Algorithmically Independent Sequences

Cristian S. Calude[1,⋆] and Marius Zimand[2,⋆⋆]

[1] Department of Computer Science, University of Auckland, New Zealand
www.cs.auckland.ac.nz/ ~cristian
[2] Department of Computer and Information Sciences, Towson University, Baltimore, MD, USA
http://triton.towson.edu/~mzimand

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 for finite strings [Cha82]. Our approach is very similar. This time the information in a string x is the complexity (plain or prefix-free) of x,

⋆ Calude was supported in part by UARC Grant 3607894/9343 and CS-PBRF Grant.
⋆⋆ Zimand was supported by NSF grant CCF 0634830. Part of this work was done while visiting the CDMTCS of the University of Auckland, New Zealand.

M. Ito and M. Toyama (Eds.): DLT 2008, LNCS 5257, pp. 183–195, 2008.

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.

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 [vL90, DH]). As a consequence, 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. One can envision various ways for doing this. One possibility is to use Levin's notion of mutual information for sequences [Lev84] (see also the survey paper [GV04]) and declare two sequences to be independent if their mutual information is small. If one pursues this direction, the main issue is to determine the right definition for "small." 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 $x\restriction n$ (the prefix of x of length n) and the complexity of $x\restriction n$ 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 $x\restriction n$ and the complexity of $x\restriction n$ given $y\restriction m$ 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 type 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. This appears to be the only counter-intuitive effect of our definitions. Note that the definition of constructive Hausdorff dimension (or of partial randomness) suffers from the same problem. For example, there exist a sequence x with constructive Hausdorff dimension 1 and a computable g such that $g(x)$ has constructive Hausdorff dimension $\leq 1/2$. It seems that if one wants to extend the notion of independence to non random sequences (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.

We also raise the question on whether given as input finitely many (finitary-) independent sequences it is possible to effectively build a new sequence that is (finitary-) independent (in a non-trivial way) 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 regarding 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 its length is a constant fraction of the length of x, y and z.

Because of space limitations, this extended abstract contains no proof. All proofs are available in the full version of the paper [CZ07].

1.1 Preliminaries

Let $\mathbb{N}$ denote the set of non-negative integers; 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 with 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 i-th bit of x and $x\restriction n$ is the substring $x(1)x(2)\cdots x(n)$. For two sequences x and y, $x \oplus y$ denotes the sequence $x(1)y(1)x(2)y(2)x(3)y(3)\cdots$ and x XOR y denotes the sequence $(x(1)$ XOR $y(1))(x(2)$ XOR $y(2))(x(3)$ XOR $y(3))\cdots$, where $(x(i)$ 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 (given) machine M that enumerates x. We also identify a sequence x with the real number in the interval $[0,1]$ whose binary writing is $0.x(1)x(2)\cdots$. 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; 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 3 tapes: (i) one tape for the input and work, (ii) one tape for storing the conditional string, (iii) 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 n-th 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 is v and the oracle is 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 is a prefix-free set. There exist universal prefix-free machines. We fix such a machine U; 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 $H(\cdot)$. Throughout this paper we use the $O(\cdot)$ notation to hide constants that depend only on the choice of the universal machine underlying the definitions of the complexities C and H. There are various equivalent definitions for (algorithmic) random sequences as defined by Martin-Löf [ML66] (see [C02]). In what follows we will use the (weak) complexity-theoretic one [Cha75] 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, $H(x\restriction n) \geq n - c$. The set of random sequences has constructive (Lebesgue) measure one [ML66]. The sequence x is random relative to the sequence y if there is a constant c such that for every n, $H^y(x\restriction n) \geq n - c$.

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 [Rya84, Sta93, Tad02, May02, Lut03, Rei04], [Sta05, CST06, DHNT06]).

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 oracle x halts and outputs 1 and if

$n \notin f(x)$ then the machine f on input n and oracle x halts and outputs 0. The function *use* is defined as follows: $use_f^x(n)$ is the index of the rightmost position on the tape of f accessed during the computation of f with oracle x on input n. The Turing reduction f is a *wtt-reduction* if there is a computable function q such that $use_f^x(n) \leq q(n)$, for all n. The Turing reduction f is a *truth-table reduction* if f halts on all inputs for every oracle. A truth-table reduction is a wtt-reduction.

2 Defining Independence

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 in [Cha82]: two strings are independent if $I(xy) \approx I(x) + I(y)$. This approach motivates our Definition 1 and Definition 2.

In case x is an infinite sequence, the information in x is characterised by the sequence $(I(x\upharpoonright n))_{n\in\mathbb{N}}$ of information in the initial segments of x. For the information upon which we condition (e.g., the y in $I(x \mid y)$), there are two possibilities: either the entire sequence is available in the form of an oracle, or we consider initial segments of it. Accordingly, we propose two notions of independence.

