roughly speaking, that there are sequences with dimension strictly between 0 and 1, that are not dilutions of random sequences or reversible combinations of sequences with positive constructive Hausdorff dimension. Formally, for every rational $\sigma \in (0, 1)$, there is a sequence x with $\dim(x) = \sigma$ so that no matter what wtt method we use for selecting from x two subsequences, either one of the resulting subsequences has low complexity or the two resulting subsequences are not independent.

Theorems 4.12 and 4.13 can be strengthened to the case of $g(n)$-finitary-independence, for any function $g \in o(n) \cap \Omega(\log n)$. More precisely, for any function $g \in o(n) \cap \Omega(\log n)$, one can replace "finitary-independent" by "$g(n)$-finitary-independent" in Theorems 4.12 and 4.13. The proofs are identical, except that instead of using Fact 4.6, one uses the following recent result from [35] (the result in [35] is slightly stronger).

Fact 4.14 (Theorem 4.1 in [35])**.** *For every* $0 < \tau \le 1$, *for every* $\delta > 0$, *for every function* $g \in o(n) \cap \Omega(\log n)$, *there exist* $\alpha \in (0, 1)$ *and a truth-table reduction* f *such that for any sequences* x *and* y *with dimension* $\ge \tau$ *that are* $g(n)$*-finitary-independent, the sequence* $f(x, y)$ *has dimension* $\ge (1 - \delta)$.

4.2. Producing independence with two sources

We have seen some limits on the possibility of constructing a finitary-independent sequences starting from one sequence. What if we are given two finitary-independent sequences: is it possible to construct from them more finitary-independent sequences?

First we observe that if x and y are two (finitary-) independent sequences and g is an arbitrary Turing reduction, then it does not necessarily follow that x and $g(y)$ are also (finitary-) independent (as one may expect). On the other hand, if x and y are independent, it does follow that x and $g(y)$ are finitary-independent.

Proposition 4.15. (a) [29] *There are two independent sequences* x *and* y *and a Turing reduction* g *such that* x *and* $g(y)$ *are not independent*.

(b) [25] *There are two finitary-independent sequences* x *and* y *and a Turing reduction* g *such that* x *and* $g(y)$ *are not finitary-independent*.

Proof. (a) Let z be a random sequence and let u, v, and w be sequences such that $z = u \oplus (v \oplus w)$. By van Lambalgen's Theorem [31], each of the sequences u, v, and w are random relative to the join of the other two. We define the sequences x and y as follows:

$$\begin{aligned} x(2^n) &= u(n), \text{ for all } n \in \mathbb{N} \\ x(m) &= v(m), \text{ for every } m \text{ that is not a power of 2} \\ y(2^n) &= u(n), \text{ for all } n \in \mathbb{N} \\ y(m) &= w(m), \text{ for every } m \text{ that is not a power of 2} \end{aligned}$$

Claim 4.16. *The sequences x and y are independent.*

Proof. Suppose x and y are not independent. Then, similarly to Proposition 2.11(e), for infinitely many n, $C^x(y\restriction n) < C(y\restriction n) - 7\log n$. Then

$$\begin{aligned} C^{u\oplus v}(w\restriction n) &\le C^{u\oplus v}(y\restriction n) + 2\log n + O(1) \\ &\quad \text{(because } w\restriction n \text{ and } y\restriction n \text{ differ in only } \log n \text{ bits)} \\ &\le C^x(y\restriction n) + 2\log n + O(1) \\ &\quad \text{(because queries to } x \text{ can be replaced by queries to } u \text{ and } v\text{)} \\ &\le C(y\restriction n) - 7\log n + 2\log n + O(1), \\ &\quad \text{for infinitely many } n \\ &\le C(w\restriction n) + 2\log n - 7\log n + 2\log n + O(1) \\ &= C(w\restriction n) - 3\log n + O(1) \\ &\le n - 3\log n + O(1). \end{aligned}$$

This contradicts that the fact that w is random with respect to $u \oplus v$. □

We continue the *Proof* of Proposition 4.15(a). It is easy to define a Turing reduction g such that $g(y) = u$. Notice that $C^x(u\restriction n) = O(\log n)$, because u is many-one reducible to x. On the other hand $C(u\restriction n) \ge n - 2\log n + O(1)$, for every n, because u is random. Therefore x and $g(y)$ are not independent.

