Stability of Properties of Kolmogorov complexity under Relativization

An. A. Muchnik[†1] and A. E. Romashchenko[2]

[1] February 24, 1958 — March 18, 2007
[2] Kharkevich Institute for Information Transmission Problems, RAS, Moscow
anromash@mccme.ru

Abstract. Assume a tuple of binary strings $\bar a = \langle a_1, \dots, a_n\rangle$ has negligible mutual information with another string b. Does this mean that properties of Kolmogorov complexity of $\bar a$ do not change significantly if we relativize them to b? This question becomes very nontrivial when we try to formalize it. In this paper we investigate this problem for a special class of properties (for properties that can be expressed by an $\exists$-formula). In particular we show that a random (conditional on $\bar a$) oracle b does not help extract common information from a_i's.

1 Introduction

Kolmogorov complexity $K(x)$ of a binary string x is the minimal length of a program that generates x. Similarly, the conditional complexity $K(x\,|\,y)$ (complexity of x given y) is the minimal length of a program that prints x given y as an input. We talk about programs in one of *optimal* programming languages (see details of the definition in [1, 2]).

We may define Kolmogorov complexity (and conditional Kolmogorov complexity) not only for individual strings but also for pairs, triples, and all tuples of strings. To this end we fix some computable enumeration of all tuples, i. e., a computable bijection between binary strings and all tuples of binary strings. The Kolmogorov complexity of a tuples is defined as Kolmogorov complexity of the string assigned to this tuple in the enumeration. The choice of a particular enumeration does not matter: if we switch from one computable enumeration to another one, this changes Kolmogorov complexity of tuples by an additive term $O(1)$ only. This is not essential since even the Kolmogorov complexity of individual strings is defined only up to an additive $O(1)$ (which depends on the choice of an optimal programming language).

The main definitions and most results in the theory of Kolmogorov complexity easily relativize: instead of plain programs we can consider algorithms with an oracle, and most arguments about Kolmogorov complexity work with any oracle. Note that if an oracle is a finite object (a string) z we do not even need to introduce new notation to talk about Kolmogorov complexity with this oracle. Relativization to an oracle z means that we put z in conditions of all complexities; e.g., relativized version of $K(x)$ is $K(x\,|\,z)$, relativized $K(x\,|\,y)$ is $K(x\,|\,y,z)$, etc.

The information *about y contained in x* is defined as the difference between the Kolmogorov complexity of y and conditional Kolmogorov complexity of y given x:

$$I(x:y) = K(y) - K(y\,|\,x)$$

On of the most fundamental facts of algorithmic information theory is the theorem about symmetry of the mutual information:

Theorem 1 (Kolmogorov–Levin [1]). *For all strings x, y we have*

$$I(x:y) = I(y:x) + O(\log N) = K(x) + K(y) - K(x,y) + O(\log N),$$

$N = K(x,y)$.

Thus, up to a logarithmic term, we can talk about the mutual information between x and y, and make no distinction between $I(x:y)$ and $I(y:x)$.

If the mutual information $I(x:y)$ is negligible in comparison with $K(x)$, $K(y)$, $K(x,y)$ (e.g., if $I(x:y)$ is logarithmic in $N = K(x,y)$), then x and y are called *independent*. This (slightly informal) terminology is very popular in the information theory community. Is the usage of the word "independent" indeed well justified in this context? Intuitively it seems that if x an y are *independent*, then the *reasonable* algorithmic properties of x (expressed in the language of Kolmogorov complexity) should not change while we switch from the plain Kolmogorov complexity to Kolmogorov complexity relativized conditional on y.

Conjecture 1 (main conjecture). If the mutual information between $\langle x_1,\dots,x_n\rangle$ and a string z is negligible, then relativization to z does not change essentially the properties of $x_1,\dots,x_n$.

Main Conjecture in terms of optimal programming languages (a reformulation suggested by the anonymous referee): Assume we are interested in properties of Kolmogorov complexities of some particular tuple of strings $x_1,\dots,x_n$. Suppose that (for some technical reason) we extended our optimal programming language, but this extension does not change essentially Kolmogorov complexity of the tuple $\langle x_1,\dots,x_n\rangle$. On the other hand, complexities of some other strings could change dramatically. Conjecture 1 claims that all natural *closed* statements about $x_1,\dots,x_n$ (a *closed* statement may involve other strings except $x_1,\dots,x_n$, but they all should be bounded by quantifiers $\forall$ or $\exists$) are substantially equivalent for the original programming language and for its extension.

We formulated Conjecture 1 in a very vague way. To specify it, we consider several examples. We start with a very simple case. Let $\bar{x}$ be a tuple of n strings: $\bar{x} = \langle x_1, x_2, \dots, x_n\rangle$. Assume that the mutual information between $\bar{x}$ and some string z is negligible. Then it is not hard to see that the very basic properties of Kolmogorov complexity for $\bar{x}$ does not change when we relativize them conditional on z:

$$K(x_i) \approx K(x_i\,|\,z), \quad K(x_i,x_j) \approx K(x_i,x_j\,|\,z), \dots$$

for all i,j, etc. More precisely, the following proposition holds:

Proposition 1 *If $\bar{x} = \langle x_1, x_2, \dots, x_n\rangle$ and $\delta = K(\bar{x}) - K(\bar{x}|z)$, then for all indexes $i_1,\dots,i_s \in \{1,\dots,n\}$*

$$\bigl|K(x_{i_1},x_{i_2},\dots,x_{i_s}) - K(x_{i_1},x_{i_2},\dots,x_{i_s}\,|\,z)\bigr| \le \delta + O(\log N),$$

where $N = K(x_1,\dots,x_n,z)$. (The constant in the O-term depends on n but not on N).

This proposition is quite trivial; for the sake of completeness we prove it in Section 4.

Consider another example. We will need an existential quantifier to formulate the next property of Kolmogorov complexity. The following theorem about conditional descriptions was proven in [3]. Let us have a tuple of strings $x_1,\dots,x_n$. Then for any y there exists a string p such that:

1. $K(y\,|\,p,x_i) = O(\log N)$ for all $i = 1,\dots,n$ (where $N = K(x_1,\dots,x_n,y)$);
2. $K(p) = \max\limits_i K(y\,|\,x_i)$.

This theorem claims that there exists a program of length $\max\limits_i K(y|x_i)$ that translates each of x_i to y. (This statement becomes trivial if we change the length of the program from $\max\limits_i K(y\,|\,x_i)$ to the sum of all conditional complexities

$$K(y\,|\,x_1) + \dots + K(y\,|\,x_n).$$

Indeed, we could take as a program p the concatenation of n shortest programs translating each of x_i to y. The theorem about conditional descriptions is rather surprising: it claims that instead of this long concatenation we can take only one program, whose length is maximum of all $K(y\,|\,x_i)$. In fact this theorem can be made even stronger: we can add another requirement $K(p\,|\,y) = O(\log N)$, see [3].) Analyzing the proof of this theorem, it is easy to verify that the above property remain true relative to any oracle z: for every y there exists a program p' such that

1. $K(y\,|\,p', x_i, z) = O(\log N)$ for $i = 1, \ldots, n$;
2. $K(p'\,|\,z) = \max\limits_i K(y\,|\,x_i, z)$.

Here we do not even need to require that z is independent of $\langle x_1, \ldots, x_n\rangle$.

In the next example relativization is not that evident. We consider the property of *extracting common information*. Let $\bar{x} = \langle x_1, x_2\rangle$ be a pair of strings. We say that q bits ofcommon information can be extracted from this pair for a threshold k if

$$\exists y \text{ such that for } i = 1, 2 \text{ we have } K(y\,|\,x_i) < k \text{ and } K(y) \geq q.$$

It is easy to show that for every string y as above we have

$$K(y) \leq I(x_1 : x_2) + O(k + \log K(x_1, x_2)).$$

This means that for small enough thresholds k we cannot extract from x_1, x_2 much more bits of common information than $I(x_1 : x_2)$. (Loosely speaking, the value of extracted common information cannot be greater than the value of the mutual information.)

It is known that the question of extracting common information from a pair x_1, x_2 cannot be reduced to values of complexities $K(x_1)$, $K(x_2)$, and $K(x_1, x_2)$. For example, knowing that $K(x_1) = K(x_2) = 2N$ and $K(x_1, x_2) = 3N$, we cannot say anything precise about the value of extractible common information. On one hand, there exist pairs $\langle x_1, x_2\rangle$ with given complexities, such that N bits of common information can be extracted for a very small threshold $k = O(1)$. On the other hand, there exist pairs of words with the same complexities, such that for rather large thresholds k only negligible amount of common information (only $O(k + \log N)$ bits) can be extracted. See detailed discussion of extracting common information in [4–7].

The question of extracting common information can be defined not only for a pair but for any tuple of $s \geq 2$ strings. However many nontrivial properties become clear already for $s = 2$. So for simplicity we will talk about common information for pairs only.

Let us make conjecture 1 more specific for the property of extracting common information:

Let the mutual information between $\bar{x} = \langle x_1, x_2\rangle$ and z be very small. Then q bits of common information can be extracted from x_1 and x_2 for a threshold k if and only if the same q bits of common information can be extracted from these strings with oracle z (*maybe, for a slightly different threshold*).

This claim consists of two parts: *if* and *only if*. The second part (*only if*) is trivial: if some common information can be extracted without any oracle then the same information can be extracted also with an oracle. The interesting part is the *if* direction of this equivalence. We need to formulate it more precisely. We think that the most natural form of this statement involves logarithmic thresholds:

Conjecture 2. For every $C_1 > 0$ there exists a $C_2 > 0$ such that for all $\bar{x} = \langle x_1, x_2\rangle$ and z, if $I(z : \bar{x}) \leq C_1 \log N$ and

$$\exists v: \quad K(v\,|\,z) \geq q, \quad K(v\,|\,x_i, z) \leq C_1 \log N, \quad i = 1, 2,$$

where $N = K(\bar{x}, y)$ (i. e., q bits of common information can be extracted from x_1, x_2 for threshold $C_1 \log N$, given z as an oracle), then

$$\exists w: \quad K(w) \geq q, \quad K(w \,|\, x_i) \leq C_2 \log N, \quad i = 1, 2,$$

i. e., the same q bits of common information can extracted from these strings without any oracle (for threshold $C_2 \log N$).

Surprisingly, this natural conjecture is very nontrivial. It was proven in [8] only for $q = I(x_1 : x_2)$. In this paper we prove a simplified version of this conjecture: for all q but with $o(\cdot)$-term instead of logarithms.

Theorem 2. *For any function $f(N)$, $f(N) = o(N)$ there exists a function $g(N)$ (also $g(N) = o(N)$) such that for every $\bar{x} = \langle x_1, x_2 \rangle$ and z, if $I(z : \bar{x}) \leq f(N)$ and*

$$\exists v: \quad K(v \,|\, z) \geq q, \quad K(v \,|\, x_i, z) \leq f(N), \quad i = 1, 2,$$

where $N = K(\bar{x}, z)$, then

$$\exists w: \quad K(w) \geq q, \quad K(w \,|\, x_i) \leq g(N), \quad i = 1, 2.$$

We prove this theorem in Section 6.

Note that asymptotical results in algorithmic information theory are typically true not just with a term $o(\cdot)$, but with a more precise term $O(\log N)$ (though a few results about correlation between the Kolmogorov complexity of a string and statistics of its subwords involve terms $O(\sqrt{N})$, see [2]). We believe that Conjecture 2 is true in the logarithmic version; but the known technique is not enough to prove it.

