Notes on sum-tests and independence tests

Bruno Bauwens * Sebastiaan A. Terwijn†

Abstract

We study statistical sum-tests and independence tests, in particular for computably enumerable semimeasures on a discrete domain. Among other things, we prove that for universal semimeasures every Σ^0_1-sum-test is bounded, but unbounded Π^0_1-sum-tests exist, and we study to what extent the latter can be universal. For universal semimeasures, in the unary case of sum-test we leave open whether universal Π^0_1-sum-tests exist, whereas in the binary case of independence tests we prove that they do not exist.

KEYWORDS: sum-tests – independence tests – Kolmogorov complexity

1 Introduction

At the intersection of statistics and computability theory one is interested in the most significant statistical tests satisfying certain computational restrictions. In this paper we investigate "identity testing" and tests for independence of two strings. In the traditional statistical framework one uses concrete and simple formula-based statistical tests for elementary probability distributions such as the Kolmogorov-Smirnov test and the correlation test for Gaussian distributions. In the course of time more and more powerful tests relative to increasingly sophisticated distributions have been constructed [12, 14]. It makes sense to ask for which computational restrictions most significant tests exist.

Suppose that one wants to test a coin for fairness. A fair coin generates sequences of coin flips according to a uniform distribution. We want

*Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 913, B-9052, Ghent, Belgium, Bruno.Bauwens@ugent.be. Supported by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

†Radboud University Nijmegen, Department of Mathematics, PO Box 9010, 6500 GL Nijmegen, the Netherlands, terwijn@logic.at. Supported by the Austrian Science Fund FWF under project P20346-N18.

to test whether a generated sequence is consistent with this distribution and does not carry more structure. This is known as "identity testing" or "randomness testing". For example, we can test whether the mean of the coin flip sequence is distributed according to a Bernoulli distribution. If the coin passes this particular test, there is still the possibility that it is tricked, but we can then go on and devise other tests. It is natural to ask whether this process of improving tests has a limit. This corresponds to the question whether there exist universal elements in a set of tests of a given complexity.

Independence testing is the process of determining whether two sources can be considered as two distinctly operating systems, or that they are part of an interacting system in which information is shared or exchanged. Such independence tests show up in many engineering applications such as source separating, dimension reduction, and noise elimination [7, 8]. In advanced practical tests [6, 13] we see an evolution of tests for more complex interactions relative to more sophisticated sources.

Identity testing has been studied for ergodic sources using universal codes in Ryabko et al. [14]. These universal codes are are optimal for compressing ergodic sources and are still sufficiently computable for use in practice. The information distance and information metric introduced in [1, 10] express how similar two objects are. Complementary to independence tests, similar objects have low distance or metric value. The information metric is neither computably enumerable (c.e.) nor co-c.e. However, its computable approximations have turned out to be very useful [2, 3].

Sum-tests have been investigated as tests for randomness for finite binary strings relative to computable distributions, cf. Li and Vitányi [11]. It is shown in [11] that there are c.e. sum-tests subsuming all computable sum-tests (cf. Section 4 below). By considering sum-tests relative the product of two universal distributions the definition of sum-tests naturally leads to independence tests. This was first noted by Levin [9], and a more general notion was mentioned in Gács [5]. In [9] it is argued that algorithmic mutual information appears naturally as an independence test relative to two universal distributions.

We now give the formal definitions of sum-tests and independence tests. Some measure-theoretic terminology is explained along with our notation at the end of the section.

Let P be a given semimeasure on the set ω of natural numbers. We call a unary function $d : \omega \to \mathbb{Z}$ with

$$\sum_{x \in \omega} P(x) 2^{d(x)} \leqslant 1 \tag{1}$$

a *sum-test* for P or simply a P-sum-test. [1]

One can think of a sum-test as a test for randomness for the case of a semimeasure on a discrete domain. Namely, if d is a P-sum-test, then for every n it easily follows from (1) that the set $\{x : d(x) \geqslant n\}$ has weight $\leqslant 2^{-n}$ under the semimeasure P. Therefore strings x for which $d(x)$ is large are not random with respect to P.

Note that it is not really essential that sum-tests are integer functions: If we would allow them to have rational values, then since $2^{d(x)} \leqslant 2^{\lfloor d(x) \rfloor + 1} \leqslant 2^{d(x)+1}$ we see that by rounding off d upwards we would only change the sum (1) by a factor 2, not changing anything essential for the theory.

Definition 1.1. Given two semimeasures P and Q, a binary function $d : \omega \times \omega \to \mathbb{Z}$ with

$$\sum_{x,y \in \omega} P(x)Q(y)2^{d(x,y)} \leqslant 1 \tag{2}$$

is called an *independence test* for P and Q.

Independence tests of this form were first studied in the PhD-research of the first author. Just as sum-tests are tests for randomness, independence tests can be thought of testing possible algorithmic dependencies between pairs of strings that are random relative to P and Q. Note that analogously to the unary case we have that if d is an independence test for P and Q then for every n it follows from (2) that the set $\{(x, y) : d(x, y) \geqslant n\}$ has weight $\leqslant 2^{-n}$ under the product semimeasure $P \cdot Q$. Therefore pairs (x, y) that are random relative to P and Q for which $d(x, y)$ is large are not independent with respect to P and Q.

Below we investigate to what extent there are universal (i.e. additively dominating all others) sum-tests and independence tests for a given Σ^0_1-semimeasure P. Our results are as follows. Let m denote Levin's universal Σ^0_1-semimeasure (cf. Theorem 3.2). First, there are no unbounded Σ^0_1-sum-tests for m (Corollary 4.2), but there are unbounded and monotone Π^0_1-sum-tests for any given Σ^0_1-semimeasure (Proposition 5.1). We prove that in the following cases there is no universal Π^0_1-sum-test for $P \in \Sigma^0_1$:

- P computable (Proposition 6.1)
- $P(x) = 0$ infinitely often (Proposition 6.2)

[1] In [11] a sum-test is a function $d : \omega \to \omega$ rather than a function into the integers. The stricter definition is only interesting for the study of proper semimeasures P, that is with $\sum_x P(x) < 1$. By suggestion of the referee we use the more liberal definition. For the questions studied in this paper the difference is immaterial, and the presentation of section 7 becomes much smoother with this definition.

- P does not have a strictly positive computable lower bound, i.e. a computable Q such that $P(x) \geqslant Q(x) > 0$ a.e. x. (Corollary 6.3)

Note that no universal Σ^0_1-semimeasure satisfies any of these. The most important question we leave open is whether for $P = m$ there is no universal Π^0_1-sum-test (Question 6.4). In Section 7 we answer this question in the binary case of independence tests: We prove that there is no universal Π^0_1-independence test in case both measures are m (Theorem 7.3).

We end this section with some notation and terminology. As we already said, ω is the set of natural numbers. This set is effectively bijective with the set of all finite binary strings.

A function f is Σ^0_1, or *computably enumerable*, if it is computably approximable from below, that is, if there exists a computable function $\hat{f}(x, s)$ that is monotonic nondecreasing in s such that $\lim_s \hat{f}(x, s) = f(x)$. Similarly, f is Π^0_1 if it is computably approximable from above, i.e. the approximation $\hat{f}$ is monotonic nonincreasing in s.