Definition 1. (**The "integral" type of independence**) *Two sequences x and y are* independent *if $C^x(y\upharpoonright n) \geq C(y\upharpoonright n) - O(\log n)$ and $C^y(x\upharpoonright n) \geq C(x\upharpoonright n) - O(\log n)$.*

Definition 2. (**The finitary type of independence**) *Two sequences x, y are* finitary-independent *if for all natural numbers n and m,*

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

Remark 1. We will show in Proposition 1, that the inequality in Definition 2 is equivalent to saying that for all n and m, $C(x\upharpoonright n \mid y\upharpoonright m) \geq C(x\upharpoonright n) - O(\log n + \log m)$, which is the finite analogue of the property in Definition 1 and is in line with our discussion above.

Remark 2. If x and y are independent, then they are also finitary-independent (Proposition 2). The converse is not true (Corollary 1).

Remark 3. 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 (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 $\mid |\langle x, y\rangle| - |xy| \mid < O(\log |x| + \log |y|)$, it follows that $|C(< x, y >) - C(xy)| < O(\log |x| + \log |y|)$, and thus any reasonable pairing function could be used instead.

Remark 4. A debatable issue is the subtraction of the logarithmic term. Indeed, 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 $C(x\restriction n\ y\restriction m) \geq C(x\restriction n) + C(y\restriction n) - O(f(x) + f(y))$, 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 "somewhat" random and $y = 00\cdots000\cdots$, then x and y are finitary-independent.

Other natural possibilities for the definition could be:
(i) if $f(x) = C(|x|)$, the definition of finitary independence–(i) would be:

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

or (ii) if $f(x) = \log C(x)$, the definition of finitary-independence–(ii) would be:

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

If sequences x and y satisfy (i), or (ii), then they also satisfy Definition 2.

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

Another advantage is the robustness discussed in Remark 3.

Remark 5. 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, $H(x\restriction n) \leq H(n) + O(1)$. This is a notion that has been intensively studied recently (see [DHNT06]). Clearly every computable sequence is H-trivial, but the converse does not hold [Zam90, Sol75]. If x is H-trivial, then it is independent with every sequence y. Indeed, $H^y(x\restriction n) \geq H(x\restriction n) - O(\log n)$, because $H(x\restriction n) \leq H(n) + O(1) \leq \log n + O(1)$, and $H^x(y\restriction n) \geq H(y\restriction n) - O(\log n)$, because, in fact, $H^x(y\restriction n)$ and $H(y\restriction n)$ are within a constant of each other [Nie05]. The same inequalities hold if we use the $C(\cdot)$ complexity (see Remark 3).

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

Definition 3. *A sequence x is called C-logarithmic if $C(x\restriction n) = O(\log n)$.*

It can be shown (for example using Proposition 1, (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 $x\restriction n$ can be constructed given the number of 1's in $x\restriction n$ (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 probably does not hold.

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 6. 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 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 3. 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 $C^x(f(x)\restriction n) = O(\log n)$, while $C(f(x)\restriction n) = \omega(\log n)$). We shall see that the finitary-independence relation satisfies P3 under some stronger assumptions for f and $f(x)$ (see Section 4.1 and in particular Theorem 6). Theorem 3 shows that the (finitary-) independence relation satisfies P4.

2.1 Properties of Independent and Finitary-Independent Sequences

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

Proposition 1. (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 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 x and y are not finitary-independent, then for every constant c there are infinitely many n such that $C(x\restriction n \mid y\restriction n) < C(x\restriction n) - c\log n$.*

Proposition 2. *If the sequences x and y are independent, then they are also finitary-independent.*

Proposition 3. *If $\dim(x) = \sigma$ and the sequences (x, y) are finitary-independent, then $\dim(x \text{ XOR } y) \geq \sigma$.*

Proposition 4. *(a) If x is random and the sequences (x, y) are finitary-independent, then $(y, x \text{ XOR } y)$ are finitary-independent.*
(b) If x is random and (x, y) are independent, then $(y, x \text{ XOR } y)$ are independent.

Proposition 5. *There are sequences x, y, and z such that (x, y) are independent, (x, z) are independent, but $(x, y \oplus z)$ are not finitary-independent.*

In Remark 5, we have listed several types of sequences that are independent or finitary-independent with any other sequence. The next result goes in the opposite direction: it exhibits a pair of sequences that can not be finitary-independent (and thus not independent).