We believe that similar properties hold relative to infinite oracles (computable or noncomputable). Probably, to prove this conjecture we need a new, more sophisticated technique.

So far we discussed a few very simple examples of statements specifying the general intuitive Conjecture 1. In the next section we suggest a formal framework suitable to formulate and prove this conjecture in a more general form.

2 General formulation of the main conjecture.

How stable are properties of Kolmogorov complexity when they are relativized? First of all, we need to specify the class of properties to consideration. Typically we are interested in general properties that hold "up to a logarithmic term".

Example 1 *For all x_1, x_2 it holds*

$$K(x_1, x_2) \leq K(x_1) + K(x_2) + O(\log K(x_1, x_2)) \tag{1}$$

and

$$K(x_1, x_2) \geq K(x_1) - O(1).$$

Also we have

$$K(x_1, x_2) = K(x_1) + K(x_2 \,|\, x_1) + O(\log K(x_1, x_2)).$$

These are the simplest properties of Kolmogorov complexity for a pair of strings, which can be expressed by *linear equalities and inequalities*. Note that even the very basic inequalities for Kolmogorov complexity are true only up to an additive logarithmic term.

Different types of Kolmogorov complexity (prefix, monotone, decision, a priori complexity, see [2, 9]) differ from each other by only an additive term of logarithmic order. So the properties that hold "with logarithmic precision" are the same for all kinds of Kolmogorov complexity. This is another reason to study properties of Kolmogorov complexity up to a logarithmic "remainder term".

How to describe this class of properties? For an n-tuple $x_1, \ldots, x_n$ we deal with Kolmogorov complexities of all tuples $x_{i_1}, \ldots, x_{i_k}$ for $1 \le i_1 < \ldots < i_k \le n$. Thus, to every n-tuple of strings there correspond $(2^n - 1)$ values of Kolmogorov complexity. We fix some order (e. g., lexicographical) on the set of all combinations of indexes $1 \le i_1 < \ldots < i_k \le n$, and correspond to every n-tuple of binary string its *complexity profile*, which is a vector in $\mathbb{Z}_+^{2^n-1}$

$$\boldsymbol{K}(x_1, \ldots, x_n) = (K(x_1), K(x_1, x_2), \ldots, K(x_2), K(x_2, x_3), \ldots).$$

In the same way we define *conditional complexity profile* $\boldsymbol{K}(x_1, \ldots, x_n \,|\, y)$ as a vector of conditional complexities of all combinations of x_i given y (we give precise definitions in Section 3).

Remark 1. There is no reason to distinguish tuples that contain the same strings but in different order. Indeed, changing the order of strings (or duplicating some strings) we change the Kolmogorov complexity of a tuple by only an additive $O(1)$ (which depends on the number of strings in the tuple but not on their complexities).

For the same reason, we do not need to involve in complexity profiles conditional complexities. By the Kolmogorov–Levin theorem, we can represent every conditional complexity as a combination of simple complexities:

$$K(x_1, \ldots, x_n \,|\, y_1, \ldots, y_m) = K(x_1, \ldots, x_n, y_1, \ldots, y_m) - K(y_1, \ldots, y_m) + O(\log N),$$

where $N = K(x_1, \ldots, x_n, y_1, \ldots, y_m)$.

Thus, the simplest Kolmogorov properties of a tuple $x_1, \ldots, x_n$ are just properties of its complexity profile $\boldsymbol{K}(x_1, \ldots, x_n)$. For example, inequality (1) is the following property of complexity profile of a pair x_1, x_2:

$$\boldsymbol{K}(x_1, x_2) = (K(x_1), K(x_1, x_2), K(x_2)) \in A = \{(u, v, w) :\ v \le u + w + O(\log v)\}.$$

Note that this simple property provides a constraint for the set of triples of integers that can represent some complexity profile. Hence, not every vector from $\mathbb{Z}_+^{2^n-1}$ is a profile of some tuple of strings (even up to additive logarithmic terms).

A more general class of Kolmogorov properties can be expressed in a similar form: we take some set of integers A and claim that *for all* $x_1, \ldots, x_n$

$$\boldsymbol{K}(x_1, \ldots, x_n) \in A.$$

Here set A represents a universal property of Kolmogorov complexity. In particular, we can express in this way any statement about a linear information inequality (see [8, 10–12]).

Even statements expressed in this simple form can be nontrivial. We have no complete description of all points in $\mathbb{Z}_+^{2^n-1}$ that are complexity profile of some tuples of strings. Moreover, for $n > 3$

we have no description of all linear information inequalities that hold up to an additive logarithmic term, see a discussion in [12].

It is natural to say that properties as above are atomic formulae in the language of Kolmogorov complexity. We can add to such a formula quantifiers $\forall$ (for each involved variable); then we get a false or true universal statement about Kolmogorov comeplxity.

Now we introduce more sophisticated properties of Kolmogorov comeplxity, which involve alternation of quantifiers. The simplest example is the property of extracting common information (discussed in Section 1). Several other examples of this type are investigated in [7]. A very general class of Kolmogorov properties for a tuple $x_1, \dots, x_n$ can be expressed by a formula as follows

$$\forall y_1 \exists y_2 \forall y_3 \dots \ \boldsymbol{K}(x_1, \dots, x_n, y_1, \dots, y_m) \in A, \tag{2}$$

where $A \subset \mathbb{Z}^{2^{n+m}-1}$. Assume that for tuples $\bar{x} = \langle x_1, \dots, x_n \rangle$ and $\bar{x}' = \langle x'_1, \dots, x'_n \rangle$ and some constant $C > 0$, for every property (2) of $\bar{x}$ there exists a similar property of $\bar{x}'$

$$\forall y_1 \exists y_2 \forall y_3 \dots \ \boldsymbol{K}(x'_1, \dots, x'_n, y_1, \dots, y_m) \in A'$$

where A and A' are C-close (i. e., for every point $\bar{a} \in A$ there exists a point $\bar{a}' \in A'$ such that Euclidian distance between $\bar{a}$ and $\bar{a}'$ is less than C, and vice versa, for every point in A' there exists a C-close point in A). Then we say that properties of the tuples $\bar{x}$ and $\bar{x}'$ are C-close.

Now we can discuss stability of these properties under relativization. Let O be an oracle (a finite or infinite binary sequence). We consider Kolmogorov complexity relativized conditional to this oracle. If O is finite, relativized complexity is just the usual conditional Kolmogorov complexity. In this paper we will consider only finite oracles O (usually we will denote by z a binary string encoding an oracle).

In the sequel we use asymptotic notations $O(f(x_1, \dots, x_n))$ involving Kolmogorov complexity. We always mean that the constant in O-terms may depend on the choice of an optimal programming language. In the rest of the section we formulate the main results of this paper.

Quantifier-free formulae. The elementary Kolmogorov properties (i. e., a property expressed by a quantifier-free formula) of a tuple $\bar{x}$ does not change under relativization to an oracle z if and only if the mutual information between $\bar{x}$ and z is negligible. This trivial statement is a reformulation of Proposition from Introduction:

Theorem 3. *Assume that for some $\bar{x} = \langle x_1, \dots, x_n \rangle$ and a string z the mutual information is small*:

$$I(\bar{x} : z) \le \delta.$$

Then the corresponding components of complexity profiles $\boldsymbol{K}(\bar{x})$ and $\boldsymbol{K}(\bar{x} \,|\, z)$ differ from each other by at most $\delta + O(\log K(\bar{x}, z))$.

$\exists$*-formulae.* Consider Kolmogorov properties expressed by $\exists$-formulae (with parameters). In this case our main conjecture can be reformulated as a pair of mutually inverse statements (a theorem and a conjecture):

Theorem 4. *Assume that for a tuple of strings $\bar{x}$ and a string z*

$$I(\bar{x} : z) \le \delta.$$

Then for every $\bar{y} = \langle y_1, \dots, y_m \rangle$ there exists a $\bar{y}' = \langle y'_1, \dots, y'_m \rangle$ such that the difference between corresponding components of complexity profiles $\boldsymbol{K}(\bar{x}, \bar{y})$ and $\boldsymbol{K}(\bar{x}, \bar{y}' \,|\, z)$ is at most $\delta + O(\log K(\bar{x}, \bar{y}, z))$.

Conjecture 3. Assume that for a tuple of strings $\bar{x}$ and a string z

$$I(\bar{x} : z) \le \delta.$$

Then for every $\bar{y} = \langle y_1, \dots, y_m \rangle$ there exists a $\bar{y}' = \langle y'_1, \dots, y'_m \rangle$ such that the difference between corresponding components in complexity profiles $\boldsymbol{K}(\bar{x}, \bar{y} \,|\, z)$ and $\boldsymbol{K}(\bar{x}, \bar{y}')$ is at most $\delta + O(\log K(\bar{x}, \bar{y}, z))$.

Conjecture 2 is a special case of Conjecture 3. We can prove Conjecture 3 only for stochastic tuples.

Definition 1. *A binary string x is called (α, β)-stochastic is there exists a set A that contains x, and*

- *complexity of a tuple $\widehat{A}$, which is lexicographically ordered list of all elements of A, is at most α;*
- *$K(x \,|\, \widehat{A}) \ge \log |A| - \beta$ (here and in the sequel all logarithms are to the base 2).*

Note that every incompressible string of length N (i. e., a string x of length N such that $K(x) \ge N$) is stochastic.

The definition of individual stochastic strings straightforwardly extends to tuples of strings. We are mostly interested in $(O(\log N), O(\log N))$-stochastic tuples $\bar{x}$ such that $N = K(\bar{x})$. For the sake of brevity we call them *stochastic* (without parameters α and β).

The fact that some strings are not stochastic, is quite nontrivial [13]. In most applications of Kolmogorov complexity, only stochastic strings or tuples of strings are used. So, the following theorem covers the most natural and important (for applications) case:

Theorem 5. *Conjecture 3 holds for all stochastic $\bar{x}$.*

For non-stochastic tuples Conjecture 3 remains unproven.

Tuples non-equivalent to any stochastic object. We say that strings a and b are C-equivalent ($a \sim_C b$) if

$$K(a \,|\, b) \le C \log K(a, b) \quad \text{and} \quad K(b \,|\, a) \le C \log K(a, b).$$

Further, we say that tuples $\bar{a} = \langle a_1, \dots, a_n \rangle$ and $\bar{b} = \langle b_1, \dots, b_n \rangle$ are C-equivalent (denote $\bar{a} \sim_C \bar{b}$) if for each $i = 1, \dots, n$ string a_i is C-equivalent to b_i.

Since we study Kolmogorov properties only with logarithmic precision, we may say that equivalent tuples have the same properties. So a natural idea is to prove Conjecture 3 as follows: first reduce every tuple $\bar{x}$ to some stochastic $\bar{x}'$ and then apply Theorem 5 to $\bar{x}'$. There is only one problem in this plan: can we find, for an arbitrary $\bar{x}$, an equivalent $\bar{x}'$?

First of all we note that for an individual x (a tuple of length 1) there exists a C-equivalent $(C \log K(x), C \log K(x))$-stochastic string x'. (Constant C does not depend on x.) Indeed, for every x there exists a shortest description p such that

$$K(p \,|\, x) = O(\log K(x)).$$

We can take this p as x'. This string is obviously C-equivalent to x (where C is a constant that depends only on the choice of an optimal programming language). The same time $K(x') \ge |x'| - O(1)$ since this string is a shortest description of x. Hence, x' is stochastic.