A function $P : \omega \to \mathbb{R}$ is a *probability measure* if $\sum_x P(x) = 1$. Since every Σ^0_1-measure is computable (Proposition 3.1), in computability theory it is often natural to consider semimeasures. A function $P : \omega \to \mathbb{R}$ is a *semimeasure* if $\sum_x P(x) \leqslant 1$.

A function f *dominates* a function g if $f(x) \geqslant g(x)$ for almost every x, and f *additively dominates* g there is a constant c such that $f(x) + c \geqslant g(x)$ for every x. As in [11], we call a function f *universal*[2] or *additively optimal* for a class $\mathcal{C}$ if $f \in \mathcal{C}$ and f additively dominates all other functions in $\mathcal{C}$. A function is called an *order* if it is monotone and unbounded.[3] Given two functions d and d', the phrase "$d' - d$ is unbounded" abbreviates the statement that for all i there is x such that $d'(x) - d(x) \geqslant i$.

2 Some general notes on Σ^0_1- and Π^0_1-functions

As a preparation for sections to follow, we list some basic folk facts about Σ^0_1- and Π^0_1-functions. (The discussion here is about functions from ω to ω.)

(i) There is no universal Σ^0_1-function. Namely if $f \in \Sigma^0_1$ then also the function $\lambda x.f(x) + x$ is Σ^0_1.

[2]Note that the term universal is used here to refer to growth rates, and should not be confused with the other common usage of the term, referring to the ability to enumerate all other functions in the class.

[3]This translation of Schnorr's term "Ordnungsfunktion" [15] has meanwhile become standard in randomness theory.

(ii) The reason we cannot build a universal (additively optimal) Σ^0_1-function is that the Σ^0_1-functions are not uniformly enumerable; in an effective enumeration of the computable approximations (which does exist) we cannot effectively separate those that remain finite from the ones that grow unbounded. That there is a universal Martin-Löf test (Martin-Löf) and that there is a universal Σ^0_1-semimeasure (Levin, Theorem 3.2) holds because these Σ^0_1-objects satisfy an extra boundedness condition that we can check along the way to see if it is violated, and if so render the object harmless by discarding it after finitely many steps.

(iii) The Π^0_1-functions are also not uniformly enumerable, but for a different reason: Every Π^0_1-function is computably bounded (namely by any of its computable approximations). If there were a universal Π^0_1-function, its computable bound would in particular dominate all computable functions, which is impossible.

(iv) Not every Σ^0_1-function is computably bounded: Take an effective enumeration of all partial computable functions φ_e and define

$$f(x) = \sum \big\{ \varphi_i(i) : i \leqslant x \wedge \varphi_i(i)\downarrow \big\}.$$

This f is a Σ^0_1-order dominating any computable function.

(v) Given any order f we can define a slow growing inverse h of f by

$$h(x) = \mu n.\ f(n) \geqslant x.$$

If $f \in \Sigma^0_1$ then $h \in \Pi^0_1$, so if we take for f the fast growing function from the previous item then we obtain an Π^0_1-order dominated by any computable order.

(vi) Conversely, given a fast growing Σ^0_1-order f we can define a slow growing Π^0_1-order h by

$$f(x) = \mu n.\ h(n) \geqslant x.$$

Hence, since there are no no universally fast growing Σ^0_1-orders, we see that there are no universally slow growing Π^0_1-orders.

(vii) Any Σ^0_1-order dominates a computable order: Given a Σ^0_1-order one easily constructs a slower growing computable order. This is also true for nonmonotonic functions: For any unbounded Σ^0_1-function f one can find an unbounded computable g such that the function $f - g$ is positive and unbounded.

In conclusion: Σ^0_1-orders can grow faster but not slower than any computable one, whereas Π^0_1-orders can grow slower but not faster than any computable one.

3 General notes on measures and semimeasures

For the record we state the following

Proposition 3.1. *1. Every Σ_1^0-measure is computable,*

2. There is a Π_1^0-measure that is not computable.

Proof. 1. This well-known and easy to see: If $P \in \Sigma_1^0$ with computable approximation P_s and $\sum_x P(x) = 1$ then to approximate $P(x)$ to within ε, find a stage s such that $1 - \sum_x P_s(x) < \varepsilon$. Then $P(x) - P_s(x) < \varepsilon$.

2. Let X be any noncomputable Π_1^0-set, with computable approximation X_s. Define a measure P as follows: At stage s assign the s values $2^{-1}, \ldots 2^{-s}$ to the first s elements of $X_s \subseteq X_{s-1}$, in such a way that the elements of X_s that were already assigned a value at a previous stage retain this, and the values that were assigned to elements in $X_{s-1} - X_s$ are given a new host element. For any element $x \notin X$ we define $P(x) = 0$. Then $P \in \Pi_1^0$, and P is not computable because otherwise, since $x \in X \Leftrightarrow P(x) > 0$, X would also be computable. Note that in general $P(x) > 0$ is not decidable for computable P, but in this case it is: x is assigned an initial value 2^{-i} with $i \leqslant x$. Computing $P(x)$ to within precision 2^{-i-1} decides whether it is 2^{-i} or 0.

□

A semimeasure P *(multiplicatively) dominates* a semimeasure Q if there is a rational constant $q > 0$ such that $P(x) > qQ(x)$. A semimeasure P is *(multiplicatively) universal* for a class of semimeasures $\mathcal{C}$ if $P \in \mathcal{C}$ and P dominates every $Q \in \mathcal{C}$. As quoted above, Levin showed that there is a universal Σ_1^0-semimeasure. Not surprisingly, there is no Π_1^0 one.

Theorem 3.2. (Levin) *There exists a universal Σ_1^0-semimeasure m.*

Proof. We sketch the proof for later reference. Let P_i be an effective enumeration of all Σ_1^0-semimeasures. Note that such an enumeration can be obtained because we can see in finitely many steps whether the condition $\sum_x P_i(x) \leqslant 1$ is violated. Define

$$m(x) = \sum_i 2^{-i} P_i(x).$$

Clearly $m(x)$ is finite, $m \in \Sigma_1^0$, and m is multiplicatively universal. □

The following easy facts are also well-known in the folklore of the field:

Proposition 3.3. (i) *There is no universal computable semimeasure.*

(ii) *There is no universal* Π^0_1*-semimeasure.*

Proof. Both item (i) and (ii) follow from the following. Let P be a Π^0_1-semimeasure. We construct a computable semimeasure Q such that

$$\forall q \in \mathbb{Q}^{>0} \exists x \;\; P(x) < qQ(x). \tag{3}$$

Given q we simply search for an x where $P(x)$ is small and set a large value for $Q(x)$. Note that x can be found effectively since $P \in \Pi^0_1$. More precisely, given $q = 2^{-i}$ find a fresh x such that $P(x) < 2^{-2i}$. Set $Q(x) = 2^{-i}$, and to make Q total set $Q(y) = 0$ for all $y < x$ that were not yet defined. The Q thus constructed is computable, clearly satisfies (3), and $\sum_x Q(x) = \sum_i 2^{-i} = 1$. □

Corollary 3.4. *Let m be the universal Σ^0_1-semimeasure and let P be a Π^0_1-semimeasure. Then the function $m(x)/P(x)$ is unbounded. In particular, $m(x) > P(x)$ infinitely often.*