(b) Let u and y be two sequences that are random relative to each other. Let x be the sequence defined by $x(2^{2^n}) = y(n)$, for all n, and $x(m) = u(m)$, for all m not of the form 2^{2^n}. Note that x and y are finitary-independent, because the complexities of $u\restriction n$ and $x\restriction n$ differ by at most $O(\log\log n)$.

Let g be the Turing reduction such that for all n, $g(x)(n) = x(2^{2^n})$. Note that $g(x)$ and y coincide and thus, obviously are not finitary-independent. □

Proposition 4.17. *If the sequences x and y are independent, and g is a Turing reduction, then x and $g(y)$ are finitary-independent (provided $g(y)$ exists).*

Proof. Since x and y are independent, there exists a constant c such that for all n,

$$C^y(x\restriction n) \ge C(x\restriction n) - c\log n.$$

Suppose that x and $g(y)$ are not finitary-independent. Then there are infinitely many n such that $C(x\restriction n \mid g(y)\restriction n) < C(x\restriction n) - (c+4)\log n$. Since $C^y(x\restriction n) \le C(x\restriction n \mid g(y)\restriction n) + 2\log n + O(1)$, it would follow that, for infinitely many n,

$$C^y(x\restriction n) \le C(x\restriction n) - (c+1)\log n,$$

which contradicts the first inequality. □

Corollary 4.18. *There are sequences that are finitary-independent but not independent.*

Proof. The sequences x and $g(y)$ from Proposition 4.15 are not independent, but they are finitary-independent by Proposition 4.17. □

If x and y are (finitary-) independent sequences, then there exist simple procedures that starting with the pair (x, y), produce a new pair of (finitary-) independent sequences. For example, (x, y_{odd}) is such a pair.

A more challenging question is whether given a pair of (finitary-) independent sequences (x, y), it is possible to effectively produce another sequence that is (finitary-) independent with x, and with y. We give a positive answer for the case when x and y are both random. The similar question for non-random x and y remains open (but see Section 4.3).

Theorem 4.19. *There exists an effective transformation f with polynomially-bounded use such that if x and y are random and independent (respectively, finitary-independent), then $(x, f(x, y))$ and $(y, f(x, y))$ are pairs of independent (respectively, finitary-independent) sequences, and the independence (respectively, finitary-independence) is not trivial (recall Remark 2.7).*

Proof. We take $f(x, y) = x$ XOR y and take into account Proposition 2.14. □

Remark: Contrast with Proposition 4.1, where we have shown that for every x, for every effective transformation f with polynomially-bounded use, x and $f(x)$ are not finitary-independent.

4.3. Producing independence: the finite case

In the previous section we discussed the issue of whether given as input several sequences that are (finitary-) independent, there is an effective way to construct a sequence that is (finitary-) independent with each sequence in the input (and the independence is not trivial). A result of this type is obtained for the case when the input consists of two random sequences x and y in Theorem 4.19. We do not know if in Theorem 4.19 we can remove the assumption that x and y are random.

In what follows we will consider the simpler case of strings. In this setting we are able to give a positive answer for the situation when we start with three[6] input strings that are independent (and not necessarily random). First we define the analogue of independence for strings.

Definition 4.20. Let $c \in \mathbb{R}^+$ and $k \in \mathbb{N}$. We say that the strings $x_1, x_2, \ldots, x_k$ in $\{0,1\}^*$ are c-independent if

$$C(x_1 x_2 \ldots x_k) \geq C(x_1) + C(x_2) + \cdots + C(x_k) - c(\log|x_1| + \log|x_2| + \cdots + \log|x_k|).$$

The main result of this section is the following theorem, whose proof draws from the techniques of [34].

Theorem 4.21. *For all constants $\sigma > 0$ and $\sigma_1 \in (0, \sigma)$, there exists a computable function $f : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ with the following property: For every $c \in \mathbb{R}^+$ there exists $c' \in \mathbb{R}^+$ such that if the input consists of a triplet of c-independent strings having sufficiently large length n and plain complexity at least $\sigma \cdot n$, then the output is c'-independent with each element in the input triplet and has length $\lfloor \sigma_1 n \rfloor$.*

More precisely, if

(i) (x, y, z) *are c-independent,*
(ii) $|x| = |y| = |z| = n$, *and*
(iii) $C(x) \geq \sigma \cdot n$, $C(y) \geq \sigma \cdot n$, $C(z) \geq \sigma \cdot n$,

then, provided n is large enough, the following pairs of strings $(f(x, y, z), x)$, $(f(x, y, z), y)$, $(f(x, y, z), z)$ are c'-independent, $|f(x, y, z)| = \lfloor \sigma_1 n \rfloor$, and $C(f(x, y, z)) \geq \lfloor \sigma_1 n \rfloor - O(\log n)$.

