On Extracting the Mutual Information for a Triple of Binary Strings

A. Romashchenko*

Abstract

We say that the mutual information of a triple of binary strings a, b, c can be extracted if there exists a string d such that a, b, and c are independent given d, and d is simple conditional to each of the strings a, b, c. This is an analog of the well-known Gács–Körner's definition of extractability of the mutual information for a *pair* of binary strings.

We prove that (in contrast to the case of two strings) there exists a criterion of extractability of the mutual information for a triple a, b, c in terms of complexities involving a, b, c. Roughly speaking, the mutual information between a, b, c can be extracted if and only if the conditional mutual informations $I(a:b|c)$, $I(a:c|b)$, $I(b:c|a)$ are negligible.

Our proof of the main result is based on a non-Shannon-type information inequality, which is a generalization of the recently discovered Zhang–Yeung inequality.

1 Introduction

1.1 Background and Motivation

The problem of extractability of the mutual information for a pair of binary strings is well known in the theory of Kolmogorov complexity. The basic question on extractability of the mutual information can be formulated as follows: let us have a pair of binary strings a, b; can we find a string c such that a and b are independent given c, and c is very simple conditional to a and conditional to b? More exactly, consider the following definition[1].

Definition 1. *We say that the mutual information between strings a, b can be extracted to an accuracy of δ if there exists a binary string c such that*

1. $K(a|c) + K(b|c) \leq K(a, b|c) + \delta$, *i.e., a and b are independent given c, and*
2. $K(c|a) \leq \delta$ *and* $K(c|b) \leq \delta$, *i.e., c is simple given a and given b.*

As usually in the theory of Kolmogorov complexity, we are interested in a 'logarithmic' accuracy, and we shall use definition 1 for $\delta = O(\log K(a, b))$.

It is easy to see that the mutual information between a, b can be extracted (to an accuracy of $O(\log K(a, b)))$ in some trivial cases: if $a = b$, we let $c = a$; if $I(a : b)$ is negligible, we let the string c be empty. But if $0 \ll I(a : b) \ll \min\{K(a), K(b)\}$, the mutual information between a, b can be non-extractable. Namely, the following theorem is known.

*Institute for Information Transmission Problems. Bolshoy Karetny 19, Moscow 101447, Russia. email:anromash@mccme.ru

[1] In terms of Gács and Körner [3] this definition claims that the *common* information for a pair a, b is equal to the *mutual* information between a and b.

Theorem 1. [7] *There exists a* $C_1 > 0$ *such that for any* $C_2 > 0$ *and for any reals* α, β, γ $(0 < \gamma < \min\{\alpha, \beta\})$ *for all large enough* $n > 0$ *there exists a pair of strings* a, b:

1. $|K(a) - \alpha n| \le C_1 \log n$,
2. $|K(b) - \beta n| \le C_1 \log n$,
3. $|I(a : b) - \gamma n| \le C_1 \log n$,

and the mutual information between a *and* b *cannot be extracted to an accuracy of* $C_2 \cdot \log K(a, b)$.

From theorem 1 it follows that for any α, β, γ $(0 < \gamma < \min\{\alpha, \beta\})$ and a large enough n a pair of strings (a, b) exists such that

$$\begin{aligned} K(a) &= \alpha n + O(\log n), \\ K(b) &= \beta n + O(\log n), \\ I(a : b) &= \gamma n + O(\log n), \end{aligned}$$

and the mutual information between a and b cannot be extracted (to an accuracy of $O(\log n)$). The same time it is easy to get another pair (a', b') with the same complexity profile, i.e.,

$$\begin{aligned} K(a') &= \alpha n + O(\log n), \\ K(b') &= \beta n + O(\log n), \\ I(a' : b') &= \gamma n + O(\log n), \end{aligned}$$

such that the mutual information between a' and b' *can* be extracted (to an accuracy of $O(\log n)$). The construction is obvious: we get three strings u, v, w such that

$$|u| = \gamma n, \ |v| = (\alpha - \gamma)n, \ |w| = (\beta - \gamma)n,$$

and $K(u, v, w) = (\alpha + \beta - \gamma)n + O(1)$. Let $a' = \langle u, v\rangle$ and $b' = \langle u, w\rangle$. Then the string u has negligible complexity conditional to a' and conditional to b', and $I(a' : b'|u)$ is $O(\log n)$. Thus for the pair (a', b') the mutual information can be extracted.