Is a similar statement true for every pair of strings $\langle x_1, x_2\rangle$? If x_1 and x_2 are independent, then we find equivalent strings to each of them (denote these strings x_1' and x_2'). It is easy to verify that the pair $\langle x_1', x_2'\rangle$ is stochastic and C-equivalent to $\langle x_1, x_2\rangle$. Another example: assume that $K(x_1 \,|\, x_2) \le O(\log K(x_2))$. Then we can find a shortest description p of x_1 and a shortest description q of x_2 conditional on x_1, such that x_1 is equivalent p and x_2 is equivalent to $\langle p, q\rangle$. This pair $\langle p, \langle p, q\rangle\rangle$ is stochastic.

Theorem 6. *Let $\alpha, \beta, \gamma, C > 0$ be reals such that $\alpha + \beta > \gamma$ and $\alpha, \beta < \gamma$. Then for all large enough n there exist strings x_1, x_2 such that*

- $K(x_1) = \alpha n + O(\log n)$;
- $K(x_2) = \beta n + O(\log n)$;
- $K(x_1, x_2) = \gamma n + O(\log n)$,

and there is no $(C \log n, C \log n)$-stochastic pair $\langle x_1', x_2'\rangle$ that is C-equivalent to $\langle x_1, x_2\rangle$.

Remark 2. It is easy to check that for every triple of reals (α, β, γ) such that $0 \le \alpha \le \gamma$, $0 \le \beta \le \gamma$, and $\gamma \le \alpha + \beta$, for all n there exist a pair x_1, x_2 such that

$$\begin{aligned} K(x_1) &= \alpha n + O(\log n),\\ K(x_2) &= \beta n + O(\log n),\\ K(x_1, x_2) &= \gamma n + O(\log n). \end{aligned}$$

In the above theorem we impose additional constraints on the values α, β, γ. The inequality $\alpha+\beta > \gamma$ means that x_1 and x_2 are not independent; the inequalities $\alpha, \beta < \gamma$ means that one of the strings x_i cannot be very simple conditional on another one. Without these constraints the theorem does not hold (see the examples before the theorem).

Thus, for some tuples of two strings there is no equivalent stochastic tuple. This means that techique of Theorem 5 does not help to prove Conjecture 3 in the general case.

Organization of the paper. In Section 3 we give principal definitions and formulate several technical lemmas (mostly know from other papers; for the sake of completeness we prove these lemmas in the Appendix). In Section 4 we prove the main conjecture for quantifier-free formulae and for $\exists$-formulae and stochastic tuples. In Section 5 we explain why non-stochastic tuples (even non-stochastic pairs) in general cannot be substituted by an equivalent stochastic tuple. In Section 6 we prove a version of the main conjecture for the property of extracting common information. In Section 7 we make conclusions and comment on open problems. In the Appendix we prove several technical lemmas borrowed from [10, 11] and a Lemma 6 (a generalization of the main lemma from [14, 15]). In Sections 4–6 we use different techniques, so these sections can be read independently.

3 Definitions, notation, and technical lemmas.

3.1 Complexity profile.

Let $\bar{x} = \langle x_1, \ldots, x_n\rangle$ be an n-tuple of binary strings. For every $V = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$, $1 \le i_1 < \ldots < i_k \le n$, we denote $\bar{x}_V$ the sub-tuple of strings x_j for $j \in V$:

$$\bar{x}_V = \langle x_{i_1}, \ldots, x_{i_k}\rangle.$$

Respectively, $K(\bar{x}_V) := K(x_{i_1}, \ldots, x_{i_k})$. If $V = \varnothing$ we assume that $K(\bar{x}_V) := K(\lambda)$ (where λ is the empty string). We use similar notation for conditional complexities: for every sets

$$V = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\} \text{ and } W = \{j_1, \ldots, j_l\} \subseteq \{1, \ldots, n\}$$

we denote $K(\bar{x}_V \,|\, \bar{x}_W) := K(x_{i_1}, \ldots, x_{i_k} \,|\, x_{j_1}, \ldots, x_{j_l})$. If W is empty, then we assume that $K(\bar{x}_V \,|\, \bar{x}_W) := K(\bar{x}_V \,|\, \lambda)$.

Definition 2. *By the* complexity profile *of an n-tuple $\bar{x} = \langle x_1, \ldots, x_n \rangle$ we call the vector of $(2^n - 1)$ values $K(\bar{x}_W)$ for all subsets $W \subseteq \{1, \ldots, n\}$ (we assume that subsets W are ordered lexicographically):*

$$\boldsymbol{K}(x_1, \ldots, x_n) = (K(x_1), K(x_1, x_2), \ldots, K(x_2), K(x_2, x_3), \ldots).$$

Similarly, by the conditional complexity profile *of an n-tuple $\bar{x}$ conditional on $y$we call the vector of $(2^n - 1)$ values $K(\bar{x}_W \,|\, y)$ computed for all subsets $W \subseteq \{1, \ldots, n\}$ (again, we assume that subsets W are ordred lexicographically):*

$$\boldsymbol{K}(x_1, \ldots, x_n \,|\, y) = (K(x_1 \,|\, y), K(x_1, x_2 \,|\, y), \ldots, K(x_2 \,|\, y), K(x_2, x_3 \,|\, y), \ldots).$$

Definition 3. *By the* extended complexity profile *of an n-tuple $x_1, \ldots, x_n$ we call the vector of complexites $K(\bar{x}_V \,|\, \bar{x}_W)$ for all pairs $V, W \subseteq \{1, \ldots, n\}$ such that $V \cap W = \varnothing$ and $V \neq \varnothing$. Note that an extended complexity profile contains unconditional complexities of $\bar{x}$: for $W = \varnothing$ we have $K(\bar{x}_V \,|\, \bar{x}_\varnothing) = K(\bar{x}_V) + O(1)$. We assume that the order of components in an extended complexity profile corresponds to the lexicographical ordering of pairs (V, W):*

$$\boldsymbol{K}'(x_1, \ldots, x_n) = (K(x_1), K(x_1 \,|\, x_2), \ldots, K(x_2), K(x_2 \,|\, x_1), K(x_2 \,|\, x_3), \ldots).$$

In the same way we define conditional extended complexity profile *of an n-tuple $x_1, \ldots, x_n$ conditional on y. It is composed of complexities of the form $K(\bar{x}_V \,|\, \bar{x}_W, y)$:*

$$\boldsymbol{K}'(x_1, \ldots, x_n \,|\, y) = (K(x_1 \,|\, y), K(x_1 \,|\, x_2, y), \ldots, K(x_2 \,|\, y), K(x_2 \,|\, x_1, y), K(x_2 \,|\, x_3, y), \ldots).$$

We will need to compare complexity profiles of different tuples. To this end we introduce a (partial) order on vectors in $\mathbb{R}^k$. We say that a vector $\bar{\alpha} \in \mathbb{R}^n$ is not greater than $\bar{\beta} \in \mathbb{R}^n$ (notation: $\bar{\alpha} \leq \bar{\beta}$) if $\alpha_i \leq \beta_i$ for all $i = 1, \ldots, n$.

We use l_∞-norm to measure the distance between vectors:

$$\rho(\bar{\alpha}, \bar{\beta}) := \max_i \{|\alpha_i - \beta_i|\}.$$

In particular, we say that complexity profile of $\bar{x} = \langle x_1, \ldots, x_n \rangle$ is not greater than complexity profile of $\bar{y} = \langle y_1, \ldots, y_n \rangle$, if every component of the first profile is not greater than the corresponding component of the second profile, i. e., for every $V \subseteq \{1, \ldots, n\}$ it holds $K(\bar{x}_V) \leq K(\bar{y}_V)$. We also say that the distance between complexity profiles of tuples $\langle x_1, \ldots, x_n \rangle$ and $\langle y_1, \ldots, y_n \rangle$ is not greater than ε if for every subset of indexes V it holds $|K(\bar{x}_V) - K(\bar{y}_V)| \leq \varepsilon$.

3.2 Typization.

In this paper we use the *typization* technique introduced in [10, 11, 16].

Definition 4. *Let $\bar{x} = \langle x_1, \dots, x_n \rangle$ and $\bar{y} = \langle y_1, \dots, y_m \rangle$ be tuples of binary strings. By* typization of $\bar{x}$ conditional on $\bar{y}$ *we call the following set of n-tuples :*

$$T(\bar{x} \,|\, \bar{y}) := \{\bar{x}' = \langle x'_1, \dots, x'_n \rangle \mid \boldsymbol{K}'(\bar{x}', \bar{y}) \le \boldsymbol{K}'(\bar{x}, \bar{y})\}.$$

*Further, we call by k-*strong typization of $\bar{x}$ conditional on $\bar{y}$ *the following set of n-tuples:*

$$ST_k(\bar{x} \,|\, \bar{y}) := \{\bar{x}' = \langle x'_1, \dots, x'_n \rangle \mid \boldsymbol{K}'(\bar{x}', \bar{y}) \le \boldsymbol{K}'(\bar{x}, \bar{y}) \text{ and } \rho(\boldsymbol{K}'(\bar{x}', \bar{y}), \boldsymbol{K}'(\bar{x}, \bar{y})) \le k\}.$$

Note that $T(\bar{x} \,|\, \bar{y})$ can be algorithmically enumerated given $\bar{y}$ and the extended complexity profile $\boldsymbol{K}'(\bar{x}, \bar{y})$. This is not the case for $ST_k(\bar{x} \,|\, \bar{y})$ (we can effectively check that Kolmogorov complexity of some tuple is less than a given threshold, but we cannot check that complexity is not too small).

The method of typization is based on the following lemmas proven in [10, 11].

Lemma 1. *For every tuples $\bar{x} = \langle x_1, \dots, x_n \rangle$ and $\bar{y} = \langle y_1, \dots, y_m \rangle$*

$$\log |T(\bar{x} \,|\, \bar{y})| = K(\bar{x} \,|\, \bar{y}) + O(\log N),$$

where $N = K(\bar{x}, \bar{y})$.

Lemma 2. *For all integers n, m there exists a $C = C(n, m)$ such that for all n-tuples $\bar{x} = \langle x_1, \dots, x_n \rangle$ and m-tuples $\bar{y} = \langle y_1, \dots, y_m \rangle$*

$$|ST_{C \log N}(\bar{x} \,|\, \bar{y})| > \frac{1}{2} |T(\bar{x} \,|\, \bar{y})|,$$

where $N = K(\bar{x}, \bar{y})$.

For the sake of brevity we denote

$$ST(\bar{x} \,|\, \bar{y}) = ST_{C \log N}(\bar{x} \,|\, \bar{y}),$$

where C is the constant from Lemma 2. We call elements of $ST(\bar{x} \,|\, \bar{y})$ by *clones* of $\bar{x}$ conditional on $\bar{y}$.

We need the following simple technical result:

Lemma 3. *Let $\bar{x} = \langle x_1, \dots, x_n \rangle$ and $\bar{y} = \langle y_1, \dots, y_m \rangle$ be tuples of strings, and δ_1, δ_2 be integers. Then we have the following two statements*:

1) *For every $\bar{x}' = \langle x'_1, \dots, x'_n \rangle$, if*

$$\boldsymbol{K}'(\bar{x}', \bar{y}) \le \boldsymbol{K}'(\bar{x}, \bar{y}) + \delta_1 \boldsymbol{e}$$

and $K(\bar{x}', \bar{y}) \ge K(\bar{x}, \bar{y}) - \delta_2$, then

$$\boldsymbol{K}'(\bar{x}', \bar{y}) \ge \boldsymbol{K}'(\bar{x}, \bar{y}) - (2\delta_1 + \delta_2 + O(\log N))\boldsymbol{e}$$

(here $N = K(\bar{x}, \bar{y})$, and $\boldsymbol{e} = (1, \dots, 1)$).