Proposition 6. [Ste07] *If x and y are left c.e. sequences, $\dim(x) > 0$, and $\dim(y) > 0$, then x and y are not finitary-independent.*

3 Examples of Independent and Finitary-Independent Sequences

We give examples of pairs of sequences that are independent or finitary-independent (other than the trivial examples from Remark 5).

Theorem 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.*

Theorem 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.*

As expected, most pairs of sequences are independent (and thus also finitary-independent).

Theorem 3. *For every x, the set $\{y \mid y \text{ independent with } x\}$ has (Lebesgue) measure one.*

4 Effective Constructions of Finitary-Independent Sequences

The examples of (finitary-) independent sequences provided so far are existential (i.e., non-constructive). In this section we investigate to what extent it is possible

to effectively construct such sequences. We show some impossibility results and therefore we focus on the weaker type of independence, finitary-independence. 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 5)? 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, where the finitary-independence is not trivial? 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 discussed in Section 4.2.

4.1 If We Have 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 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. The answer should be negative, because, intuitively, one should not be able to produce independence (this is property P3 in Remark 6).

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. We do not solve the question, but we show that every reduction f that potentially does the job must have non-polynomial *use*.

Proposition 7. *Let f be a Turing reduction. For every sequence x, if the function $use_f^x(n)$ is polynomially bounded, then x and $f(x)$ are not finitary-independent, unless one of them is 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.

Definition 4. *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 theorems in this section are based on results from [NR06], [BDS07], and [Zim08].

We proceed to 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 for a function f mapping sequences to sequences:

TASK A: for every $x \in \{0,1\}^\infty$ with $\dim(x) > 0$, the following should hold: (a) $f(x)$ exists, (b) $f(x)$ is C-superlogarithmic, (c) x and $f(x)$ are finitary-independent.

Theorem 4. *There is no Turing reduction f that satisfies TASK A.*

We next consider the uniform variant of Question (b). We posit the following task for two functions f_1 and f_2 mapping sequences to sequences:

TASK B: for every $x \in \{0,1\}^\infty$ with $\dim(x) > 0$, 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 5. *There are no Turing reductions f_1 and f_2 satisfying TASK B.*

The non-uniform variants of Questions (a) and (b) remain open. In the particular case when f is a wtt-reduction, we present impossibility results analogous to those in Theorem 4 and Theorem 5.

Theorem 6. *For all rational $\sigma \in (0,1)$, there exists $\dim(x) = \sigma$ such that for every wtt-reduction f, at least one of the following statements (a), (b), (c) holds true: (a) $f(x)$ does not exist, (b) $f(x)$ is not finitary-independent with x, (c) $f(x)$ is not C-superlogarithmic.*

Theorem 7. *For all rational $\sigma \in (0,1)$, there exists x with $\dim(x) = \sigma$ such that for every wtt-reductions f_1 and f_2, at least one of the following statements (a), (b), (c) 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.*

4.2 If We Have 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 (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 8. *(a)* [Ste07] *There are two independent sequences x and y and a Turing reduction g such that x and $g(y)$ are not independent.*

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

Proposition 9. *If x and y are independent, and g is a Turing reduction, then x and $g(y)$ are finitary-independent (provided $g(y)$ exists).*

Corollary 1. *There are sequences that are finitary-independent but not independent.*

By Proposition 8, we see that (finitary-) independence is not preserved by computable functions. However, we note that there exists a simple procedures that, starting with a finitary-independent pair (x, y), produces a new pair of finitary-independent sequences. Namely, the pair (x, y_{odd}) is finitary-independent. Another question is whether given a pair of (finitary-)independent strings (x, y), it is possible to effectively produce another sequence that is (finitary-)independent with both x and y. The answer is positive in the case when x and y are both random. Indeed, if x and y are random and independent (respectively finitary-independent), then x XOR y is independent (respectively, finitary-independent) with both x and y. The similar question for non-random x and y remains open. (See Section 4.3 for some results for finite strings).

4.3 Producing Independence: The Finite Case

In what follows we attack the question on whether it is possible to effectively produce an object which is independent to each of several given independent objects for the simpler case of strings. In this setting we are able to give a positive answer for the situation when we start with three[1] input strings that are independent (and not necessarily random). First we define the analogue of independence for strings.

Definition 5. *Let $c \in \mathbb{R}^+$ and $k \in \mathbb{N}$. We say that strings $x_1, x_2, \ldots, x_k$ in $\{0,1\}^*$ are c-independent if*

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

The main result of this section is the following theorem, whose proof draws from the techniques of [Zim08].

Theorem 8. *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 $C(f(x, y, z)) \geq \lfloor \sigma_1 n \rfloor - O(\log n)$.

[1] The case when the input consists of two independent strings remains open.