So we have two pairs (a', b') and (a, b), which have the same complexities, and the mutual information can be extracted for the first pair but not for the second. This means that the *structure* of a pair (x, y) cannot be determined by its complexity profile $(K(x), K(y), K(x, y))$.

Originally the property of "extractability" of the mutual information and a related notion of *common information* was introduced by Gács and Körner in [3][2]. They proved a result which is close to theorem 1 (Gács and Körner proved an analog of theorem 1 for an accuracy about $\sqrt{n}$). In the form presented above theorem 1 follows from a more strong theorem of An.A. Muchnik [7, theorem 2].

Extractability of the mutual information is interesting as a simplest 'structural' information property that can be defined for a tuple of strings. The property of extractability of the mutual information is known to have several important application, e.g., estimations for the value of *common information* is a standard technique in the theory of multi-source information networks (see a survey in [2]). Another interesting field is investigation of combinatorial meaning of extractability

[2] Gács and Körner defined two variants of the notion 'common information': for Shannon entropy and for Kolmogorov complexity. Here we are interested only in the second (related to Kolmogorov complexity) variant of the definition. To avoid technical difficulties we do not formulate the definition of *common information* in this paper. Instead we talk about *extractability of the mutual information* (definition 1).

of the mutual information [7, 1, 8]. In [6] the property of extractability of the mutual information was employed in the proof of a new information inequality.

In most applications it is important to have a simple and natural criterion of extractability of the mutual information, e.g., a criterion in terms of complexities of the strings involved. As explained above, there is no such a universal criterion for a pair of strings. In contrast to this fact, we prove that a criterion of extractability of the mutual information for a *triple* a, b, c can be formulated in terms of complexities involving a, b, c. Let us note that our method works not only for triples but for any collection of $n \geq 3$ strings.

1.2 Main Results

We consider the property of extractability of the mutual information for a triple of binary string a, b, c. That means we are interested if there exists a string d such that a, b, and c are independent given d, and d is simple conditional to each of the strings a, b, c. Let us formulate this idea more precisely.

Definition 2. *We say that the mutual information of a triple of strings a, b, c can be extracted to an accuracy of δ if a string d exists such that*

1. *$K(a|d) + K(b|d) + K(c|d) \leq K(a, b, c|d) + \delta$, i.e., a, b, and c are independent given d; and*
2. *the values $K(d|a)$, $K(d|b)$, $K(d|c)$ are less than δ, i.e., d is simple conditional to a, b, and c.*

We are interested in definition 2 for $\delta = O(\log K(a, b, c))$. Loosely speaking, we shall prove that the mutual information between a, b, c can be extracted if and only if the values $I(a : b|c)$, $I(a : c|b)$, $I(b : c|a)$ are negligible. Our result is a counterpart of the following simple fact from the theory of Shannon entropy.

Proposition 1. *For a triple of discrete random variables α, β, γ*

$$I(\alpha; \beta|\gamma) = I(\alpha; \gamma|\beta) = I(\beta; \gamma|\alpha) = 0$$

if and only if there exists a random variable ξ such that $H(\xi|\alpha) = H(\xi|\beta) = H(\xi|\gamma) = 0$ and $H(\alpha, \beta, \gamma|\xi) = H(\alpha|\xi) + H(\beta|\xi) + H(\gamma|\xi)$, where $H(\cdot|\cdot)$ and $I(\cdot;\cdot|\cdot)$ are Shannon's conditional entropy and mutual information.

This proposition trivially follows from [2, Exercise 3.4.25]. Let us formulate two theorems, which are analogs of proposition 1 for Kolmogorov complexity (these theorems are the counterparts of 'if' and 'only if' parts of proposition 1 respectively).