2) *For every z and for every $\bar{x}' = \langle x'_1, \dots, x'_n \rangle$ such that $\boldsymbol{K}'(\bar{x}', \bar{y}) \le \boldsymbol{K}'(\bar{x}, \bar{y}) + \delta_1 \boldsymbol{e}$ and $K(\bar{x}', \bar{y} \,|\, z) \ge K(\bar{x}, \bar{y}) - \delta_2$, we have*

$$\boldsymbol{K}'(\bar{x}', \bar{y} \,|\, z) \ge \boldsymbol{K}'(\bar{x}, \bar{y}) - (2\delta_1 + \delta_2 + O(\log N))\boldsymbol{e},$$

where $\boldsymbol{e} = (1, \dots, 1)$ and $N = K(\bar{x}, \bar{y})$.

3.3 Combinatorial entropy.

Let $A \subseteq X_1 \times \ldots \times X_n$ be a set of n-tuple (usually X_i are finite sets of strings). For each set of indexes $I = \{i_1, \ldots, i_k\} \subseteq \{1, \ldots, n\}$ denote by $\pi_I(A)$ the projection of A onto the corresponding coordinates:

$$\pi_I(A) := \{\bar{x}_I \mid \bar{x} \in A\}.$$

For example, if $I = \{1, \ldots, n\}$, then we have $\pi_I(A) = A$.

Further, for every tuple $\bar{x} = \langle x_1, \ldots, x_k \rangle$ we denote by $\sigma_I(A \,|\, \bar{x})$ the section of A corresponding to the value $\bar{x}$ in the I-projection:

$$\sigma_I(A \,|\, \bar{x}) := \{\bar{y} \mid \bar{y} \in A \text{ such that } \bar{y}_I = \bar{x}\}.$$

Let $X_1, \ldots, X_n$ be finite sets and $A \subseteq X_1 \times \ldots \times X_n$ be some set of n-tuples. We use the following notation:

- $n_I(A)$ is the number of elements in $\pi_I(A)$;
- $n_{I|J}(A \,|\, \bar{x})$ is the number of elements in $\pi_I \sigma_J(A \,|\, \bar{x})$;
- $n_{I|J}(A) = \max\limits_{\bar{x} \in \pi_J(A)} n_{I|J}(A \,|\, \bar{x})$. In particular, if $J = \varnothing$, then $n_{I|J}(A) = n_I(A)$.

Definition 5. *Let $X_1, \ldots, X_n$ be finite sets and $A \subseteq X_1 \times \ldots \times X_n$ be some set of n-tuples. For all sets of indexes $I, J \subseteq \{1, \ldots, n\}$ we set*

- $\mathrm{ent}_I(A) := \lceil \log n_I(A) \rceil$*;*
- $\mathrm{ent}_{I|J}(A) := \lceil \log n_{I|J}(A) \rceil$ *(if $J = \varnothing$, then $\mathrm{ent}_{I|\varnothing}(A) = \mathrm{ent}_I(A)$).*

Lemma 4. *Let $A \subseteq \mathbb{B}^n$ be a finite set of n-tuples. Denote by $\mathrm{list}(A)$ the list of all elements A (in some computable encoding). Then for every $\bar{x} \in A$, for all $V, W \subseteq \{1, \ldots, n\}$*

$$K(\bar{x}_V \,|\, \bar{x}_W, \mathrm{list}(A)) \leq \mathrm{ent}_{V|W}(A) + O(1).$$

Proof. Given the list of all elements of A and a tuple $\bar{x}_W$, we can find the list of all elements in

$$B = \pi_V \sigma_W(A \,|\, \bar{x}_W).$$

Obviously, $\bar{x}_V \in B$. Now to describe $\bar{x}_V$ we only need to specify the ordinal number of $\bar{x}_V$ in the list of elements of B. The binary representation of this number takes at most $\lceil \log n_{V|W}(A) \rceil$ bits. □

3.4 Method of bunches.

The following definition of a *bunch* was introduced in [14]:

Definition 6. *An (α, β, γ)-*bunch *X is a set of strings such that*

1. *$|X| = 2^\alpha$;*
2. *$K(x_1 \,|\, x_2) < \beta$ for all $x_1, x_2 \in X$;*
3. *$K(x) < \gamma$ for all $x \in X$.*

Lemma 5 [14, 15]. *There exists an algorithm that takes a triple of integers α, β, γ as an input, and enumerates a list of (α, β, γ)-bunches $U_0, \ldots, U_q$ such that*:

- *for every* (α,β,γ)*-bunch* U *there exists* $i \le q$ *such that* $|U \cap U_i| \ge 2^{\beta-\varepsilon}$, *where* $\varepsilon = 2(\beta-\alpha)+O(1)$;
- $q < 2^{\beta+\gamma-2\alpha+O(1)}$.

We need to modify the definition of a bunch:

Definition 7. *An* (α,β,γ)-semi-bunch *is a set of strings* X *such that*

1. $|X| = 2^{\alpha}$;
2. *for each* $x_1 \in X$, *for the majority of* $x_2 \in X$ *it holds* $K(x_2\,|\,x_1) < \beta$;
3. $K(x) < \gamma$ *for all* $x \in X$.

The following result is similar to Lemma 5.

Lemma 6. *There exists an algorithm that takes a triple of integers* α,β,γ *as an input and enumerates a sequence of* (α,β,γ)*-semi-bunches* $U_0,\dots,U_q$ *such that*:

- *for every* (α,β,γ)*-semi-bunch* U *there exists* $i \le q$ *such that* $|U \cap U_i| \ge 2^{\beta-\varepsilon}$, *where* $\varepsilon = 2(\beta-\alpha)+O(1)$;
- $q < 2^{\beta+\gamma-2\alpha+O(1)}$.

This algorithm may never stop (*it prints only a finite number of semi-bunches* U_i, *but we never know whether the last bunch is already obtained*).

We will call the semi-bunches $U_0,\dots,U_q$ from Lemma 6 *standard semi-bumches* (for given parameters α,β,γ). The enumeration of standard semi-bunches is deterministic. Hence, for all α,β,γ and for every $i \le q(\alpha,\beta,\gamma)$ complexity of the list of all elements in a standard semi-bunch U_i given i is $O(\log\gamma)$.

In the Appendix we prove Lemmas 1–3 and Lemma 6.

4 Quantifier-free and existential formulae

Proof of Theorem 3: On one hand, for every subset of indexes $V \subseteq \{1,\dots,n\}$

$$K(\bar{x}_V\,|\,z) \le K(\bar{x}_V) + O(1).$$

On the other hand, since $I(\bar{x}_V : z) \le I(\bar{x} : z) + O(\log N)$, we have

$$K(\bar{x}_V) - K(\bar{x}_V\,|\,z) = I(\bar{x}_V : z) \le I(\bar{x} : z) + O(\log N) \le \delta + O(\log N).\square$$

Proof of Theorem 4: We need to prove that there exists a tuple $\bar{y}'$ such that the distance between $\boldsymbol{K}(\bar{x},\bar{y})$ and $\boldsymbol{K}(\bar{x},\bar{y}'\,|\,z)$ is at most $\delta + O(\log N)$. We prove a more strong statement: the distance between the corresponding *extended* profiles is not greater than $\delta + O(\log N)$.

By Lemma 1 the set $T(\bar{y}\,|\,\bar{x})$ contains $2^{K(\bar{y}|\bar{x})+O(\log N)}$ m-tuples. Hence, we can choose $\bar{y}' \in T(\bar{y}\,|\,\bar{x})$ such that $K(\bar{y}'\,|\,\bar{x},z) \ge K(\bar{y}\,|\,\bar{x}) - O(\log N)$. For the chosen $\bar{y}'$ we have

$$\begin{aligned} K(\bar{x},\bar{y}'\,|\,z) &= K(\bar{x}\,|\,z) + K(\bar{y}'\,|\,\bar{x},z) \\ &\ge K(\bar{x}) - \delta + K(\bar{y}\,|\,\bar{x}) - O(\log N) \\ &= K(\bar{x},\bar{y}) - \delta - O(\log N). \end{aligned}$$

We apply Lemma 3 and get

$$\rho\big(\boldsymbol{K}'(\bar{x},\bar{y}),\boldsymbol{K}'(\bar{x},\bar{y}'\,|\,z)\big) \le \delta + O(\log N).\square$$

We believe that Conjecture 3 (a generalization of Theorem 4) is true for all tuples. However, we can prove it only for *stochastic* tuples $\langle x_1,\dots,x_n\rangle$.

Proof of Theorem 5: Step 1. Denote $N = K(\bar{x}, \bar{y}, z)$. At first, instead of each $y_1, \dots, y_m$ we substitute the corresponding shortest program (in an optimal programming language). This substitution changes complexities in the extended profile $\boldsymbol{K}'(\bar{x}, \bar{y} \,|\, z)$ by at most $O(\log N)$. So, w.l.o.g. we may assume that $\bar{y} \in (\mathbb{B}^N)^m$. Since n-tuple $\bar{x}$ is stochastic, there exists $S \subset (\mathbb{B}^*)^n$ such that

$$K(S) = O(\log N) \text{ and } \log |S| = K(\bar{x}) + O(\log N).$$

Thus, $\langle \bar{x}, \bar{y} \rangle \in S \times (\mathbb{B}^N)^m$.

Step 2. Denote $A_0 = T(\bar{x}, \bar{y} \,|\, z) \cap \big(S \times (\mathbb{B}^N)^m\big)$. The sizes of sections and projections of A_0 are not greater than exponents in the corresponding values of the extended complexity profile $\boldsymbol{K}'(\bar{x}, \bar{y} \,|\, z)$. We say that a set $A \subset S \times (\mathbb{B}^N)^m$ is *correct* if all its combinatorial entropies $\mathrm{ent}_{I|J}(A)$ are not grater than the corresponding combinatorial entropies of A_0. In other words, A is correct if for all $I, J \subseteq \{1, \dots, (n+m)\}$

$$\log n_{I|J}(A) \le \mathrm{ent}_{I|J}(A_0).$$

In particular, the set A_0 is correct. Note that from correctness of A it follows that $n_{I|J}(A) < 2n_{I|J}(A_0)$.

Given extended profile $\boldsymbol{K}'(\bar{x}, \bar{y} \,|\, z)$ and the list of all elements of S, we can algorithmically find *all* correct sets (the list of all correct sets is very large, but it is finite!). Since complexity of the list of all elements of S is logarithmic, we get that the list of all correct sets also has complexity $O(\log N)$. Denote by $A_1, A_2, \dots$ the lexicographically ordered list of all correct sets.

By definition, the size of every section of a correct set is at most twice as large as the size of the corresponding section of A_0. Of course, sections of a correct set can be much smaller.

We call by *strong* projection of A_i onto the first n coordinates (i. e., onto S) the set B_i that consists of points corresponding to large enough sections:

$$B_i = \big\{ \bar{x}' \in \pi_{1,\dots,n}(A_i) \;\big|\; \log |\sigma_{1,\dots,n}(A_i \,|\, \bar{x}')| \ge \mathrm{ent}_{n+1,\dots,n+m|1,\dots,n}(A_0) - C_1 \log N \big\}$$

(the constant C_1 will be specified below). In particular, we denote B_0 the strong projection of A_0. We fix two algorithms: they take z, vector $\boldsymbol{K}'(\bar{x}, \bar{y} \,|\, z)$, and constant C_1 as an input, and enumerate A_0 and B_0 respectively.