Proof. Suppose for a contradiction that $c \in \omega$ is a constant such that $m(x)/P(x) \leqslant c$ for every x. By Proposition 3.3, let Q be a computable measure such that (3) holds. Then a fortiori

$$\forall q \in \mathbb{Q}^{>0} \exists x \;\; m(x) < q \cdot c \cdot Q(x),$$

contradicting that m is multiplicatively universal. □

Call a semimeasure P *monotone* if $x \leqslant y$ implies $P(x) \geqslant P(y)$. We note that there does not exist a monotone universal Σ^0_1-semimeasure. This is not difficult to prove directly, but it also follows from the Coding Theorem (10) below. Namely, if m is universal then $-\log m(x) = K(x)$ up to a fixed additive constant, hence if m were monotone then K would also be monotone, which is of course not the case. There is a Σ^0_1-semimeasure that is multiplicatively universal among the monotonic ones, namely $m'(x) = \min_{y \leqslant x} m(y)$, which is within a multiplicative constant equal to

$$\frac{1}{xm(\log x)}.$$

4 Σ^0_1-sum-tests

In Li and Vitányi [11, Theorem 4.3.5] it is proven that for every strictly positive computable measure P the Σ^0_1-function

$$\log\big(m(x)/P(x)\big)$$

is a Σ^0_1-universal sum-test for P. In particular, since by Corollary 3.4 the function $m(x)/P(x)$ is unbounded, there is an unbounded P-sum-test. We prove here that for $P = m$ this is no longer true.

Proposition 4.1. *For any unbounded Σ^0_1-function $d : \omega \to \mathbb{Z}$ there is a computable measure P such that*

$$\sum_{x\in\omega} P(x)2^{d(x)} = \infty. \tag{4}$$

Proof. Suppose that $d : \omega \to \mathbb{Z}$ is Σ^0_1 and unbounded. We construct a computable measure P such that

$$\sum_{x\in\omega} P(x) = 1. \tag{5}$$

and (4) holds. The construction is in ω stages. At stage s, search for a fresh (i.e. hitherto not used in the construction) element x such that $d(x) \geqslant s$. Such x can be found effectively since d is unbounded and Σ^0_1. For this x define $P(x) = 2^{-s}$. To make sure that P is total, define $P(y) = 0$ for all $y < x$ for which $P(y)$ was not yet defined at a previous stage. End of construction.

Clearly the P thus constructed satisfies (4) and (5), since at stage s of the construction we contribute an amount of 2^{-s} to $\sum_x P(x)$ and an amount of at least 1 to $\sum_x P(x)2^{d(x)}$. □

Corollary 4.2. *Every Σ^0_1-sum-test for the universal Σ^0_1-semimeasure m is bounded.*

Proof. Suppose that d is unbounded. Let P be as in Proposition 4.1. Since m is universal, there is $q > 0$ with $m(x) \geqslant qP(x)$ for all x. Then $\sum_x m(x)2^{d(x)} \geqslant \sum_x qP(x)2^{d(x)} = \infty$, hence d is not a sum-test for m. □

We remark that for every computable semimeasure P there is a computable order d that is a sum-test for P, as is easily seen. (One can use for example the proof of Proposition 6.1 below, taking d constant.)

For later purposes we note the following variant of Proposition 4.1:

Proposition 4.3. *If d and d' are computable functions such that the function $d' - \max(0, d)$ is unbounded, then there is a computable semimeasure P such that*

$$\sum_{x\in\omega} P(x)2^{d'(x)} = \infty. \tag{6}$$

and

$$\sum_{x\in\omega} P(x)2^{d(x)} \leqslant 1. \tag{7}$$

That is, d is a sum-test for P and d' is not.

Proof. The proof is similar to that of Proposition 4.1, except that at stage s we now search for a fresh number x such that

$$d'(x) - \max(0, d(x)) \geqslant s.$$

For this x define $P(x) = 2^{-\max(0,d(x))-s}$. Again, to make P total, define $P(y) = 0$ for all $y < x$ for which $P(y)$ was not yet defined at a previous stage. Note that P is indeed a semimeasure.

Now P satisfies (6) and (7), since at stage s of the construction we contribute an amount of $2^{-\max(0,d(x))-s}2^{d(x)} \leqslant 2^{-s}$ to $\sum_x P(x)2^{d(x)}$ and an amount of $P(x)2^{d'(x)} \geqslant 2^{-\max(0,d(x))-s}2^{\max(0,d(x))+s} = 1$ to $\sum_x P(x)2^{d'(x)}$. □

Finally, we claim that there is a semimeasure $P \in \Sigma^0_1$ without Σ^0_1-universal sum-test. This is trivial to see if we allow $P(x) = 0$ for infinitely many x, but it also holds for strictly positive P:

Proposition 4.4. *There exists a strictly positive Σ^0_1-semimeasure P such that there is no Σ^0_1-universal sum-test for P.*

Proof. Since the constant zero function is a sum-test for any semimeasure, a universal sum-test is bounded from below by some constant $k \in \mathbb{Z}$. So in proving that such a universal sum-test does not exist we may restrict ourselves to such functions.

Let d_i be an effective enumeration of all Σ^0_1-functions from ω to $\mathbb{Z} \cup \{\infty\}$ that are bounded from below by some (possibly negative) constant. (The latter assumption is needed to have an effectively enumerable class of functions; for the rest of the proof it is not needed.) Let $d_{i,s}$ denote the approximation of d_i. We construct a semimeasure $P \in \Sigma^0_1$ and functions $d'_i \in \Sigma^0_1$ so that for every i it holds that $d'_i - d_i$ is unbounded and

$$\sum_x P(x)2^{d_i(x)} \leqslant 1 \Longrightarrow \sum_x P(x)2^{d'_i(x)} \leqslant 1. \tag{8}$$

Let $\langle x, y\rangle$ be a bijective pairing function from ω^2 to ω. We assign an infinite computable domain R_i to the strategy for d_i as follows. Define

$$R_i = \big\{\langle x, i\rangle : x \in \omega\big\}$$

and

$$d'_{i,s}(x) = \begin{cases} d_{i,s}(x) + x & \text{if } x \in R_i \\ 0 & \text{otherwise.} \end{cases}$$

We construct P by defining its approximation P_s as follows. Let $P_0(x) = 2^{-2x-1}$, so that P is strictly positive. At stage s of the construction, for every $i \leqslant s$, if s is the first stage such that

$$\sum_{x<s} P_s(x)2^{d'_{i,s}(x)} > 1 \tag{9}$$

then define

$$P_{s+1}(x) = P_s(x)2^{d'_{i,s}(x)-d_{i,s}(x)} = P_s(x)2^x$$

for every $x \in R_i$. Note that since this can happen only once, we have that $P_s(x)$ equals either $P_0(x)$ or $P_0(x)2^x$. This ends the construction.