Before we delve into the proof, we establish several preliminary facts.

Lemma 4.22. *If x_1, x_2, x_3 are three strings that are c-independent, then*

$$C(x_1 \mid x_2 x_3) \geq C(x_1) - (c+2)(\log|x_1| + \log|x_2| + \log|x_3|) - O(1).$$

Proof. The following inequalities hold for every three strings and in particular for the strings x_1, x_2, and x_3:

$$C(x_1 x_2 x_3) \leq C(x_2 x_3) + C(x_1 \mid x_2 x_3) + 2\log|x_1| + O(1),$$

and

$$C(x_2 x_3) \leq C(x_2) + C(x_3) + 2\log|x_2| + O(1).$$

Then

$$\begin{aligned} C(x_1 \mid x_2 x_3) &\geq C(x_1 x_2 x_3) - C(x_2 x_3) - 2\log|x_1| - O(1) \\ &\geq C(x_1) + C(x_2) + C(x_3) - c(\log|x_1| + \log|x_2| + \log|x_3|) \\ &\quad - (C(x_2) + C(x_3) + 2\log|x_2| + O(1)) - 2\log|x_1| - O(1) \\ &\geq C(x_1) - (c+2)(\log|x_1| + \log|x_2| + \log|x_3|) - O(1). \end{aligned}$$ □

The next lemma establishes a combinatorial fact about the possibility of colouring the cube $[N] \times [N] \times [N]$ with M colours such that every sufficiently large planar rectangle contains all the colours in about the same proportion. Here N and M are natural numbers, $[K]$ (with K natural) denotes the set $\{1, 2, \ldots, K\}$, and a planar rectangle is a subset of $[N] \times [N] \times [N]$ having one of the following three forms: $B_1 \times B_2 \times \{k\}$, $B_1 \times \{k\} \times B_2$, or $\{k\} \times B_1 \times B_2$, where $k \in [N]$, $B_1 \subseteq [N]$ and $B_2 \subseteq [N]$.

[6] (Added at revision). The case when the input consists of two independent strings has been recently solved in [35]. However the result in [35] only achieves $\sigma_1 \approx \sigma/2$. For further investigation of this issue see also [36].

Lemma 4.23. *Let $0 < \sigma_1 < \sigma_2 < 1$. For every sufficiently large n it is possible to colour the cube $[2^n] \times [2^n] \times [2^n]$ with $M = 2^{\lfloor \sigma_1 n \rfloor}$ colours in such a way that every planar rectangle satisfying $\|B_1\| = a \cdot 2^{\lceil \sigma_2 n \rceil}$ and $\|B_2\| = b \cdot 2^{\lceil \sigma_2 n \rceil}$ for some natural numbers a and b contains at most $(2/M)\|B_1\|\|B_2\|$ occurrences of colour c, for every colour $c \in [M]$.*

Proof. We use the probabilistic method. Let $N = 2^n$. We colour each cell of the $[N] \times [N] \times [N]$ cube with one colour chosen independently and uniformly at random from $[M]$. For $i, j, k \in [N]$, let $T(i, j, k)$ be the random variable that designates the colour of the cell (i, j, k) in the cube. For every fixed cell (i, j, k) and for every fixed colour $c \in [M]$, $\text{Prob}(T(i, j, k) = c) = 1/M$, because the colours are assigned independently and uniformly at random. Let us first consider some fixed subsets B_1 and B_2 of $[N]$ having size $2^{\lceil \sigma_2 n \rceil}$, a fixed $k \in [N]$, and a fixed colour $c \in [M]$. Let A be the event "the fraction of occurrences of c in the planar rectangle $B_1 \times B_2 \times \{k\}$ is greater than $2/M$." Using the Chernoff bounds (the standard version indicated in Section 1.1), it follows that

$$\text{Prob}(A) < e^{-(1/3)(1/M)N^{2\sigma_2}}.$$

The same upper bounds hold for the probabilities of the similar events regarding the planar rectangles $B_1 \times \{k\} \times B_2$ and $\{k\} \times B_1 \times B_2$. Thus, if we consider the event B "there is some colour with a fraction of appearances in one of the three planar rectangles mentioned above greater than $(2/M)$", then, by the union bound,