To get $\bar{y}$ from $\bar{x}, z$ we need to know the profile $\boldsymbol{K}'(\bar{x}, \bar{y} \,|\, z)$ and specify the order number of $\bar{y}$ in the enumeration of all elements of A_0 corresponding to the given $\bar{x}$ (i. e., in the enumeration of $\sigma_{1,\dots,n}(A_0 \,|\, \bar{x})$). Hence,

$$K(\bar{y} \,|\, \bar{x}, z) \le \log |\sigma_{1,\dots,n}(A_0 \,|\, \bar{x})| + O(\log N).$$

By the definition of A_0, we have

$$\mathrm{ent}_{n+1,\dots,n+m|1,\dots,n}(A_0) \le K(\bar{y} \,|\, \bar{x}, z).$$

We sum up these two inequalities and get that for large enough C_1, tuple $\bar{x}$ belongs to B_0.

Step 3. We select from the list of correct sets some *special* subsequence as follows. Assume that correct sets $A_1, \dots, A_{s-1}$ are already examined, and $A_{i_1}, \dots, A_{i_k}$ are already selected as special. Then we examine the next correct set A_s; we include A_s in our special subsequence if the difference

$$B_s \setminus \left(\bigcup_{r \le k} B_{i_r} \right)$$

contains at least $2^{K(\bar{x}|z)-C_2\log N}$ elements (C_2 to be specified below).

Notice that the special subsequence contains at most

$$\frac{|S|}{2^{K(\bar{x}|z)-C_2\log N}} = 2^{I(\bar{x}:z)+C_2\log N+O(\log N)}$$

correct sets. Denote by $\widehat{A}$ the union of all correct sets from the defined special subsequence $A_{i_1}, A_{i_2}, \dots$ Denote by $\widehat{B}$ the projection of $\widehat{A}$ onto the first n coordinates.

Obviously, $K(\widehat{A}) = O(\log N + \log C_2)$ and $K(\widehat{B}) = O(\log N + \log C_2)$ since the lists of elements of these sets can be found algorithmically if we are given the extended profile $\boldsymbol{K}'(\bar{x}, \bar{y}\,|\,z)$, the set S and constant C_2.

Remark 3. For $\widehat{A}$ constructed above

$$\log n_{I|J}(\widehat{A}) \le \mathrm{ent}_{I|J}(A_0) + I(\bar{x}:z) + C_2\log N + O(\log N)$$

for all $I, J \subset \{1,\dots,n\}$. The list of all elements of $\widehat{A}$ has Kolmogorov complexity $O(\log N + \log C_2)$. Hence, from Lemma 4 it follows that for every $\bar{u} \in \widehat{A}$

$$K(\bar{u}_I\,|\,\bar{u}_J) \le \mathrm{ent}_{I|J}(A_0) + I(\bar{x}:z) + C_2\log N + O(\log N + \log C_2)$$

for all $I, J \subset \{1,\dots,n\}$.

Lemma 7. *The tuple $\bar{x}$ belongs to $\widehat{B}$ (for large enough $C_2 = C_2(n,m)$).*

Proof of lemma: Assume for the sake of contradiction that $\bar{x}$ does not belong to the strong projection of $\widehat{A}$. This means that A_0 (which is a correct set) was not selected for the special subsequence of correct sets. It follows that the cardinality of the difference $B_0 \setminus \widehat{B}$ is less than $2^{K(\bar{x}|z)-C_2\log N}$. So, to describe the n-tuple $\bar{x}$ given z and the extended profile $\boldsymbol{K}'(\bar{x},\bar{y})$, we need to specify the list of all elements of $\widehat{B}$ and the ordinal number of $\bar{x}$ in the natural enumeration of $B_0 \setminus \widehat{B}$. Hence,

$$\begin{aligned} K(\bar{x}\,|\,z) &\le \log|(B_0 \setminus \widehat{B})| + O(\log N + \log C_2) \\ &\le K(\bar{x}\,|\,z) - C_2\log N + O(\log N + \log C_2). \end{aligned}$$

We get a contradiction be choosing a large enough constant C_2. □

Step 4. Thus, $\bar{x} \in \widehat{B}$. Denote Q the section of $\widehat{A}$ corresponding to $\bar{x}$:

$$Q = \{\bar{y}' \mid \langle \bar{x}, \bar{y}' \rangle \in \widehat{A}\}.$$

From the construction of $\widehat{A}$ it follows that the number of elements in Q cannot be too small. More precisely,

$$\log|Q| \ge K(\bar{y}\,|\,\bar{x}, z) - O(\log N).$$

It remains to take in Q an m-tuple $\bar{y}'$ that has maximal possible complexity conditional on $\bar{x}$. By usual counting argument we get that for some $\bar{y}' \in Q$

$$K(\bar{y}'\,|\,\bar{x}) \ge \log|Q| \ge K(\bar{y}\,|\,\bar{x}, z) - O(\log N).$$

Using $K(\bar{x}) \ge K(\bar{x}\,|\,z) + I(\bar{x}:z) - O(\log N)$ we get

$$K(\bar{x}, \bar{y}') \ge K(\bar{y}'\,|\,\bar{x}) + K(\bar{x}) \ge K(\bar{x}, \bar{y}\,|\,z) + I(\bar{x}:z) - O(\log N).$$

On the other hand, from the construction of $\widehat{A}$ it follows (see above) that

$$\boldsymbol{K}'(\bar{y}'\,|\,\bar{x}) \le \boldsymbol{K}'(\bar{y}\,|\,\bar{x}) + (I(\bar{x}:z) - O(\log N))\boldsymbol{e}.$$

We apply Lemma 3 with $\delta_1 = -\delta_2 = I(\bar{x}:z)$ and get

$$\rho(\boldsymbol{K}'(\bar{x},\bar{y}'), \boldsymbol{K}'(\bar{x},\bar{y}\,|\,z)) \le I(\bar{x}:z) + O(\log N),$$

which completes the proof of the theorem. □

5 Pairs that have no equivalent stochastic pairs

Proof of Theorem 6: In this proof for the sake of brevity we call $(C\log n, C\log n)$-stochastic pairs simply *stochastic*, without specifying parameters (the constant C is taken from the condition of the theorem). Fix a large enough n. Denote by S_1 the set of all strings of length αn, and by S_2 the set of all strings of length βn. We will construct an effectively enumerable set $A \subset S_1 \times S_2$ of size $2^{\gamma n - O(\log n)}$. Further we will show that some element of A satisfies the theorem, i. e., it has the required complexity profile, and there is no C-equivalent stochastic pair. It is convenient to consider A as a bipartite graph with parts S_1 and S_2.

If x_1' is C-equivalent to some string in S_1, then complexity of x_1' is at most $\alpha n + 2C\log n$; denote by L_1 the set of all strings with such complexities. Similarly, if x_2' is C-equivalent to some string from S_2, then its complexity is at most $\beta n + 2C\log n$; we denote by L_1 the set of all strings with complexity at most $\beta n + 2C\log n$. Thus, if a pair $\langle x_1, x_2\rangle$ from A is C-equivalent to some pair $\langle x_1', x_2'\rangle$, then $x_1' \in L_1$ and $x_2' \in L_2$. Also we can bound the complexity of $\langle x_1', x_2'\rangle$:

$$K(x_1', x_2') \le K(x_1, x_2) + K(x_1'\,|\,x_1) + K(x_2'\,|\,x_2) + O(\log(K(x_1'\,|\,x_1) + K(x_2'\,|\,x_2))),$$

which means

$$K(x_1', x_2') \le \gamma n + C\log n + C\log n + O(\log\log n) < \gamma n + 3C\log n.$$

By definition, a stochastic pair $\langle x_1', x_2'\rangle$ must belong to some R such that a) complexity of the list of all elements of R is less than $C\log n$, and b) the number of elements in R is not greater than $2^{K(x_1', x_2') + C\log n}$. These conditions mean that $R \subset L_1 \times L_2$ such that:

- $|R| \le 2^{\gamma n + 4C\log n}$;
- $K(R) \le C\log n$, i. e., the list of all elements of R has complexity less than $C\log n$.

The number of all sets R as above is not greater than $2^{C\log n}$. Given the number of all such sets (we need $C\log n$ bits to specify this number), we can enumerate all these sets R. Denote by $\widehat{R}$ the union of these sets. It is also convenient to consider $\widehat{R}$ as the set of edges in a bipartite graph, where the two parts are L_1 and L_2.

To construct this bipartite graph with the set of edges $\widehat{R}$ we need $O(\log n)$ bits of information. Further we construct a graph A which has many edges that are not equivalent to any edge in $\widehat{R}$.

There is one obstacle: the relation of C-equivalence is not computable. To overcome this problem, we define a small and computable class of relations (that contains C-equivalence as a special case). We call by *nearness relation* any

$$D \subset S_1 \times L_1 \cup S_2 \times L_2,$$

satisfying the following two conditions (for $i = 1, 2$):

- for every $x \in S_i$ there are at most $2^{C\log n+1}$ elements $y \in L_i$ such that $\langle x, y\rangle \in D$;
- for every $y \in L_i$ there are at most $2^{C\log n+1}$ elements $x \in S_i$ such that $\langle x, y\rangle \in D$.

The relation of C-equivalence

$$D_0 = \{\langle x, y\rangle \in S_1 \times L_1 \cup S_2 \times L_2 \ : \ x \sim_C y\}$$

is a closeness relation (though there are many other closeness relation). Obviously, the number of closeness relations is not greater than

$$(|L_1|^{\mathrm{poly}(n)})^{|S_1|} \cdot (|L_2|^{\mathrm{poly}(n)})^{|S_2|}.$$

W.l.o.g we may assume that $\alpha \geq \beta$. Then number of different closeness relations is $2^{2^{\alpha n+O(\log n)}}$. We say that an edge $\langle x_1, x_2\rangle \in A$ is D-close to $\langle x_1', x_2'\rangle \in \widehat{R}$ if $\langle x_i, x_i'\rangle \in D$ for $i = 1, 2$.

Now we are ready to construct a graph A. The construction will depend on three integer parameters C_1, C_2, C_3 (independent of n). We specify appropriate values C_1, C_2, C_3 below.

We say that an edge $\langle x_1, x_2\rangle \in A$ is *suitable* if degrees of x_1 and x_2 are not greater than $2^{(\gamma-\alpha)n+C_3\log n}$ and $2^{(\gamma-\beta)n+C_3\log n}$ respectively. We will need that

$$|A| = 2^{\gamma n - C_1 \log n},$$

and the following condition holds:

> *For every closeness relation D there exists at most $2^{\gamma n - C_2\log n}$ suitable edges $\langle x_1, x_2\rangle \in A$ that are not D-close to any edge in $\widehat{R}$.* (3)

Below we construct such a graph A effectively, i. e., its complexity will be $O(\log n)$. At least $2^{\gamma n - C_2\log n}$ suitable edges in A are not C-equivalent to any stochastic pair. It remains to select from this graph an edge of complexity at least $\gamma n - C_2 \log n$, and we are done. Indeed, for this edge $\langle x_1, x_2\rangle$ we have

$$K(x_1) \leq \alpha n + O(1) \text{ and } K(x_2) \leq \beta n + O(1)$$

(since $x_1 \in S_1$ and $x_2 \in S_2$), and

$$K(x_2 \,|\, x_1) \leq (\gamma - \alpha)n + O(\log n) \qquad \text{and} \qquad K(x_1 \,|\, x_2) \leq (\gamma - \beta)n + O(\log n)$$

(since $\langle x_1, x_2\rangle$ is a suitable edge). If C_1 is chosen large enough, we get also

$$K(x_1, x_2) < \gamma n.$$

From Lemma 3 part 2) (with δ_1 and δ_2 of order $O(\log n)$) it follows that the inequalities above become equalities (with logarithmic precision). In particular, we get

$$K(x_1) = \alpha n + O(\log n), \qquad K(x_2) = \beta n + O(\log n), \qquad K(x_1, x_2) = \gamma n + O(\log n),$$

and the theorem is proven.