Theorem 2. *For any $C_1 > 0$ there exists a $C_2 > 0$ such that if the mutual information of a triple of strings a, b, c can be extracted to an accuracy of $C_1 \cdot \log K(a, b, c)$, then the values $I(a : b|c)$, $I(a : c|b)$, $I(b : c|a)$ are less than $C_2 \cdot \log K(a, b, c)$.*

Theorem 3. *For any $C_1 > 0$ there exists a $C_2 > 0$ such that if for a triple of strings a, b, c the values $I(a : b|c)$, $I(a : c|b)$, $I(b : c|a)$ are less than $C_1 \cdot \log K(a, b, c)$, then the mutual information of a, b, c can be extracted to an accuracy of $C_2 \cdot \log K(a, b, c)$.*

Proof of theorem 2 is straightforward; it is mostly the same as the proof of 'if' part of proposition 1. The second part of the criterion (theorem 3) is quite non-trivial, and its proof is the main goal of this paper. Our proof is quite technical. The plan of the proof is as follows.

- We define *bunch of strings* (definition 3). Roughly speaking, a bunch X is a large (exponential in n, e.g., $|X| \approx 2^{\alpha n}$) collection of strings of length n such that for any $x, y \in X$ the mutual information $I(x:y)$ is large, i.e., not much less then $(1-\alpha)n$.

  Example: lets us have an incompressible string w_0 of length $(1-\alpha)n$ and all strings x_i of length αn. Define X_0 as a set of all concatenations $w_0 x_i$. Then X_0 is a bunch. Note that for most pairs $a, b \in X_0$ we have $I(a:b) \approx (1-\alpha)n$, so the mutual information between a and b can be extracted; really, a 'materialization' of the mutual information between a and b is just a string w_0.

- We prove (theorem 5) that given a bunch X we can find a string w of length αn that is very simple given any $x \in X$. Roughly speaking, all strings in a bunch must have a common part w of length αn, i.e., *any* bunch has approximatly the same structure as the bunch X_0 in the example above. Thus for any bunch X and for most $x, y \in X$ the mutual information between x and y can be extracted.

- Now we explain how to prove theorem 3. To evoid boring details we consider the simplest case. Let us have a triple a, b, c such that

$$\begin{array}{l} K(a) \approx K(b) \approx K(c) \approx n, \\ I(a:b) \approx I(a:c) \approx I(b:c) \approx (1-\alpha)n, \\ I(a:b|c) \approx I(a:c|b) \approx I(b:c|a) \approx 0. \end{array}$$

  Given a triple of strings a, b, c we shall construct (lemma 3) a bunch X that contains the strings a, b, c. Then we get a string w of length $(1-\alpha)n$ that is simple conditional to any $x \in X$. Thus w is simple conditional to a, b, and c. Clearly, a, b, c are independent given this w, and we are done.

  In the general case to prove theorem 3 we need additional technical steps (see lemma 4).

One of the main tools in our proof (steps 2 and 3 in the plan above) is the information inequality from [6]: for some $C > 0$ and for any strings v, w, x, y, z

$$\begin{aligned} I(v:w) \quad \leq \quad & I(v:w|x) + I(v:w|y) + I(x:y) + \\ & I(v:w|z) + I(v:z|w) + I(w:z|v) + C \cdot \log K(v,w,x,y,z). \end{aligned} \tag{1}$$

This is a non-Shannon-type inequality, i.e., it cannot be presented as a combination of Shannon's basic inequalities of type

$$K(a) + K(a,b,c) \leq K(a,b) + K(a,c) + O(\log K(a,b,c)).$$

Inequality (1) is a generalization of the very first example of non-Shannon-type information inequality

$$\begin{aligned} I(v:w) \quad \leq \quad & 2I(v:w|x) + I(v:w|y) + I(x:y) + \\ & I(v:x|w) + I(w:x|v) + O(\log K(v,w,x,y)), \end{aligned} \tag{2}$$

which was found by Z.Zhang and R.W.Yeung [9] [3].

[3]More precisely, Zhang and Yeung proved an analog of (2) for Shannon entropy. But the class of all valid linear inequalities for Kolmogorov complexity coincides with the class of all valid linear inequalities for Shannon entropy [4]. Hence we can formulate Zhang-Yeung inequality as (2). See [6] for a more detailed discussion of non-Shannon-type inequalities for Kolmogorov complexity and Shannon entropy.

1.3 Kolmogorov Type Properties and Relativization

Further we discuss one application of theorem 2. Namely, we prove that some 'Kolmogorov type' properties of a pair of strings (a, b) does not change under relativization with respect to a string c if $I(ab : c)$ is negligible.