$$\text{Prob}(B) < 3M \cdot e^{-(1/3)(1/M)N^{2\sigma_2}}. \tag{9}$$

The number of ways to choose $B_1 \subseteq [N]$ with $\|B_1\| = 2^{\lceil \sigma_2 n \rceil}$, $B_2 \subseteq [N]$ with $\|B_2\| = 2^{\lceil \sigma_2 n \rceil}$ and $k \in [N]$ is approximately (ignoring truncation) $\binom{N}{N^{\sigma_2}} \cdot \binom{N}{N^{\sigma_2}} \cdot N$, which is bounded by

$$e^{2N^{\sigma_2}} \cdot e^{2N^{\sigma_2}(1-\sigma_2)\ln(N)} \cdot e^{\ln N}, \tag{10}$$

(we have used the inequality $\binom{n}{k} < (en/k)^k$). Clearly, for our choice of M, (10) times the right hand side in (9) is less than 1. It means that there exists a colouring where no colour appears a fraction larger than $(2/M)$ in every planar rectangle with B_1 and B_2 having size exactly $2^{\lceil \sigma_2 n \rceil}$. For planar rectangles having the sizes of B_1 and B_2 an integer multiple of $2^{\lceil \sigma_2 n \rceil}$, the assertion holds as well because such rectangles can be partitioned into subrectangles having the size exactly $2^{\lceil \sigma_2 n \rceil}$. □

Proof (*of Theorem 4.21*)**.** We take n sufficiently large so that all the following inequalities hold. Let x^*, y^* and z^* be a triplet of strings of length n satisfying the assumptions in the statement. Let $N = 2^n$ and let us consider a constant $\sigma_2 \in (\sigma_1, \sigma)$. By exhaustive search we find the minimal (in some canonical ordering) colouring $T : [N] \times [N] \times [N] \to [M]$ satisfying the properties in Lemma 4.23. Identifying the strings x^*, y^* and z^* with their indices in the lexicographical ordering of $\{0, 1\}^n$, we define $w^* = T(x^*, y^*, z^*)$. Note that the length of w^* is $\log M = \lfloor \sigma_1 n \rfloor$, which we denote by m. We will show that $C(w^* \mid z^*) \geq m - c' \log m$, for $c' = 3c + d + 13$, where d is a constant that will be specified later. Since $C(w^*) \leq m + O(1)$, it follows that w^* and z^* are independent. In a similar way, it can be shown that w^* and x^* are independent, and w^* and y^* are independent.

For the sake of obtaining a contradiction, suppose that $C(w^* \mid z^*) < m - c' \log m$. The set $A = \{w \mid C(w \mid z^*) < m - c' \log m\}$ has size $< 2^{m - c' \log m}$ and, by our assumption, contains w^*.

Let t_1 be such that $C(x^*) = t_1$ and t_2 be such that $C(y^* \mid z^*) = t_2$. Note that $t_1 \geq \sigma n > \sigma_2 n$. The integer t_2 is also larger than $\sigma_2 n$, because $C(y^* \mid z^*) \geq C(y^* \mid z^* x^*) - 2 \log n - O(1) \geq C(y^*) - (c+4)(3 \log n) - O(1) \geq \sigma n - (3c + 12) \log n - O(1) > \sigma_2 n$. For the second inequality we have used Lemma 4.22.

Let $B_1 = \{x \in \{0, 1\}^n \mid C(x) \leq t_1\}$. Note that the size of B_1 is bounded by 2^{t_1+1}. We take a set B_1' including B_1 having size exactly 2^{t_1+1}. Similarly, let $B_2 = \{y \in \{0, 1\}^n \mid C(y \mid z^*) \leq t_2\}$ and let B_2' be a set that includes B_2 and has size exactly 2^{t_2+1}. Let k be the index of z^* in the lexicographical ordering of $\{0, 1\}^n$. By Lemma 4.23, it follows that for every $a \in [M]$,

$$\|T^{-1}(a) \cap (B_1' \times B_2' \times \{k\})\| \leq (2/M)\|B_1'\|\|B_2'\|.$$