We check that requirements (8) are satisfied for every i. Suppose that $\sum_x P(x)2^{d'_i(x)} > 1$. Then (9) holds for some s, hence

$$\begin{aligned}
\sum_{x\in\omega} P(x)2^{d_i(x)} &\geqslant \sum_{x\notin R_i} P_s(x)2^{d_{i,s}(x)} + \sum_{x\in R_i} P_{s+1}(x)2^{d_{i,s}(x)} \\
&\geqslant \sum_{x\notin R_i} P_s(x) + \sum_{x\in R_i} P_s(x)2^{d'_{i,s}(x)-d_{i,s}(x)}2^{d_{i,s}(x)} \\
&\geqslant \sum_{x\notin R_i} P_s(x) + \sum_{x\in R_i} P_s(x)2^{d'_{i,s}(x)} \\
&= \sum_{x\in\omega} P_s(x)2^{d'_{i,s}(x)} \;>\; 1.
\end{aligned}$$

hence (8) is satisfied. Clearly $P \in \Sigma^0_1$, so it only remains to show that P is a semimeasure. Since the domains R_i partition ω we have

$$\begin{aligned}
\sum_{x\in\omega} P(x) &= \sum_i \sum_{x\in R_i} P(x) \\
&\leqslant \sum_i \sum_{x\in R_i} P_0(x)2^x \\
&= \sum_i \sum_{x\in R_i} 2^{-x-1} \\
&= \sum_{x\in\omega} 2^{-x-1} \;=\; 1.
\end{aligned}$$

□

5 Unbounded Π^0_1-sum-tests

We saw in Section 4 that there are Σ^0_1-semimeasures with no nontrivial sum-tests: all Σ^0_1-sum-tests for m are bounded. We now prove that for Π^0_1 there are nontrivial, unbounded, examples.

Proposition 5.1. *For every Σ^0_1-semimeasure P there is a Π^0_1-order d that is a sum-test for P.*

Proof. The idea is to monitor the tails of the sum $\sum_x P(x)$, and estimate at every stage the first element x_i such that $\sum_{y\geqslant x_i} P(y) \leqslant 2^{-i}$. The x_i may grow, but eventually come to a finite limit. If we know them we can add suitable large factors $2^{d(x)}$ that satisfy $\sum_x P(x)2^{d(x)} \leqslant 1$. If x_i

turned out to be wrong, we simply decrease $d(x)$, but we have to do this only finitely often. Formally the construction proceeds as follows.

Start with $x_{i,0} = i$. At stage s, when

$$\sum_{y \geqslant x_{i,s}} P_s(y) \leqslant 2^{-i}$$

let $x_{i,s+1} = x_{i,s}$, otherwise set $x_{j,s+1} = x_{j,s} + 1$ for all $j \geqslant i$. For all $x \in [x_{i,s}, x_{i+1,s})$ define

$$d_s(x) = \lfloor \log i \rfloor.$$

End of construction.

First note that $\lim_s x_{i,s} = x_i$ exists for every i since $\sum_x P(x)$ converges. Since $x_{i,s}$ is nondecreasing, $d_s(x)$ can only decrease, and since the limit exists it can do so only finitely many times.[4] Hence $d \in \Pi^0_1$, and it is unbounded since $d(x_i) = \lfloor \log i \rfloor$. Finally,

$$\begin{aligned}
\sum_{x \in \omega} P(x) 2^{d(x)} &\leqslant \sum_{i \in \omega} \sum_{x \in [x_i, x_{i+1})} P(x) 2^{\log i} \\
&\leqslant \sum_{i \in \omega} i \sum_{x \geqslant x_i} P(x) \\
&\leqslant \sum_{i \in \omega} 2^{-i} i = 2.
\end{aligned}$$

Therefore, $d(x) - 1$ defines a sumtest for P. □

We can improve Proposition 5.1 as follows:

Proposition 5.2. *For every Σ^0_1-semimeasure P and every computable sum-test d for P, there is a Π^0_1-sum-test d' for P such that $d' - d$ is unbounded. If d is an order then d' can be chosen to be an order as well.*

Proof. The proof is similar to that of Proposition 5.1. The only difference is that we now monitor the tails of the sum $\sum_x P(x)2^{d(x)}$, and estimate at every stage the first element x_i such that $\sum_{y \geqslant x_i} P(y)2^{d(y)} \leqslant 2^{-i}$. If this holds at stage s, we let

$$d'_s(x) = d_s(x) + \lfloor \log i \rfloor$$

for all $x \in [x_{i,s}, x_{i+1,s})$. That $\lim_s x_{i,s}$ exists follows because d is computable, so the values $P_s(x)2^{d(x)}$ can only go up. If d is an order then d' is also an order. □

[4] Note that since $d_0(x) = \log x$, $d_s(x)$ can change at most $\log x$ times, but the number of times $x_{i,s}$ changes is not computably bounded. Hence the limit function d can in general be very slow growing, that is, be dominated by any computable order.

We now turn to the rate of growth of sum tests. If d is any (not necessarily Π^0_1) m-sum-test then d does not grow very fast:

Proposition 5.3. *If d is any m-sum-test then d is dominated by all Π^0_1-functions f with*

$$\sum_{x\in\omega} 2^{-f(x)} < \infty.$$

This also holds on any computable subset $R \subseteq \omega$: $d(x) \leqslant f(x)$ for almost every $x \in R$ whenever $\sum_{x\in R} 2^{-f(x)} < \infty$.

Proof. We prove only the first part, since the second is just an easy modification. Given f as above, suppose that f does not dominate d, so that $d(x) > f(x)$ infinitely often. We produce a semimeasure $P \in \Sigma^0_1$ such that d is not a sum-test for P. (Hence by universality of m the same holds with m in place of P.) Simply put $P(x) = 2^{-f(x)}$ for every x. Then $\sum_x P(x) < \infty$, so a suitable tail of P is a semimeasure. Without loss of generality we may assume that P itself is a semimeasure. Since $f \in \Pi^0_1$ we have $P \in \Sigma^0_1$. Finally,

$$\sum_{x\in\omega} P(x)2^{d(x)} \geqslant \sum_{d(x)\geqslant f(x)} P(x)2^{d(x)} \geqslant \sum_{d(x)\geqslant f(x)} 2^{-f(x)}2^{f(x)} = \infty,$$

hence d is not a P-sum-test. □

Corollary 5.4. *If d is a Π^0_1-sum-test for m then $\sum_x 2^{-d(x)} = \infty$.*

Proof. If we would have $\sum_x 2^{-d(x)} < \infty$ then also $\sum_x 2^{-(d(x)-1)} < \infty$, hence by Proposition 5.3 the Π^0_1-function $d(x) - 1$ would dominate d, contradiction. □

Next we turn to the question when a sum-test can be replaced by an order dominating it.

Proposition 5.5. *There exist a computable measure P and a computable P-sum-test d such that every (not necessarily effective) order d' dominating d is not a P-sum-test.*

Proof. To construct P and d, simply let $d(x)$ be large when $P(x)$ is small and vice versa: For every x define

$$\begin{aligned} P(2x) &= 0 & d(2x) &= x \\ P(2x+1) &= 2^{-x-1} & d(2x+1) &= 0 \end{aligned}$$

Clearly P is a measure and d is a P-sum-test. If d' is an order dominating d then $d'(2x+1) \geqslant d'(2x) \geqslant d(2x) = x$, hence $\sum_x P(x)2^{d'(x)} \geqslant \sum_x 2^{-x-1}2^x = \infty$. □

Proposition 5.5 also holds if we require that P be strictly positive, with the same proof idea. At this point we ask what happens when $P = m$ and $d \in \Pi^0_1$:

Question 5.6. *Suppose that d is a Π^0_1-sum-test for m. Is there always a Π^0_1-order d' dominating d that is a sum-test for m ?*

6 Universal Π^0_1-sum-tests

We have seen that for the universal Σ^0_1-semimeasure m there are only trivial Σ^0_1-sum-tests, namely the bounded ones, and that there are nontrivial Π^0_1-sum-tests for m. In this section we investigate if Σ^0_1-semimeasures can have a universal Π^0_1-sum-test. We do not obtain a complete answer to this question, but only prove that no universal Π^0_1-sum-test exists in specific cases. In particular we leave open the case of universal Σ^0_1-semimeasures.