Thus, it remain to construct A. Note that for a given graph, property (3) can be verified algorithmically. We will prove that for a randomly chosen set of $2^{\gamma n - C_1 \log n}$ edges in $S_1 \times S_2$, condition (3) is true with positive probability. So, a required graph A exists, and we can find it by a brute-force search (if there are many graphs that satisfy (3), we chose lexicographically first one).

Fix one closeness relation D. We say that an edge $\langle x_1, x_2\rangle \in S_1 \times S_2$ is *spoiled* if it is D-close to some edge in $\widehat{R}$. Let us count the probability for a random edge in $S_1 \times S_2$ to be spoiled. Graph $\widehat{R}$ contains $2^{\gamma n + O(\log n)}$ edges. For each of them there are $\mathrm{poly}(n)$ D-close edges in $S_1 \times S_2$. Hence,

$$\mathrm{Prob}[\langle x_1, x_2\rangle \text{ is spoiled}] \le \frac{2^{\gamma n + O(\log n)}}{2^{(\alpha+\beta)n}} \ll 1/2.$$

Here we used condition $\alpha + \beta > \gamma$.

Let $k = 2^{\gamma n - C_1 \log n}$ and $l \le 2^{\gamma n - C_2 \log n + 1}$. Then probability that $(k-l)$ of k randomly chosen edges are spoiled, is not greater than

$$\binom{k}{l}(1/2)^{k-l} \le k^l (1/2)^{k-l} \le 2^{2^{\gamma n - C_2 \log n + O(\log n)}} (1/2)^{2^{\gamma n - C_1 \log n + O(\log n)}},$$

which is equal to $2^{-2^{\gamma n - O(\log n)}}$, provided that the difference between C_1 and C_2 is large enough. Sum up this probability for $l = 0, \ldots, 2^{\gamma n - C_2 \log n + 1}$. Obviously, multiplying this probability by the number of different l (i. e., by $2^{\gamma n}$), we do not change substantially its asymptotics.

Thus, we bounded the probability that a random A contains too many spoiled edges for a fixed closeness relation D. Further, sum up this probability for all closeness relations. Having in mind the number of different closeness relations, we get that the probability for a random A to have too many spoiled edges for at least one relation D, is at most

$$\frac{2^{2^{\alpha n + O(\log n)}}}{2^{2^{\gamma n - O(\log n)}}} \ll 1.$$

Here we used the assumption $\alpha < \gamma$ (recall that $\beta \le \alpha$).

Thus, we see that in a random graph A with probability close to 1, for all closeness relation D (including the relation of C-equivalence) there are at least $2^{\gamma n - C_2 \log n + 1}$ edges that are not equivalent to any edge in $\widehat{R}$. However, some of these edges can be not suitable (if one of its vertices has too large degree). It remains to prove that with high probability (close to 1) the number of non-suitable edges is not greater than $2^{\gamma n - C_2 \log n}$. From this bound it follows immediately that with positive probability in a random A there are at least $2^{\gamma n - C_2 \log n}$ edges that are at the same time suitable and non-spoiled, i. e., satisfy (3).

Fix any ordinal number of an edge in A (this is an integer j between 1 and $|A|$). Denote by x and y the left and right ends of this edge (in S_1 and S_2 respectively). Vertices x and y in a random A can be incident to some other edges. We need to bound the probability that in a random A the jth edge is not suitable, i. e., degrees of x or y are too large.

We may think of the distribution on random graphs A as follows: first we choose at random ends of the jth edge, and then randomly choose the remaining $(|A|-1)$ edges. The average number of edges (except the jth) that are incident to x in a random A, is equal to

$$\frac{|A|-1}{|S_1|} < 2^{(\gamma-\alpha)n - C_1 \log n}.$$

From Chebyshev's inequality it follows that the probability that x is incident to at least $2^{(\gamma-\alpha)n + C_3 \log n}$ edges, is not greater than $1/n^{C_1+C_3}$. A similar bound holds for y. Thus, the probability that the jth edge of A is not suitable, is not greater than $2/n^{C_1+C_3}$.

The proven bound holds for each $j = 1, \ldots, |A|$. Hence, expectation of the number of non-suitable edges is not greater than

$$2|A|/n^{C_1+C_3} \le 2^{\gamma n - 2C_1 \log n - C_3 \log n + 1}.$$

Apply Chebyshev's inequality again: the probability that the number of non-suitable edges is greater than $2^{\gamma n - C_2 \log n}$, is bounded by $2n^{C_2}/n^{2C_1+C_3}$. It remains to choose C_3 large enough, and the probability above tends to zero for large n. Thus, we have proven that there exists a graph A that satisifies (3).

It remains to comment on how we chose parameters C_1, C_2, C_3. Every edge in A has Kolmogorov complexity

$$\gamma n - C_1 \log n + O(\log n) + O(\log(C_1 + C_2 + C_3))$$

(with some absolute constant in $O(\cdot)$-terms). The value C_1 should be chosen so large, that complexity of every edge in A is not greater than γn. Further, we choose C_2 so that the difference $(C_2 - C_1)$ is large enough (greater than some absolute constant, as we explained above). At last, C_3 should be chosen greater than $(C_2 - 2C_1)$. □

6 Extracting common information

Proof of Theorem 2: At first we introduce several notation and accept some assumptions. W.l.o.g. we may assume that $f(N) > \log N$, and $f(N)$ is a monotone function (that is, $f(N+1) \ge f(N)$ for all N).

Further, we choose $g(N)$ and $\delta(N)$ that increase not too fast and not too slowly (so that the construction of our proof works well). There is a certain degree of freedom in choice of these functions. We fix the following definitions:

$$\delta(N) = N \Big/ \sqrt{\log \frac{N}{f(N)}} \qquad \text{and} \qquad g(N) = 3^{C\sqrt{\log \frac{N}{f(N)}}} f(N) + 2\delta(N)$$

(constant C to be specified later). For the sake of brevity we will write δ (without an argument) if the value of N is clear from the context.

Informal idea. The main trick of the proof is typization of v and z conditional on $\bar{x}$. We take the set of "clones" of pair $\langle v, z \rangle$, which all have the same conditional complexity profile (conditional on $\bar{x}$). Then there are two cases to consider:

Simple case. Assume that the set of clones is well consolidated: most clones have large mutual information. Then we apply Lemma 6 and extract from the set of clones a common kernel w. This w contains q bits of information, and it must be very simple conditional on any x_i. Thus, we extract q bits of common information from strings x_i, and we are done

Difficult case. Assume that the set of clones is not well consolidated. Then there exists a pair of clones that have very small mutual information. On this stage we cannot extract from x_i their common information. Then we substitute instead of z another string z_1 such that with the new oracle z_1 we can extract from x_1, x_2 at least q_1 bits of common information (where q_1 is much greater than q). Thus, we convert the original problem about strings x_1, x_2, an oracle z_1, and a parameter q to a similar problem with the same strings x_1, x_2, a new oracle z and a new parameter

q_1. We increase the value q but we pay for it: the original threshold $f(N)$ should be substituted by rougher bound $f_1(N)$.

Let us explain how to construct the new oracle z_1. Recall that the set of "clones" is not well consolidated. We choose at random two clones: $\langle v', z' \rangle$ and $\langle v'', z'' \rangle$. We claim that pair $\langle z', z'' \rangle$ can be used as z_1. Indeed, with the new oracle we can extract from strings x_i both v' and v''. The strings v' and v'' together result in q_1 bits of common information ($q_1 > q$; more precisely, $q_1 \geq q + \delta/2$).

We iterate the above trick again and again, until at some stage we achieve a well consolidated set of clones.

In the rest of the section we expose this plan in full detail.

Formal argument. By hypothesis of the theorem, there exists a string v such that $K(v \,|\, x_i, z) \leq f(N)$ for $i = 1, 2$. W.l.o.g. we may assume that $q = K(v \,|\, z)$ (if $K(v \,|\, z) > q$, we can increase parameter q; this make the statement of the theorem even stronger). Denote $m = K(z)$. We need to construct a string w such that $K(w \,|\, x_i) \leq g(N)$ and $K(w) \geq q - g(N)$.

We take the strong typization of $\langle v, z \rangle$ conditional on $\bar{x}$: set $A = ST(v, z \,|\, \bar{x})$. By Lemma 1 and Lemma 2 we get $|A| = 2^{K(v,z|\bar{x}) - O(f(N))}$. Further, we have

$$K(v, z \,|\, \bar{x}) = K(z \,|\, \bar{x}) + K(v \,|\, z, \bar{x}) + O(\log N),$$

and $K(z \,|\, \bar{x}) \geq K(z) - f(N)$ (the mutual information between z and $\bar{x}$ is negligible), and similarly $K(v \,|\, z, \bar{x}) \leq f(N)$ (the string v is easy to extract from each x_i given the oracle z). Hence, $|A| = 2^{m - O(f(N))}$. Note that for every $\langle v', z' \rangle \in A$ it holds

$$K(v', z') = K(z') + K(v' \,|\, z) + O(\log N) = m + q + O(f(N)).$$

Further, we have two cases to consider:

Case 1^0. For every $\langle v', z' \rangle \in A$, for the majority of $\langle v'', z'' \rangle \in A$

$$I(v'z' : v''z'') \geq q - \delta.$$

This inequality means that

$$K(v'z' \,|\, v''z'') = K(v', z') - I(v'z' : v''z'') \leq m + \delta + O(f(N)).$$

Thus, A is a semi-bunch with parameters

$$(m - O(f(N)), m + \delta + O(f(N)), m + q + O(f(N))).$$

We apply Lemma 6 and get that there exists a *standard* bunch U_j (with the same parameters) such that

$$|A \cap U_j| \geq 2^{m - \delta + O(f(N))}$$

and the number j is not greater than $2^{q + \delta + O(f(N))}$. In other words, the Kolmogorov complexity of j is not greater than $q + \delta + O(f(N))$.

Further, for the strings x_i, $i = 1, 2$ we have two properties:

- For every pair $\bar{u} \in A \cap U_j$ it holds $K(x_i \,|\, \bar{u}) \leq K(x_i \,|\, v, z)$ (by definition of $A = ST(v, z \,|\, \bar{x})$); recall that $K(x_i \,|\, v, z) \leq K(x_i) - q + f(N)$;
- For every pair $\bar{v} \in A \cap U_j$ it holds $K(\bar{u} \,|\, j) \leq \log |U_j| + O(\log N) \leq m$ (given j, we can algorithmically enumerate elements of semi-bunch U_j).

These two properties (and a bound for the number of elements in $A \cap U_j$) imply that strings x_1 and x_2 belong to the following sets $X(1)$ and $X(2)$ respectively:

$$X(i) = \left\{ \widehat{x} \;\middle|\; \begin{array}{l} \text{there exists at least } 2^{m-\delta+O(f(N))} \text{ strings } \bar{u} \text{ such that} \\ K(\widehat{x}\,|\,\bar{u}) \le K(x_i) - q + f(N) \text{ and } K(\bar{u}\,|\,j) \le m \end{array} \right\}.$$

We can enumerate $X(i)$ given j and $O(\log N)$ bits of additional information: binary representations of numbers m, $m - \delta + O(f(N))$ and $K(x_i) - q + f(N)$ (i. e., the numbers involved in the definition of $X(i)$).