Consequently,

$$\begin{aligned}
\|T^{-1}(A) \cap (B_1 \times B_2 \times \{k\})\| &\leq \|T^{-1}(A) \cap (B_1' \times B_2' \times \{k\})\| \\
&= \textstyle\sum_{a \in A} \|T^{-1}(a) \cap (B_1' \times B_2' \times \{k\})\| \\
&< 2^{m - c' \log m} \cdot (2/2^m)\|B_1'\|\|B_2'\| \\
&= 2^{t_1 + t_2 + 3 - c' \log m}.
\end{aligned}$$

Note that given z^*, $m - c' \log m$, t_1 and t_2, and σ_1 and σ_2, we can enumerate $T^{-1}(A) \cap (B_1 \times B_2 \times \{k\})$ (the table T can be constructed from n, σ_1 and σ_2, and n can be retrieved from z^*). Since (x^*, y^*, z^*) is in this set, it follows that the complexity of $x^* y^*$ given z^* is bounded by the rank of the triplet (x^*, y^*, z^*) in a fixed enumeration of the set and the information needed to perform the enumeration. Thus,

$$\begin{aligned} C(x^* y^* \mid z^*) &\leq t_1 + t_2 + 3 - c' \log m + 2\log(m - c' \log m) + 2\log t_1 + 2\log t_2 + O(1) \\ &\leq t_1 + t_2 - (c' - 2)\log m + 2\log t_1 + 2\log t_2 + O(1). \end{aligned}$$

On the other hand, by the conditional version of the Symmetry of Information Equation (4), there exists a constant d such that for all strings u, v, w, $C(uv \mid w) \geq C(v \mid w) + C(u \mid uw) - d(\log |uv|)$. It follows that

$$\begin{aligned} C(x^* y^* \mid z^*) &\geq C(y^* \mid z^*) + C(x^* \mid y^* z^*) - d\log n - O(1) \\ &\geq t_2 + t_1 - (c+2)(3\log n) - d\log n - O(1) \\ &= t_1 + t_2 - (3c + d + 6)\log n - O(1). \end{aligned}$$

For the second inequality we have used Lemma 4.22. Note that $t_1 < n + O(1)$ and $t_2 < n + O(1)$ and $m = \sigma_1 n$. Combining the above inequalities, we obtain $(c' - 2)\log \sigma_1 n \leq (3c + d + 10)\log n + O(1)$. Since $c' = 3c + d + 13$, we have obtained a contradiction. □

Acknowledgments

We are grateful to André Nies and Frank Stephan for their insightful comments. In particular, Definition 2.1 has emerged after several discussions with André, and Propositions 2.16 and 4.15 are due to Frank [29]. We thank Alexander Shen for suggesting Theorem 3.3 and Proposition 4.15(b). We thank Jan Reimann for his assistance with establishing Fact 4.3. We thank Joe Miller for bringing to our attention the paper [18]. We thank one anonymous referee whose suggestions helped improve the quality of the presentation. This referee suggested a C-logarithmic sequence which is not H-trivial.