Proposition 6.1. *Suppose that P is a computable semimeasure. Then there is no universal Π^0_1-sum-test for P.*

Proof. The idea is similar to that of Proposition 3.3. Given $d \in \Pi^0_1$ such that $\sum_x P(x)2^{d(x)} \leqslant 1$, construct $d' \in \Pi^0_1$ such that for all i there is x such that $d'(x) \geqslant d(x) + i$. Given i, effectively search for x such that $P(x)2^{d(x)} < 2^{-2i}$ (which is possible since such x exist and $d \in \Pi^0_1$), so that $P(x)2^{d(x)+i} < 2^{-i}$. For this x define $d'(x) = d(x)+i$, and set $d'(y) = d(y)$ for all $y < x$ for which $d'(y)$ was not yet defined. Then

$$\sum_{x\in\omega} P(x)2^{d'(x)} \leqslant \sum_{d'(x)=d(x)} P(x)2^{d(x)} + \sum_{i\in\omega} 2^{-i} < \infty,$$

hence $d'-c$, for some c large enough, is a Π^0_1-sum-test for P not dominated by d. □

Note that the proof of Proposition 6.1 in fact works for every Π^0_1-semimeasure P.

Proposition 6.2. *If a Σ^0_1-semimeasure P has a coinfinite support, i.e. if $P(x) = 0$ for infinitely many x, then there is no universal Π^0_1-sum-test for P.*

Proof. Given a Π^0_1-sum-test d and a computable order f, define the function

$$d'_t(x) = \begin{cases} d_t(x) + f(x) & \text{if } P_t(x) = 0 \\ d_t(x) & \text{otherwise.} \end{cases}$$

Remark that $d' = \lim d'_t$ is again a Π^0_1-sum-test for P. If P has a coinfinite support then $d'(x) - d(x)$ is unbounded, hence d is not Π^0_1-universal. □

Corollary 6.3. *If $P \in \Sigma_1^0$ does not have a strictly positive computable lower bound (i.e. a computable Q such that $P(x) \geqslant Q(x) > 0$ a.e. x) then there is no universal Π_1^0-sum-test for P.*

Proof. This follows from Proposition 6.2, since if $P \in \Sigma_1^0$ is a.e. strictly positive then it has such a computable lower bound. □

Question 6.4. *Let P be any Σ_1^0-semimeasure. Then there is no universal Π_1^0-sum-test for P. In particular there is no universal Π_1^0-sum-test for m.*[5]

In the remaining part of this section we make some further remarks about universal sum-tests. We first prove that there are Σ_1^0-semimeasures P for which the class of computable sum-tests has a universal element. In fact, *every* computable function is such a universal sum-test:

Proposition 6.5. *Given any computable function $d : \omega \to \omega$, the Σ_1^0-semimeasure*

$$P(x) = m(x)2^{-d(x)}$$

satisfies:

- *d is (additively) universal for the class*

$$\big\{d' \text{ computable} : d' \text{ is } P\text{-sum-test}\big\},$$

- *P is (multiplicatively) universal for the class*

$$\big\{P' \in \Sigma_1^0 : d \text{ is } P'\text{-sum-test}\big\}.$$

Proof. For the first item, suppose that d' is a sum-test for P that is not additively dominated by d, i.e. $d' - d$ is unbounded. Then $P'(x) = m(x)2^{d'(x)-d(x)}$ is a Σ_1^0-semimeasure that is not multiplicatively dominated by m, contradicting Theorem 3.2. For the second item, suppose that P' is a Σ_1^0-semimeasure for which d is a sum-test. Then $Q(x) = P'(x)2^{d(x)}$ is a Σ_1^0-semimeasure, hence by Theorem 3.2, $P(x)2^{d(x)} = m(x)$ multiplicatively dominates $Q(x)$, and hence $P(x)$ multiplicatively dominates $P'(x)$. □

Note that the proof of Proposition 6.5 does not work for Π_1^0-functions: For d constant we obtain the universal semimeasure m, but by Proposition 5.1 there are Π_1^0-functions d' dominating every constant that are still sum-tests for m, hence d is not universal. In fact, Proposition 5.2 shows that Proposition 6.5 fails for Π_1^0: There are $d \in \Pi_1^0$ that are not Π_1^0-universal

[5] Note added in proof: There is now a draft by the first author containing a concept proof solving the second part of this question for m in the affirmative.

for any $P \in \Sigma^0_1$, namely any computable d. In Proposition 6.6 we show that, given a computable d, there is even a uniform witness d' showing that d is not Π^0_1-universal.

Say that a given semimeasure P *splits* two functions d and d' if d is a P-sum-test and $\sum_x P(x)2^{d'(x)} = \infty$ (in that order). Proposition 4.3 says that every pair of computable d and d' with $d' - d$ unbounded can be split by a computable semimeasure.

Proposition 6.6. *For any computable $d : \omega \to \omega$, there is $d' \in \Pi^0_1$ such that $d' - d$ is unbounded and such that no Σ^0_1-semimeasure splits d and d'.*

Proof. Let $P(x) = m(x)2^{-d(x)}$ be as in Proposition 6.5. Let $d'(x) = d(x) + b(x)$ where b is the unbounded sum-test for m as constructed in Proposition 5.1. Suppose that Q is a Σ^0_1-semimeasure and that d is a sum-test for Q. Then P dominates Q by Proposition 6.5. If $q > 0$ is such that $qQ(x) < P(x)$ then

$$\begin{aligned} \sum_x Q(x)2^{d'(x)} &\leqslant \frac{1}{q}\sum_x P(x)2^{d'(x)} \\ &= \frac{1}{q}\sum_x m(x)2^{-d(x)}2^{d(x)+b(x)} \\ &\leqslant \frac{1}{q} < \infty \end{aligned}$$

Hence Q does not split d and d'. □

7 Independence tests

Recall the definition of independence test from Section 1. The results about sum-tests from previous sections also hold, mutatis mutandis, for the binary case of independence tests, with the same proofs except for Proposition 6.5. In particular, in the case of $P = Q = m$, Corollary 4.2 now states that there are no unbounded computable and Σ^0_1-independence tests. There exist unbounded Π^0_1 tests and we will show that there is no Π^0_1-universal test (Theorem 7.3). Note that this answers the binary analogue of Question 6.4. As a corollary to the proof it follows that for all enumerable semimeasures P, Q, a Π^0_1-independence test for (P, Q) exist, with $d(x, y) \geqslant l(x) - O(\log l(x))$ for infinitely many binary strings x, y with length $l(x) = l(y)$, and for each Π^0_1-independence test d for (m, m), there is a test d' such that $d'(x, y) - d(x, y)$ exceeds $l(x) - O(\log l(x))$ infinitely often. Since $P = Q = m$ throughout this

section, "independence test" will abbreviate "independence test for m and m".