Let us remind that the *complexity profile* of strings $x_1, \dots, x_n$ is a sequence of all Kolmogorov complexities involving strings x_i:

$$\overrightarrow{K}(x_1, \dots, x_n) = (K(x_1), \dots, K(x_n), K(x_1x_2), \dots, K(x_1x_2x_3), \dots, K(x_1 \dots x_n)),$$

e.g., $\overrightarrow{K}(a, b) = (K(a), K(b), K(a, b))$. Similarly, the *conditional complexity profile* of strings $x_1, \dots, x_n$ with respect to a string y is a sequence of all Kolmogorov complexities of x_i's given y:

$$\overrightarrow{K}(x_1, \dots, x_n|y) = (K(x_1|y), \dots, K(x_n|y), K(x_1x_2|y), \dots, K(x_1x_2x_3|y), \dots, K(x_1 \dots x_n|y)),$$

e.g., $\overrightarrow{K}(a, b|c) = (K(a|c), K(b|c), K(a, b|c))$.

We use l_∞-norm to measure the difference between complexity profiles. So we shall say that $\overrightarrow{K}(x_1, \dots, x_n)$ is δ-close to $\overrightarrow{K}(x_1, \dots, x_n|y)$ if l_∞-norm of the difference between these vectors is less then δ.

Obviously, if $I(x_1 \dots x_n : y)$ is negligible, then $\overrightarrow{K}(x_1, \dots, x_n)$ is very close to $\overrightarrow{K}(x_1, \dots, x_n|y)$:

Proposition 2. *If $I(x_1 \dots x_n : y) \le \delta$, then $\overrightarrow{K}(x_1, \dots, x_n)$ is $\delta + O(\log N)$-close to $\overrightarrow{K}(x_1, \dots, x_n|y)$, where $N = K(x_1, \dots, x_n, y)$.*

(The proof is straightforward.) Informally proposition 2 claims that relativization with respect to y does not change the simplest 'Kolmogorov type' properties of $x_1, \dots, x_n$ on condition that the mutual information between $x_1, \dots, x_n$ and y is negligible. We believe that the relativization above does not change also more complicated properties of $x_1, \dots, x_n$. For example, using technique from [4] it is not hard to prove the following proposition.

Proposition 3. *Assume $I(x_1 \dots x_n : y) \le \delta$. Then for any z there exists a string z' such that $\overrightarrow{K}(x_1, \dots, x_n, z'|y)$ is $\delta + O(\log N)$-close to $\overrightarrow{K}(x_1, \dots, x_n, z)$, where $N = K(x_1, \dots, x_n, y, z)$.*

(We omit the proof.) We suppose that the following conversion of proposition 3 holds.

Conjecture 1. *Assume $I(x_1 \dots x_n : y) \le \delta$. Then for any z there exists a string z' such that $\overrightarrow{K}(x_1, \dots, x_n, z')$ is $\delta + O(\log N)$-close to $\overrightarrow{K}(x_1, \dots, x_n, z|y)$, where $N = K(x_1, \dots, x_n, y, z)$.*

Using theorem 2 we prove a special case of conjecture 1. Namely, we prove that if the mutual information between a and b can be extracted given c, then the same is true without relativization (assuming $I(ab : c)$ is negligible). More precisely, we have the following theorem:

Theorem 4. *For any $C_1 > 0$ there exists a $C_2 > 0$ such that the following statement holds. Assume that $\langle a, b\rangle$ and c are independent, i.e. $I(ab : c) \le C_1 \cdot \log m$, where $m = K(a, b, c)$. Suppose the mutual information between a and b can be extracted given c to an accuracy of $C_1 \log m$, i.e., there exists a string d such that*

1. *$K(d|ac) \le C_1 \cdot \log m$,*
2. *$K(d|bc) \le C_1 \cdot \log m$,*

3. $K(a|cd) + K(b|cd) \le K(ab|cd) + C_1 \cdot \log m$.

Then the mutual information between a and b can be extracted (without relativization) to an accuracy of $C_2 \log n$.

The rest of the paper is organized as follows. In Section 2 we introduce the basic definitions and notation. In Section 3 we define *bunch of strings* and prove some properties of bunches. In Section 4 we prove the main results (theorem 2 and theorem 3). In Section 4 we prove theorem 4. In Conclusion we formulate a few open problems.