References

[1] L. Bienvenu, D. Doty, F. Stephan, Constructive dimension and Turing degrees, in: Computation and Logic in the Real World—Third Conference of Computability in Europe, Lecture Notes in Computer Science, vol. 4497, Springer-Verlag, pp. 63–72. Available as Technical Report arXiv:cs/0701089 ar arxiv.org.
[2] C.S. Calude, Information and Randomness: An Algorithmic Perspective, Revised and Extended, second ed., Springer-Verlag, Berlin, 2002
[3] C.S. Calude, L. Staiger, S. Terwijn, On partial randomness, Annals of Pure and Applied Logic 138 (2006) 20–30.
[4] C.S. Calude, N.J. Hay, F.C. Stephan, Representation of left-computable ε-random reals, CDMTCS Research Report 365, 2009.
[5] G. Chaitin, A theory of program size formally identical to information theory, Journal of the ACM 22 (1975) 329–340.
[6] G. Chaitin, Gödel's theorem and information, International Journal of Theoretical Physics 21 (1982) 941–954.
[7] R. Downey, D. Hirschfeldt, Algorithmic Randomness and Complexity, Springer, in press.
[8] R. Downey, D. Hirschfeldt, A. Nies, S. Terwijn, Calibrating randomness, The Bulletin of Symbolic Logic 12 (3) (2006) 411–492.
[9] W. Feller, Introduction to Probability Theory and its Applications, third ed., John Wiley & Sons, 1968.
[10] P. Grünwald, P. Vitanyi, Shannon information and Kolmogorov complexity, 2004, CORR Technical Report arXiv:cs.IT/0410002, revised May 2006.
[11] S.M. Kautz, Independence properties of algorithmically random sequences, 2003, CORR Technical Report arXiv:cs/0301013.
[12] L. Levin, Laws of information conservation and aspects of the foundation of probability theory, Prob. Inf. Transmission 10 (1974) 206–210.
[13] L. Levin, Randomness conservation inequalities: information and independence in mathematical theories, Information and Control 61 (1) (1984) 15–37.
[14] M. Li, P. Vitanyi, An Introduction to Kolmogorov Complexity and its Applications, Springer-Verlag, Berlin, 1993
[15] J. Lutz, The dimensions of individual strings and sequences, Information and Control 187 (2003) 49–79.
[16] E. Mayordomo, A Kolmogorov complexity characterization of constructive Hausdorff dimension, Information Processing Letters 84 (2002) 1–3.
[17] P. Martin-Löf, The definition of random sequences, Information and Control 9 (1966) 602–619.
[18] J. Miller, Extracting information is hard: a Turing degree of non-integral effective Hausdorff dimension, Advances in Mathematics, in press. Available from: <http://www.math.uconn.edu/josephmiller/Papers/dimension.pdf>.
[19] A. Nies, Lowness properties and randomness, Advances in Mathematics 197 (2005) 274–305.
[20] A. Nies, Computability and Randomness, Oxford University Press, 2009.
[21] A. Nies, J. Reimann, A lower cone in the wtt degrees of non-integral effective dimension, in: C. Chong, Q. Feng, Th. Slaman, W.H. Woodin, Y. Yang (Eds.), Proceedings of IMS Workshop on Computational Prospects of Infinity, Singapore, 2006, pp. 249–260.
[22] Open Problem List, AIM/ARCC Workshop in Effective Randomness, Updated December 17, 2007 at <http://www.math.dartmouth.edu/frg/aimRandQsLive.pdf.>
[23] J. Reimann, Computability and Fractal Dimension, Technical Report, Ph.D. Thesis, Universität Heidelberg, 2004.
[24] B. Ryabko, Coding of combinatorial sources and Hausdorff dimension, Doklady Akademii Nauk SSR 277 (1984) 1066–1070.
[25] A. Shen, Private Communication, June 2008.
[26] R. Solovay, Draft of a paper (or series of papers) on Chaitin's work, 1975, unpublished manuscript, IBM Thomas J. Watson Research Center, 215 pp.
[27] L. Staiger, Kolmogorov complexity and Hausdorff dimension, Inform. and Comput. 103 (1993) 159–194.
[28] L. Staiger, Constructive dimension equals Kolmogorov complexity, Information Processing Letters, 93 (2005) 149–153. Preliminary version: Research Report CDMTCS-210, Univ. of Auckland, January 2003.
[29] F. Stephan, Email Communication, May 2007.
[30] K. Tadaki, A generalization of Chaitin's halting probability Ω and halting self-similar sets, Hokkaido Math. J. 31 (2002) 219–253.
[31] M. van Lambalgen, The axiomatization of randomness, Journal of Symbolic Logic 55 (1990) 1143–1167.
[32] D. Zambella, On sequences with simple initial segments, ILLC Technical Report, ML 1990-05, University of Amsterdam.
[33] M. Zimand, Computational Complexity: A Quantitative Perspective, North-Holland Mathematics Studies, Elsevier, 2004.
[34] M. Zimand, Two sources are better than one for increasing the Kolmogorov complexity of infinite sequences, in: E.A. Hirsch, A.A. Razborov, A.L. Semenov, A. Slissenko (Eds.), Proceedings CSR 2008, Moscow, Russia, June 7–12, 2008, Lecture Notes in Computer Science, vol. 5010, Springer, Berlin, 2008, pp. 326–338.

[35] M. Zimand, Extracting the Kolmogorov complexity of strings and sequences from sources with limited independence, in: Susanne Albers, Jean-Yves Marion (Eds.), Proceedings 26th STACS, Freiburg, Germany, February 26–29, 2009, pp. 697–708.
[36] M. Zimand, On generating independent random strings, 2009, in: Proceedings CiE 2009, Heidelberg, in press. Available as CORR Technical Report. arXiv:0903.3889.
[37] A. Zvonkin, L. Levin, The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms, Russian Mathematical Surveys 25 (6) (1970) 83–124.