We start with an informal argument why there is no Π_1^0-universal independence test. Consider the set

$$D = \big\{(x,y) : l(x) = l(y) \wedge x, y \text{ random and dependent}\big\}.$$

D is a natural example of a d.c.e. set, that is, a set that is the difference of two c.e. sets, in this case the set of pairs (x,y) with x and y dependent minus the set of pairs where one of x and y is not random. Now suppose that d is a Π_1^0 independence test. As pointed out in Section 1, it follows directly from (2) that the set of pairs x, y where $d(x,y)$ is large, is small in measure. Thus d provides us with an effective method for detecting dependencies in such pairs. Now suppose that for all $(x,y) \in D$, $d(x,y)$ would be large. Then we would have that x and y are dependent if and only if $d(x,y)$ is large. Since the latter is a Π_1^0-event, we obtain that $D \in \Pi_1^0$, a contradiction. This means that there are $(x,y) \in D$ such that $d(x,y)$ is small, that is, x and y are dependent but d does not see this. Since D is a set of small measure, we could construct a new d' with d' higher on such pairs (thus showing that d is not universal). To recognize such pairs, we have to recognize more dependencies than d does by allowing for more computation time. Some pairs (x,y) may fall through at a later time when it turns out that one of x and y is not random, but if we allow for enough computation time we will also find pairs in D that were not recognized by d, and hence we can show that d is not universal. The proof below is more informative, since it shows that the functions d^i of the specific form defined there form a strict hierarchy of independence tests, and that every independence test is dominated by some d^i.

In this section we use Kolmogorov complexity. For general background we refer to Li and Vitányi [11] and the forthcoming Downey and Hirschfeldt [4]. We fix our notation for this section. Let $\langle x,y\rangle$ denote a computable bijective mapping from $\omega \times \omega$ to ω. Let Φ be an optimal universal prefix-free Turing machine. $\Phi_s(p|z)\downarrow = x$ if and only if $\Phi(p|z)$ outputs x in less than s steps using an auxiliary tape for string z. The prefix-free complexity functions are $K_s(x|z) = \min\{l(p) : \Phi_s(p|z)\downarrow = x\}$, $K(x|z) = \lim_s K_s(x|z)$, $K(x) = K(x|\emptyset)$, and $K(x,y) = K(\langle x,y\rangle)$. The complexity of a partial computable function f is defined by

$$K(f) = \min\{l(p) : \forall x \in \mathrm{dom} f\ [\Phi(p|x)\downarrow = f(x)]\}.$$

The algorithmic complexity of a one-argument Σ_1^0-function or Π_1^0-function $d(x)$ is given by the lowest complexity $K(d_t(x))$ of a two-argument function $d_t(x)$ that is the computable approximation of $d(x)$ as $t \to \infty$.

$f(x) \leqslant^+ g(x)$ or $f(x) \leqslant g(x) + O(1)$ means that there exists a constant c such that for all x as indicated or allowed in the context of the proof, we have: $f(x) \leqslant g(x) + c$. $f(x) =^+ g(x)$ means $f(x) \leqslant^+ g(x)$ and $g(x) \leqslant^+ f(x)$. Similarly for the $O(\log)$ notation. Theorem 3.2 stated the existence of a universal Σ_1^0-semimeasure. The Coding Theorem [11] states that the function

$$m(x) = 2^{-K(x)} \tag{10}$$

is a multiplicatively universal Σ_1^0-semimeasure. Let $l(x)$ be the length of the number x, seen as a finite binary string, and let from now on n be short for $l(x)$.

Definition 7.1.

- $R = \{(x, y) : l(x) = l(y) \wedge K(x), K(y) \geqslant n - \log n\}$.
- A function f *R-dominates* g (notation $f \overset{\mathrm{R}}{\succcurlyeq} g$) if

$$\exists c \forall^\infty (x, y) \in R\big[f(x, y) + c \log n \geqslant g(x, y)\big].$$

- Define for each i the total functions:

$$\begin{aligned} T^i(n) &= \max\{\Phi(p|n) : l(p) \leqslant i, \lambda m.\Phi(p|m) \text{ is total}\}, \\ K^i(x, y) &= K_{T^i(l(\langle x,y\rangle))}(x, y) \\ K^i(x) &= K^i(x, \emptyset), \\ d^i(x, y) &= K(x) + K(y) - K^i(x, y). \end{aligned}$$

Note that domination implies R-domination and that R-domination defines a semi-order on the binary functions. The function $T^i(n)$ is $\emptyset''$-computable, but for *fixed* i it is computable. Hence for fixed i also $K^i(x, y)$ is computable.
There is a prefix-free code such that every $n \in \omega$ is encoded with length $2 \log n$. Let z be the binary expansion of n. Remark that $l(z) = \lceil \log n \rceil$. The code word

$$z_0 0 z_1 0 z_2 0 ... z_{\lceil \log n \rceil} 1$$

for n has length $2 \log n$. Remark that the set of these code words is prefix-free. The time needed to decode this sequence is bounded by a computable function of n. Combining a prefix-free code for n with a prefix-free code for x given n results in a prefix-free code for x. Therefore, without loss of generality it can be assumed about the universal machine Φ implicit in K that:

$$\exists c \forall i \geqslant c \forall x [K^{i+c}(x) - 2 \log n - c \leqslant K^i(x|n) \leqslant K^i(x)]. \tag{11}$$

Lemma 7.2. *For all i, d^i is a Π_1^0-independence test.*

Proof. Since K is a Π_1^0-function, d^i is Π_1^0. Clearly $d^i(x,y)$ is increasing in i and $\lim_i K^i(x,y) = K(x,y)$, therefore:

$$d^i(x,y) \leqslant K(x) + K(y) - K(x,y),$$

and

$$\sum_{x,y} m(x)m(y)2^{d^i(x,y)} \leqslant \sum_{x,y} 2^{-K(x,y)} \leqslant 1. \qquad \square$$

Theorem 7.3. *There is no universal Π_1^0-independence test.*

Proof. Because domination implies R-domination, the absence of a universal element in the set of Π_1^0 independence tests follows from the absence of a universal element with respect to R-domination: if there were a Π_1^0-independence test dominating all other Π_1^0-independence tests, it would also R-dominate any Π_1^0-independence test. We show in two steps that this is impossible:

- Lemma 7.5: For all Π_1^0-independence tests d, there is an i such that $d^i \overset{\mathrm{R}}{\succcurlyeq} d$.
- Lemma 7.9: For all i, there is a j such that $d^i \overset{\mathrm{R}}{\not\succcurlyeq} d^j$.

Suppose d were R-universal, then by Lemma 7.5 and by transitivity of R-domination, there should also be an R-universal element among the set of $d^i, i \in \omega$. However this is not possible by Lemma 7.9. $\square$

In the proof of Lemma 7.5 and 7.6 the following lemma is used.