Now we can find an upper bound for the size of $X(i)$. For a fixed j there are at most 2^{m+1} different tuples $\bar{u}$ such that $K(\bar{u}\,|\,j) \le m$; for each $\bar{u}$ there are at most $2^{K(x_i)-q+f(N)}$ different $\widehat{x}$ such that $K(\widehat{x}\,|\,\bar{u}) \le K(x_i) - q + f(N)$. Since for every $\widehat{x} \in X(i)$ there should be *at least* $2^{m-\delta+O(f(N))}$ different tuples $\bar{u}$, we get

$$\log |X(i)| \le \log \frac{2^m \cdot 2^{K(x_i)-q+f(N)}}{2^{m-\delta+O(f(N))}} \le K(x_i) - q + \delta + O(f(N)).$$

Thus, $K(x_i\,|\,j) \le K(x_i) - q + \delta + O(f(N))$ (in other words, the mutual information between j and x_i is not less than $q - \delta - O(f(N))$). From the symmetry of the mutual information it follows

$$K(j\,|\,x_i) = K(x_i\,|\,j) + K(j) - K(x_i) + O(\log N) \le 2\delta + O(f(N)).$$

Set $w = j$. Since $K(w) \ge I(w : x_i) \ge q - \delta - O(f(N))$, we have $K(w) \ge q - g(N)$ and $K(w\,|\,x_i) \le g(N)$ (for threshold $g(N)$ defined in the beginning of the proof). Thus, in Case 1^0 we are done.

Case 2^0. Consider the case when for some $\langle v', z' \rangle \in A$, for the majority of $\langle v'', z'' \rangle \in A$ it holds

$$I(v'z' : v''z'') < q - \delta.$$

This inequality means that

$$K(v'v''z'z'') \ge 2m + q + \delta - O(\log N). \tag{4}$$

By our assumption, this inequality holds for the majority of $\langle v'', z'' \rangle \in A$. We select from all these pairs $\langle v'', z'' \rangle$ the tuple that has maximal Kolmogorov complexity conditional on $\langle x, v', z' \rangle$. This maximal Kolmogorov complexity is close to the number of elements in A (this is a standard counting argument: the number of shorter programs is too small to serve the majority of elements in A).

Kolmogorov complexity of the selected pair $\langle v'', z'' \rangle$ conditional on x and conditional on $\langle x, v', z' \rangle$ is equal approximately to $\log |A|$. In other words, $\langle v', z' \rangle$ and $\langle v'', z'' \rangle$ are independent conditional on $\bar{x}$. It follows that z' and z'' are independent conditional to $\bar{x}$ (i. e., $I(z' : z'' \,|\, \bar{x}) = O(\log N)$). Further, for all strings $\bar{x}, z', z''$ it holds

$$I(z'z'' : \bar{x}) \le I(z' : \bar{x}) + I(z'' : \bar{x}) + I(z' : z'' \,|\, \bar{x}) + O(\log N)$$

This inequality is the sum of two elementary properties of Kolmogorov complexity:

$$K(z'z'') \le K(z') + K(z'') + O(\log N),$$
$$K(z'\,|\,x) + K(z''\,|\,x) = K(z'z''\,|\,x) + I(z' : z''\,|\,x) + O(\log N),$$

(the last equation easily follows from the Kolmogorov–Levin theorem [1]). For the strings under consideration, the values $I(z' : \bar{x})$ and $I(z'' : \bar{x})$ are not greater than $f(N)$ (i. e., $\bar{x}$ and z are independent), and $I(z' : z'' \,|\, \bar{x}) = O(\log N) \ll f(N)$. So, we get

$$I(z'z'' : \bar{x}) \le 3f(N). \tag{5}$$

By the definition of A, the complexities of the strings z', z'' are not greater than $m = K(z)$. Hence, complexity of $\langle z', z'' \rangle$ not greater than $2m + 2\log m$. We apply (4) to $z_1 = \langle z', z'' \rangle$ and $v_1 = \langle v', v'' \rangle$ to get

$$K(v_1 \,|\, z_1) \ge K(v_1 z_1) - K(z_1) - O(\log N) \ge q + \delta - 3f(N) - O(\log N) \ge q + \delta/2.$$

Thus, we obtained a string z_1 such that $I(z_1 : \bar{x}) \le 3f(N)$ (from (5)) and

$$\exists v_1 : \quad K(v_1 \,|\, z_1) \ge q + \delta/2, \quad K(v_1 \,|\, x_i, z_1) \le 3f(N), \quad i = 1, 2.$$

We summarize the results. Instead of the original pair $\langle v, z \rangle$ we get a new one $\langle v_1, z_1 \rangle$. By the construction, z_1 is independent of $\bar{x}$ (though "precision" of independence becomes three times worse: $I(z_1 : \bar{x}) \le 3f(N)$). String v_1 is simple conditional on each x_i given oracle z_1 ("simple" means that conditional Kolmogorov complexity is not greater than $3f(N)$). Complexity of v_1 conditional on z_1 is not less than $q + \delta/2$. Thus, $q + \delta/2$ bits of common information can be extracted from x_1, x_2 with threshold $3f(N)$ relative to oracle z_1. Note that the complexities of v_1, z_1 are not greater than $3N$.

Then we iterate the argument above. We repeat the same trick with v_1, z_1. Denote $q_1 = q + \delta/2$, $m_1 = K(z_1)$, and $f_1(N) = 3f(N)$. Take the strong typization of $\langle v_1, z_1 \rangle$ conditional on $\bar{x}$:

$$A^1 = ST(v_1, z_1 \,|\, \bar{x}).$$

Again, there are two cases:

Case 1^1. Assume that for every $\langle v', z' \rangle \in A^1$ for the majority of $\langle v'', z'' \rangle \in A^1$ it holds

$$I(v'z' : v''z'') \ge q_1 - \delta.$$

Then A^1 is a semi-bunch with parameters

$$(m_1 - O(f_1(N)), m_1 + \delta + O(f_1(N)), m_1 + q_1 + O(f_1(N))).$$

By Lemma 6 there exists a number j such that for $i = 1, 2$

$$K(j \,|\, x_i) \le 2\delta + O(f_1(N)) \text{ and } I(j : x_i) \ge q_1 - \delta + O(f_1(N)).$$

Similarly to Case 1^0, we set $w := j$, and we are done.

Case 2^1. Assume that for each $\langle v', z' \rangle \in A^1$ for the majority of pairs $\langle v'', z'' \rangle \in A^1$ we have $I(v'z' : v''z'') < q_1 - \delta$. Then there exists a pair $\langle v_2, z_2 \rangle$ such that

1. $K(z_2) = m_2 < 3m_1$;
2. $I(z_2 : \bar{x}) \le f_2(N) := 3f_1(N)$;
3. $K(v_2 \,|\, z_2, x_i) \le f_2(N)$;
4. $K(v_2 \,|\, z_2) = q_2 \ge q_1 + \delta/2$.

We iterate this construction again and again; on each step s we get strings v_s, z_s such that

1. $K(z_s) = m_s = 3m_{s-1}$;
2. $I(z_s : \bar{x}) \le f_s(N) := 3f_{s-1}(N) = 3^s f(N)$;
3. $K(v_s \,|\, z_s, x_i) \le f_s(N)$;
4. $K(v_s \,|\, z_s) = q_s > q_{s-1} + \delta/2 = q + s\delta/2$.

We iterate the construction in Cases $2^1, 2^2, 2^3 \ldots, 2^j, \ldots$, until at some stage $s_{\max}$ we get Case $1^{s_{\max}}$.

This iterative procedure cannot be too long. Indeed, in $s = C\sqrt{\log \frac{N}{f(N)}}$ steps (provided that constant C is chosen large enough) we get a contradiction with inequality

$$K(v_s \,|\, z_s) \le K(v_s \,|\, x_1, z_s) + K(v_s \,|\, x_2, z_s) + I(x_1 : x_2 \,|\, z_s) + O(\log N)$$

(it is not hard to show that this inequality is true for all strings; see, e. g., the proof of (6) in [10]). Indeed, the left-hand side of this inequality is at least $CN/2$, and the right-hand side is

$$2f_s(N) + I(x_1 : x_2 \,|\, z_s) + O(\log N) = O(N)$$

(with some absolute constant in O-term).

Thus, after several iterations of Case 2^s, at stage $s_{\max} < C\sqrt{\log \frac{N}{f(N)}}$ we come to Case $1^{s_{\max}}$. This means that we obtain a string w such that

$$K(w) \ge q + s_{\max}\delta/2 - O(f_{s_{\max}}(N)) > q - g(N)$$

and

$$K(w \,|\, x_i) \le 2\delta + f_{s_{\max}} < 2\delta + 3^{C\sqrt{\log \frac{N}{f(N)}}} f(N) < g(N), \quad i = 1, 2.$$

In a word, q bits of common information are extracted from x_i's with threshold $g(N)$.

Remark 4. In the argument above we ignore additive terms of order $O(\log K(y^s, w^s))$. This is legal since $\log K(v_s, z_s) \ll f(N)$. Indeed, for all involved strings $K(v_s), K(z_s) < N^2$ since $s \ll \log N$. Note also that at each next step s_i we need greater threshold $f_i(N)$. In the very beginning of the proof we chose $g(N)$ such that all $f_i(N)$ under consideration are less than $g(N)$. □

7 Conclusion

Our results do not give complete answers to the posed questions. Conjecure 3 is unproven for non-stochastic tuples. Even a more specific Conjecture 2 remains an open problem (the statement of our Theorem 2 looks not very natural beacuse we used $o(N)$-terms instead of logarithmic thresholds). It would also be interesting to find justifications or counterexamples for the main intuitive Conjecture 1 for general properties with several alternations of quantifiers. We have no results about relativization with infinite oracles. We suppose that further progress in these problems requires developing new techniques to be developed.

We are grateful to the participants of the Kolmogorov seminar of the department of mathematics and mechanics at Moscow state university for many fruitful discussions. We are particularly thankful to the anonymous referee for very helpful criticism and detailed comments.

Here we prove technical lemmas used in the main text.

Proof of Lemma 1: First of all, for every $\bar{x}' \in T(\bar{x}\,|\,\bar{y})$

$$K(\bar{x}'\,|\,\bar{y}) \le K(\bar{x}\,|\,\bar{y}).$$

Hence, the number of such tuples $\bar{x}'$ is not greater than $2^{K(\bar{x}|\bar{y})+1}$. Further, we get a lower bound for the size of $T(\bar{x}\,|\,\bar{y})$. Note that we can algorithmically enumerate the list of elements of $T(\bar{x}\,|\,\bar{y})$ given $\bar{y}$ and all numbers of the complexity profile $\boldsymbol{K}'(\bar{x}\,|\,\bar{y})$ (though we cannot know when this enumerating is completed unless we are given very large supplementary information). So, to get $\bar{x}$ from $\bar{y}$, we need to know complexity profile $\boldsymbol{K}(\bar{x}\,|\,\bar{y})$ and the ordinal number of $\bar{x}$ in this enumeration. It follows that

$$K(\bar{x}\,|\,\bar{y}) \le \log|T(\bar{x},\bar{y})| + O(\log N),$$

which provides a lower bound for the size of $T(\bar{x}\,|\,\bar{y})$. □

Proof of Lemma 2: By Lemma 1 we have

$$|T(\bar{x}\,|\,\bar{y})| \ge 2^{K(\bar{x}|\bar{y})-C\log N}$$

for some constant C. Hence, for at least half of tuples $\bar{x}' \in T(\bar{x}\,|\,\bar{y})$ Kolmogorov complexity conditional on $\bar{y}$ is not less than

$$K(\bar{x}\,|\,\bar{y}) - C\log N - 1.$$

Let $ST(\bar{x}\,|\,\bar{y})$ be the set of all such tuples $\bar{x}' \in T(\bar{x}\,|\,\bar{y})$.