Acknowledgments

We are grateful to André Nies and Frank Stephan for their insightful comments. In particular, Definition 1 has emerged after several discussions with André, and Proposition 6 and Proposition 8, (a) are due to Frank [Ste07]. We also thank Jan Reimann for his assistance in modifying a result from [NR06], which was needed for Theorem 6 and Theorem 7. We thank Alexander Shen for suggesting Theorem 3 and Proposition 8, (b).

References

[BDS07] Bienvenu, L., Doty, D., Stephan, F.: Constructive dimension and weak truth-table degrees. In: Cooper, S.B., Löwe, B., Sorbi, A. (eds.) CiE 2007. LNCS, vol. 4497, Springer, Heidelberg (to appear, 2007) Available as Technical Report arXiv:cs/0701089 ar arxiv.org

[C02] Calude, C.S.: Information and Randomness: An Algorithmic Perspective, Revised and Extended, 2nd edn. Springer, Berlin (2002)

[CST06] Calude, C., Staiger, L., Terwijn, S.: On partial randomness. Annals of Pure and Applied Logic 138, 20–30 (2006)

[CZ07] Calude, C.S., Zimand, M.: Algorithmically Independent Sequences. CDMTCS Research Report 317, 25 (2008)

[Cha75] Chaitin, G.: A theory of program size formally identical to information theory. Journal of the ACM 22, 329–340 (1975)

[Cha82] Chaitin, G.: Gödel's theorem and information. International Journal of Theoretical Physics 21, 941–954 (1982)

[DH] Downey, R., Hirschfeldt, D.: Algorithmic Randomness and Complexity. Springer, Heidelberg (to be published)

[DHNT06] Downey, R., Hirschfeldt, D., Nies, A., Terwijn, S.: Calibrating randomness. The Bulletin of Symbolic Logic 12(3), 411–492 (2006)

[GV04] Grünwald, P., Vitanyi, P.: Shannon information and Kolmogorov complexity, 2004. CORR Technical report arxiv:cs.IT/0410002, revised (May 2006)

[Kau03] Kautz, S.M.: Independence properties of algorithmically random sequences, CORR Technical Report arXiv:cs/0301013 (2003)

[Lev84] Levin, L.: Randomness conservation inequalities: information and independence in mathematical theories. Information and Control 61(1) (1984)

[Lut03] Lutz, J.: The dimensions of individual strings and sequences. Information and Control 187, 49–79 (2003)

[May02] Mayordomo, E.: A Kolmogorov complexity characterization of constructive Hausdorff dimension. Information Processing Letters 84, 1–3 (2002)

[ML66] Martin-Löf, P.: The definition of random sequences. Information and Control 9, 602–619 (1966)

[Nie05] Nies, A.: Lowness properties and randomness. Advances in Mathematics 197, 274–305 (2005)

[NR06] Nies, A., Reimann, J.: A lower cone in the wtt degrees of non-integral effective dimension. In: Proceedings of IMS workshop on Computational Prospects of Infinity, Singapore (to appear, 2006)

[Rei04] Reimann, J.: Computability and fractal dimension, Technical report, Universität Heidelberg, Ph.D. thesis (2004)

[Rya84] Ryabko, B.: Coding of combinatorial sources and Hausdorff dimension. Doklady Akademii Nauk SSR 277, 1066–1070 (1984)

[Sol75] Solovay, R.: Draft of a paper (or series of papers) on Chaitin's work, unpublished manuscript, IBM Thomas J. Watson Reserach Center, p. 215 (1975)

[Sta93] Staiger, L.: Kolmogorov complexity and Hausdorff dimension. Inform. and Comput. 103, 159–194 (1993)

[Sta05] Staiger, L.: Constructive dimension equals Kolmogorov complexity. Information Processing Letters 93, 149–153 (2005)

[Ste07] Stephan, F.: Email communication (May 2007)

[Tad02] Tadaki, K.: A generalization of Chaitin's halting probability Ω and halting self-similar sets. Hokkaido Math. J. 31, 219–253 (2002)

[vL90] van Lambalgen, M.: The axiomatization of randomness. The Journal of Symbolic Logic 55, 1143–1167 (1990)

[Zam90] Zambella, D.: On sequences with simple initial segments, ILLC Technical Report ML 1990-05, University of Amsterdam (1990)

[Zim08] Zimand, M.: Two sources are better than one for increasing the Kolmogorov complexity of infinite sequences. Proceedings of CSR 2008, Moscow (June 2008) (Also available as CORR Techical Report. arXiv:0705.4658)

[ZL70] Zvonkin, A., Levin, L.: 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), 83–124 (1970)