Lemma 7.4. *For all n, let $P(x,y|n) > 0$ be a positive computable semimeasure over all binary strings x,y, with $l(x) = l(y) = n$. If for some i, there is a binary string p satisfying:*

$$\Phi_{T^i(n)}(p|x,y,n)\downarrow = \lceil -\log P(x,y|n) \rceil,$$

then

$$K^{i+O(1)}(x,y|n) \leqslant^+ l(p) - \log P(x,y|n).$$

Proof. For any computable semimeasure P, Shannon-Fano coding [11] provides a prefix-free code for all (x,y) of length n with maximal encoding length $-\log P(x,y|n) + O(1)$. To decode the Shannon-Fano code of (x,y), a fixed algorithm needs to be executed that requires an amount of computation steps bounded by $f(n, T^i(n)) \leqslant T^{i+O(1)}(n)$ for some computable function f. The encoding of (x,y) contains two parts: the encoding of P with length $l(p)$, and the corresponding Shannon-Fano code. $\square$

Lemma 7.5. *For all Π_1^0-independence tests d, there is an i such that $d \overset{R}{\preccurlyeq} d^i$.*

Proof. By universality of m there exists a constant c such that

$$-\log m(x) \leqslant n + 2\log n + c. \tag{12}$$

For any n, the values $d_s(u,v)$ can be evaluated for increasing s and all (u,v) with $l(u) = l(v) = n$ until a time $s = \tau(n)$ is found such that

$$\sum_{l(u)=l(v)=n} 2^{d_s(u,v)-2n-4\log n-2c} \leqslant 1.$$

Such s always exists because of (2), (10) and (12). Hence the "code length" function

$$cl(u,v) = -d_s(u,v) + 2n + 4\log n + 2c$$

defines a semimeasure $P(u,v|n) = 2^{-cl(u,v)}$. The function $\tau(n)$ that evaluates s for each n is computable, and by the above construction it has complexity $K(\tau) \leqslant K(d) + O(1)$, so that $\tau(n) \leqslant T^{K(d)+O(1)}(n)$. Therefore, a program p exists that computes $\lceil -\log P(u,v|n) \rceil$ from n, u, v within time $T^{K(d)+O(1)}(n)$, and $l(p) \leqslant^+ K(d)$. Let c be the constant from inequality (11). Lemma 7.4 shows that for some $i = K(d) + c + O(1)$, we have:

$$K^{i-c}(x,y|n) \leqslant^+ K(d) + 2n + 4\log n - d_s(x,y).$$

Inequality (11) shows:

$$K^i(x,y) \leqslant 2n - d_s(x,y) + O(\log n).$$

Hence for $(x,y) \in R$,

$$\begin{aligned} d^i(x,y) &= K(x) + K(y) - K^i(x,y) \\ &\geqslant 2(n - \log n) - K^i(x,y) \\ &\geqslant d_s(x,y) - O(\log n) \\ &\geqslant d(x,y) - O(\log n). \end{aligned}$$

□

Notation: From now on all constants implicit in the $O()$ notation do not depend on i, whereas constants implicit in the $\leqslant^+$ notation may be dependent on i. For the proof of Lemma 7.9 we need Lemmas 7.6, 7.7 and 7.8.

Lemma 7.6. *For almost all i and all x, y with $l(x) = l(y) = n$, we have:*

$$\begin{aligned} &K^{i+O(1)}(x|n) + K^{i+O(1)}(y|x) \\ &\quad \leqslant^+ K^i(x,y|n) \\ &\quad \leqslant^+ K^{i-O(1)}(x|n) + K^{i-O(1)}(y|x). \end{aligned}$$

Proof. The second inequality follows from combining minimal programs from the definition of $K^{i-O(1)}(x|n)$ and $K^{i-O(1)}(y|x)$ into one program producing $\langle x, y\rangle$ from n in time $T^i(n)$. It remains to prove the first inequality. For all i large enough, we do this by defining a semimeasure $P(x, y|n)$ over all pairs of strings of length n:

$$P(x, y|n) = 2^{-K^i(x,y|n)}, . \tag{13}$$

The computable marginal and conditional semimeasures of P are:

$$\begin{aligned} P(x|n) &= \sum_{u:l(u)=n} P(x, u|n), \\ P(y|x) &= P(x, y|n)/P(x|n). \end{aligned} \tag{14}$$

Both measures are computable and can be evaluated in time $T^{i+O(1)}(n)$. Remark that the Kolmogorov complexity of these measures is bounded by $K(T^i)+O(1) \leqslant^+ 0$, since constants that only depend on i are absorbed in the $\leqslant^+$ notation. From Lemma 7.4 it follows that:

$$\begin{aligned} K^{i+O(1)}(x|n) &\leqslant^+ -\log P(x|n), \\ K^{i+O(1)}(y|x) &\leqslant^+ -\log P(y|x). \end{aligned} \tag{15}$$

The first inequality of the lemma follows from combining (13), (14) and (15). □

Lemma 7.7. *For almost all i and n, there exist strings x and a such that:*

- $l(a) = l(x) = n$
- $K^{i+O(1)}(a|n) \leqslant^+ 0$
- $K(x|n) \geqslant^+ n$
- $K^i(a|x) \geqslant^+ n$.

Proof. Let c be a large enough constant. Let a be the lexicographic first string of length n that cannot be produced from n by a program of length less than n in time less than $T^{i+c}(n)$. There is always such a string a. Obviously this string can be produced by running all possible programs for $T^{i+c}(n)$ steps, and searching for the lexicographic first string of length n that not has been output. This program needs a computation time bounded by $T^{i+2c}(n)$, for c large enough. To produce a from n in time $T^{i+2c}(n)$, it suffices to have a description of T^{i+c} and execute a constant amount of instructions. By this, the second condition is satisfied, since $K(T^{i+c})$ is absorbed in the $\leqslant^+$ notation.

There is at least one binary string of length n with $K(x|a) \geqslant n$. Pick one such string to be x. Note that $K(x|n) \geqslant^+ K(x|a) \geqslant n$, and by this the third condition is satisfied. By definition of a and x we find:

$$2n \leqslant^+ K^{i+c}(a|n) + K^{i+c}(x|a).$$

Let c_1 and c_2 correspond to the $O(1)$ constants in $K^{i+O(1)}$ and $K^{i-O(1)}$ from Lemma 7.6. Apply Lemma 7.6 for $i \to i+c_1$, and assume $c \geqslant c_1+c_2$:

$$2n \leqslant^+ K^i(x|n) + K^i(a|x).$$

Now it holds that $K(x|n) \leqslant^+ n$ [11], hence for i large enough we have $K^i(x|n) \leqslant^+ n$, and

$$2n \leqslant^+ n + K^i(a|x).$$

By this, the last condition is satisfied. □

Lemma 7.8. *For any function f and any set N, if*

$$\exists c \exists^\infty n \in N[n - c\log n < f(n)],$$

then

$$\forall c \exists^\infty n \in N[c\log n < f(n)].$$

Proof. Let c be a constant, and n_i, $i \in \omega$ be an infinite increasing sequence witnessing the first expression. For any c', take j large enough such that $n_j > (c + c')\log n_j$. Then the infinite sequence n_i, $i \geqslant j$, satisfies the second inequality. □

Lemma 7.9. *For all i, there is a j, such that $d^i \overset{\text{R}}{\not\geqslant} d^j$.*

Proof. We prove that there exists a constant c such that for all $i \geqslant c$, $d^{i-c} \overset{\text{R}}{\not\geqslant} d^{i+c}$. By the converse of the definition of R-domination it needs to be shown that:

$$\forall c' \exists^\infty (x,y) \in R\big[d^{i-c}(x,y) + c'\log n < d^{i+c}(x,y)\big].$$

By Lemma 7.8, it suffices to prove that

$$\exists c' \exists^\infty (x,y) \in R\big[d^{i-c}(x,y) + n - c'\log n < d^{i+c}(x,y)\big]. \qquad (16)$$

For any n large enough, pick x and a as in Lemma 7.7, and let $y = \mathrm{XOR}(x,a)$, where XOR is the bitwise exclusive-or operator. We now derive inequalities (17), (19), and (20).