By the construction, every $\bar{x}' \in ST(\bar{x}\,|\,\bar{y})$ also belongs to $T(\bar{x}\,|\,\bar{y})$; hence,

$$\boldsymbol{K}'(\bar{x}',\bar{y}) \le \boldsymbol{K}'(\bar{x},\bar{y}).$$

It remains to prove the inverse inequality (up to an additive $O(\log N)$). Thus, for all $V_1, V_2 \subset \{1,\dots,n\}$ and $W_1, W_2 \subset \{1,\dots,m\}$ we need to show that

$$K(\bar{x}'_{V_1},\bar{y}_{W_1}\,|\,\bar{x}'_{V_2},\bar{y}_{W_2}) \ge K(\bar{x}_{V_1},\bar{y}_{W_1}\,|\,\bar{x}_{V_2},\bar{y}_{W_2}) - O(\log N). \tag{6}$$

To this end, we note that

$$K(\bar{x}',\bar{y}) = K(\bar{x}'_{V_2},\bar{y}_{W_2}) + K(\bar{x}'_{V_1},\bar{y}_{W_1}\,|\,\bar{x}'_{V_2},\bar{y}_{W_2}) + K(\bar{x}',\bar{y}'\,|\,\bar{x}'_{V_1\cup V_2},\bar{y}_{W_1\cup W_2}) + O(\log N).$$

The right-hand side of this inequality is not greater than

$$K(\bar{x}_{V_2},\bar{y}_{W_2}) + K(\bar{x}'_{V_1},\bar{y}_{W_1}\,|\,\bar{x}'_{V_2},\bar{y}_{W_2}) + K(\bar{x},\bar{y}\,|\,\bar{x}_{V_1\cup V_2},\bar{y}_{W_1\cup W_2}) + O(\log N),$$

since $\bar{x}' \in T(\bar{x}\,|\,\bar{y})$. Further, from

$$\begin{aligned} K(\bar{x}',\bar{y}) + O(\log N) &= K(\bar{x},\bar{y}) \\ &= K(\bar{x}_{V_2},\bar{y}_{W_2}) + K(\bar{x}_{V_1},\bar{y}_{W_1}\,|\,\bar{x}_{V_2},\bar{y}_{W_2}) + K(\bar{x},\bar{y}'\,|\,\bar{x}_{V_1\cup V_2},\bar{y}_{W_1\cup W_2}) + O(\log N), \end{aligned}$$

we get (6). □

Proof of Lemma 3: The proof is similar to the previous argument. It is enough to prove a more general claim 2). For all U_1, U_2, U_3, U_4 we have

$$K(\bar{x}'_{U_1},\bar{y}_{U_3}\,|\,\bar{x}'_{U_2},\bar{y}_{U_4},z) \le K(\bar{x}'_{U_1},\bar{y}_{U_3}\,|\,\bar{x}'_{U_2},\bar{y}_{U_4}) + O(1) \le K(\bar{x}_{U_1},\bar{y}_{U_3}\,|\,\bar{x}_{U_2},\bar{y}_{U_4}) + \delta_1.$$

For the sake of contradiction, assume that for some V_1, V_2, W_1, W_2

$$K(\bar{x}'_{V_1}, \bar{y}_{W_1} \,|\, \bar{x}'_{V_2}, \bar{y}_{W_2}, z) < K(\bar{x}_{V_1}\bar{y}_{W_1} \,|\, \bar{x}_{V_2}, \bar{y}_{W_2}) - \delta - D\log N,$$

Then (similarly to the proof of Lemma 2) we have

$$K(\bar{x}', \bar{y} \,|\, z) < K(\bar{x}, \bar{y}) - \delta + 2\delta_1 - D\log N + O(\log N).$$

Set $\delta = 2\delta_1 + \delta_2$, and we get a contradiction for large enough D. □

Proof of Lemma 6: Fix an algorithm that takes a triple of integers α, β, γ as an input and enumerates the list of *all* (α, β, γ)-semi-bunches. We call this algorithm by *simple enumerator*. Though the number of semi-bunches (for every triple of parameters) is finite, the simple enumerator never stops, since we cannot decide when all semi-bunches are already found. We only guarantee that each semi-bunch is generated by the simple enumerator at some moment.

Now we construct another enumerator that selects some subsequence from the list of semi-bunches generated by the simple enumerator. It works as follows. We run the simple enumerator and examine the semi-bunches that it returns one by one. We *select* some semi-bunches from this list by the following rule. Assume that semi-bunches $U_0, \ldots, U_s$ are already *selected*; let the simple enumerator return another semi-bunch V. Denote $\varepsilon = 2(\beta - \alpha + 2)$. If $|V \cap U_i| < 2^{\beta-\varepsilon}$ for every $i = 0, \ldots, s$, then we *select* this semi-bunch and let $U_{s+1} = V$. Otherwise we skip V and wait for the next semi-bunch from the simple enumerator.

Let $U_0, \ldots, U_q$ be the list of all semi-bunches that are selected in this procedure (for some parameters α, β, γ). From the construction it follows that for every semi-bunch V either $V = U_i$ or at least $|V \cap U_i| \geq 2^{\beta-\varepsilon}$ (for some $i \leq q$). Also it follows from the construction that $|U_i \cap U_j| < 2^{\beta-\varepsilon}$ for every two *selected* semi-bunches U_i, U_j. It remains to show that q is not too large.

It is enough to prove that every string x belongs to less than $2^{\beta-\alpha+2}$ semi-bunches. Indeed, there exist less than 2^γ strings x such that $K(x) < \gamma$. If every x is covered by at most $2^{\beta-\alpha+2}$ selected semi-bunches U_i, and every semi-bunch consists of at most 2^α strings, then the number of selected semi-bunches is not greater than

$$\frac{2^\gamma \cdot 2^{\beta-\alpha+2}}{2^\alpha} = 2^{\beta+\gamma-2\alpha+2}.$$

Thus, it remains to bound the number of selected semi-bunches that cover one string x.

Fix a string x and assume that there are $N = 2^{\beta-\alpha+2}$ different semi-bunches U_i that contain x. Denote

$$U'_i = U_i \cap \{y \mid K(y \,|\, x) < \beta\}$$

for all these semi-bunches U_i. From the definition of semi-bunches it follows that U'_i contains at most $2^{\alpha-1}$ elements. On one hand we have

$$\left|\bigcup U'_i\right| \leq |\{y \mid K(y \,|\, x) < \beta\}| < 2^\beta.$$

On the other hand,

$$\left|\bigcup U'_i\right| \geq \sum_i |U'_i| - \sum_{i<j} |U'_i \cap U'_j|.$$

Since $|U'_i| \geq 2^{\alpha-1}$ and $|U'_i \cap U'_j| \leq |U_i \cap U_j| \leq 2^{\beta-\varepsilon}$, we have

$$\left|\bigcup U'_i\right| \geq N \cdot 2^{\alpha-1} - N^2 \cdot 2^{\beta-\varepsilon} = 2^\beta,$$

and we get a contradiction. □

References

1. Zvonkin, A.K. and Levin, L.A., Complexity of Finite Objects and the Algorithmic Concepts of Information and Randomness, *Uspekhi Mat. Nauk*, 1970, vol. 25, no. 6, pp. 85–127 [*Russian Math. Surveys* (Engl. Transl.), 1970, vol. 25, no. 6, pp. 83–124].
2. Li, M. and Vitányi, P., *An Introduction to Kolmogorov Complexity and Its Applications*, New York: Springer, 2008, 3rd ed.
3. Muchnik, An.A., Conditional Complexity and Codes, *Theoret. Comput. Sci.*, 2002, vol. 271, no. 1–2, pp. 97–109.
4. Gács, P. and Körner, J., Common Information Is Far Less than Mutual Information, *Probl. Control Inform. Theory*, 1973, vol. 2, no. 2, pp. 149–162.
5. Ahlswede, R. and Körner, J., Appendix: On Common Information and Related Characteristics of Correlated Information Sources, *General Theory of Information Transfer and Combinatorics*, Ahlswede, R., Bäumer, L., Cai, N., Aydinian, H.K., Blinovsky, V., Deppe, C., and Mashurian, H., Eds., Lect. Notes Comp. Sci., vol. 4123, Berlin: Springer, 2006, pp. 664–677.
6. Muchnik, An.A., On Common Information, *Theoret. Comput. Sci.*, 1998, vol. 207, no. 2, pp. 319–328.
7. Chernov, A., Muchnik, An.A., Romashchenko, A., Shen, A., and Vereshchagin, N.K., Upper Semi-lattice of Binary Strings with the Relation "x Is Simple Conditional to y," *Theoret. Comput. Sci.*, 2002, vol. 271, no. 1–2, pp. 69–95.
8. Romashchenko, A.E., Pairs of Word with Nonmaterializable Mutual Information, *Probl. Peredachi Inf.*, 2000, vol. 36, no. 1, pp. 3–20 [*Probl. Inf. Trans.* (Engl. Transl.), 2000, vol. 36, no. 1, pp. 1–18].
9. Uspensky, V.A. and Shen, A., Relations between Varieties of Kolmogorov Complexities, *Math. Syst. Theory*, 1996, vol. 29, no. 3, pp. 271–292.
10. Hammer, D., Romashchenko, A., Shen, A., and Vereshchagin, N., Inequalities for Shannon Entropy and Kolmogorov Complexity, *J. Comput. Syst. Sci.*, 2000, vol. 60, no. 2, pp. 442–464.
11. Makarychev, K., Makarychev, Yu., Romashchenko, A., and Vereshchagin, N., A New Class of Non-Shannon-type Inequalities for Entropies, *Commun. Inf. Syst.*, 2002, vol. 2, no. 2, pp. 147–162.
12. Zhang, Z., and Yeung, R.W., On Characterization of Entropy Function via Information Inequalities, *IEEE Trans. Inform. Theory*, 1998, vol. 44, no. 4, pp. 1440–1452.
13. Shen', A.Kh., The Concept of Kolmogorov (α, β)-Stochasticity and Its Properties, *Dokl. Akad. Nauk SSSR*, 1983, vol. 271, no. 6, pp. 1337–1349 [*Soviet Math. Doklady* (Engl. Transl.), 1983, vol. 28, pp. 295–299].
14. Romashchenko, A., Extracting the Mutual Information for a Triple of Binary Strings, in *Proc. 18th IEEE Annual Conf. on Computational Complexity* (*CCC'03*)*, Aarhus, Denmark, 2003*, pp. 221–229.
15. Romashchenko, A.E., A Criterion for Extractability of Mutual Information for a Triple of Strings, *Probl. Peredachi Inf.*, 2003, vol. 39, no. 1, pp. 166–175 [*Probl. Inf. Trans.* (Engl. Transl.), 2003, vol. 39, no. 1, pp. 148–157].
16. Romashchenko, A., Shen, A., and Vereshchagin, N., Combinatorial Interpretation of Kolmogorov Complexity, *Theoret. Comput. Sci.*, 2002, vol. 271, no. 1–2, pp. 111–123.