- Note that $\mathrm{XOR}(y,a) = \mathrm{XOR}(\mathrm{XOR}(x,a),a) = x$. This provides a program for x given a and y. It follows that $K(x) \leqslant^+ K(y)+K(a|y)$ and hence:

$$\begin{aligned} K(y) &\geqslant^+ K(x) - K(a|y) \\ &\geqslant^+ K(x) - K^{i+O(1)}(a|n) \\ &\geqslant^+ K(x) \\ &\geqslant^+ n. \end{aligned} \tag{17}$$

  It follows that $(x,y) \in R$ for n large enough.

- Since $\mathrm{XOR}(y,x) = a$, it follows that any program computing y from x, also computes a from x. The extra time for this computation is bounded by some computable function. Therefore, for some c' large enough:

$$K^{i-c'}(y|x) \geqslant^+ K^i(a|x) \geqslant^+ n. \tag{18}$$

  Furthermore we have $K^i(x) \geqslant^+ n$. Hence, for $c - c'$ large enough, Lemma 7.6 can be applied with $i \to i - c$. Inequalities (11) and (18) imply:

$$\begin{aligned} K^{i-c}(x,y) &\geqslant^+ K^{i-c'}(x|n) + K^{i-c'}(y|x) - O(\log n) \\ &\geqslant^+ K(x|n) + n - O(\log n) \\ &\geqslant^+ 2n - O(\log n). \end{aligned} \tag{19}$$

- Since $\mathrm{XOR}(x,a) = y$, it follows for c' large enough, that

$$K^{i+2c'}(y|x) \leqslant^+ K^{i+c'}(a|x) \leqslant^+ 0.$$

  The last inequality follows from the second condition of Lemma 7.7. Remark that for i large enough, $K^{i+2c'}(x) \leqslant^+ n+2\log n$. Assuming $c - 2c'$ large enough, a bound for $K^{i+c}(x,y)$ can be derived using Lemma 7.6 with $i \to i + c$:

$$\begin{aligned} K^{i+c}(x,y) &\leqslant^+ K^{i+2c'}(x) + K^{i+2c'}(y|x) \\ &\leqslant^+ n + O(\log n). \end{aligned} \tag{20}$$

Combining inequalities (17), (19), (20), and $K(x) \geqslant^+ n$, we obtain

$$\begin{aligned} d^{i-c}(x,y) &\leqslant^+ K(x) + K(y) - K^{i-c}(x,y) \\ &\leqslant^+ O(\log n), \\ d^{i+c}(x,y) &\geqslant^+ K(x) + K(y) - K^{i+c}(x,y) \\ &\geqslant^+ n - O(\log n). \end{aligned}$$

Hence the constructed pair $(x, y) \in R$ satisfies

$$d^{i-c}(x, y) + n - O(\log n) \leqslant^+ d^{i+c}(x, y). \tag{21}$$

Such a pair can be constructed for every large enough i and n. This proves statement (16). □

Corollary 7.10. *Algorithmic mutual information*

$$I(x; y) = K(x) + K(y) - K(x, y)$$

is an independence test that R-dominates all Π_1^0-independence tests.

Proof. Because $K(x, y) = \inf_i\{K^i(x, y)\}$ it follows that:

$$I(x; y) = \sup_i\{d^i(x, y)\}.$$

By Lemma 7.5 it R-dominates all Π_1^0-independence tests. □

Corollary 7.11. *There exists a constant c, such that for all Σ_1^0-semi-measures P, Q, there exist a Π_1^0-independence test d for P, Q such that $d(x, y) \geqslant n - c \log n$ for infinitely many (x, y) with $l(x) = l(y) = n$.*

Proof. For some i large enough, there are infinitely many x, y with $l(x) = l(y)$ and

$$d^i(x, y) \geqslant n - c \log n - c_i,$$

where c_i is the constant implicit in the $\leqslant^+$ notation of (21). By universality of m, we have that $P(x) \leqslant 2^{c_P} m(x)$ and $Q(x) \leqslant 2^{c_Q} m(x)$, for some constants c_P, c_Q. Remark that $d(x) = d^i(x) - c_P - c_Q$ satisfies inequality (2), and is therefore a Π_1^0-independence test for P, Q. For $\log n > c_i + c_P + c_Q$ and infinitely many x, y with $l(x) = l(y)$ we have:

$$d(x, y) \geqslant n - (c + 1) \log n.$$ □

From the proof it also follows that

Corollary 7.12. *There is a constant c, such that for all Π_1^0-independence tests d, there is a Π_1^0-independence test d' with*

$$d'(x, y) - d(x, y) \geqslant n - c \log n,$$

for infinitely many x, y with $l(x) = l(y) = n$.

Proof. Note that for $i = K(d) + O(1)$ we have

$$d^i(x, y) - d(x, y) \geqslant n - c \log n - c_i.$$

Hence for all n with $\log n \geqslant c_i$ we have

$$d^i(x, y) - d(x, y) \geqslant n - (c + 1) \log n.$$ □

Acknowledgement We thank the anonymous referee for extensive comments on the paper.

References

[1] C. H. Bennett, P. Gacs, M. Li, P. M. B. Vitányi, and W. H. Zurek, *Information distance*, IEEE Transactions on Information Theory 44(4) (1998) 1407–1423.

[2] R. Cilibrasi and P. M. B. Vitányi, *Clustering by compression*, IEEE Transactions on Information Theory 51(4) (2005) 1523–1545.

[3] R. L. Cilibrasi and P. M. B. Vitányi, *The Google similarity distance*, IEEE Transactions on Knowledge and Data Engineering 19(3) (2007) 370–383.

[4] R. Downey and D. Hirschfeldt, *Algorithmic randomness and complexity*, Springer, forthcoming.

[5] P. Gács, *Uniform test of algorithmic randomness over a general space*, Theoretical Computer Science 341(1) (2005) 91–137.

[6] A. Gretton, R. Herbrich, A. Smola, O. Bousquet, and B. Schölkopf, *Kernel methods for measuring independence*, Journal of Machine Learning Research 6 (2005) 2075–2129.

[7] A. Hyvarinen, J. Karhunen, and H. Oja, *Independent component analysis*, Wiley, New York, 2001.

[8] C. J. Ku and T. L. Fine, *A Bayesian independence test for small datasets*, IEEE Transactions on Signal Processing 54(10) (2006) 4026–4031.

[9] L. A. Levin, *Randomness conservation inequalities; information and independence in mathematical theories*, Information and Control 61(1) (1984) 15–37.

[10] M. Li, X. Chen, X. Li, B. Ma, and P. Vitányi, *The similarity metric*, IEEE Transactions on Information Theory, 50(12) (2004) 3250–3264.

[11] M. Li and P. Vitányi, *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag, second edition, 1997.

[12] D. D. Mari and S. Kotz. *Correlations and dependence*, Imperial College Press, 2001.

[13] P. Pajunen, *Blind source separation using algorithmic information theory*, Neurocomputing 22(1) (1998) 35–48.

[14] B. Ryabko, J. Astola, and A. Gammerman, *Application of Kolmogorov complexity and universal codes to identity testing and nonparametric testing of serial independence for time series*, Theoretical Computer Science 359 (2006) 440–448.

[15] C. P. Schnorr, *Zufälligkeit und Wahrscheinlichkeit*, Lecture Notes in Mathematics 218, Springer, 1971.