2 Basic Definitions and Notation

For brevity we use the notation

$$\log n := \begin{cases} \log_2 n, & \text{if } n > 0 \\ 1, & \text{if } n = 0. \end{cases}$$

Let us fix any programming language that is optimal by Kolmogorov [5]. Then Kolmogorov complexity of a binary string x (denoted by $K(x)$) is the number of binary digits in the shortest program that prints x on the empty input. Kolmogorov complexity of a binary string x given a binary string y (denoted by $K(x|y)$) is the number of binary digits in the shortest program that prints x on the input y.

Fix any recursive enumeration of all finite sequences of binary strings. Denote by $\langle u_1, \ldots, u_n \rangle$ the binary representation of the integer corresponding to the tuple of binary strings $u_1, \ldots, u_n$. For brevity we use the notation $K(u_1 \ldots u_n | v_1 \ldots v_m)$ instead of $K(\langle u_1, \ldots, u_n \rangle | \langle v_1, \ldots, v_m \rangle)$.

We use the following standard notation:

- denote by $I(x : y) := K(x) + K(y) - K(x, y)$ the mutual information between strings x and y;
- denote by $I(x : y|z) := K(x|z) + K(y|z) - K(x, y|z)$ the mutual information between strings x and y conditional to z.

Let us remind that from the Kolmogorov – Levin theorem [10] it follows that for some $C > 0$ and for any binary strings x, y we have $|K(x, y) - K(x) - K(y|x)| \le C \cdot \log K(x, y)$. Hence,

$$|I(x : y) - (K(x) - K(x|y))| \le C \cdot \log K(x, y).$$

We need also a notation for the mutual information of a triple of binary strings:

$$\begin{aligned} I(x : y : z) \quad := \quad & K(x) + K(y) + K(z) - \\ & -K(x, y) - K(x, z) - K(y, z) + \\ & +K(x, y, z). \end{aligned}$$

From the Kolmogorov – Levin theorem it follows that the mutual information $I(x : y : z)$ is equal to $I(x : y) - I(x : y|z)$ and to $I(x : yz) - I(x : y|z) - I(x : z|y)$ up to an additive term $O(\log K(x, y, z))$.

Besides we use the notation

$$W(x, y, z) := I(x : y|z) + I(x : z|y) + I(y : z|x)$$

introduced by Z.Zhang. Note that if the value $W(x,y,z)$ is negligible (e.g., as in theorem 3), then the differences $|I(x:y:z)-I(x:y)|$ and $|I(x:y:z)-I(xy:z)|$ are also negligible.

Further we define a 'typization' set (typization sets were introduced in [4]; see also use of typization sets in [6]).

Lemma 1. [6] *For some number $C>0$ and for any strings x,y,z there exists a set $Type(x|y,z)$ such that*

$$2^{K(x|yz)-C\log K(x,y,z)} \le |Type(x|y,z)| \le 2^{K(x|yz)},$$

$x \in Type(x|y,z)$, and for all $x' \in Type(x|y,z)$ the following conditions hold:

1. $K(x) - C\cdot\log K(xyz) \le K(x') \le K(x)$,
2. $K(x|yz) - C\cdot\log K(xyz) \le K(x'|yz) \le K(x|yz)$,
3. $K(y|x) - C\cdot\log K(xyz) \le K(y|x') \le K(y|x)$,
4. $K(z|x) - C\cdot\log K(xyz) \le K(z|x') \le K(z|x)$,
5. $K(y|xz) - C\cdot\log K(xyz) \le K(y|x'z) \le K(y|xz)$,
6. $K(z|xy) - C\cdot\log K(xyz) \le K(z|x'y) \le K(z|xy)$,
7. $K(yz|x) - C\cdot\log K(xyz) \le K(yz|x') \le K(yz|x)$.

We shall say that the set $Type(x|y,z)$ from lemma 1 is *the typization set for a string x with respect to y,z.*

For the sake of completeness we prove lemma 1 in Appendix.

3 Bunches of Strings

Definition 3. *A (k,l,n)-***bunch** *is a set of binary strings X such that*

1. $|X| = 2^k$,
2. $K(x_1|x_2) < l$ *for all* $x_1, x_2 \in X$,
3. $K(x) < n$ *for all* $x \in X$.

The definition of a (k,l,n)-bunch is interesting only if $k<l<n+O(1)$. In the sequel we shall see that the most important case is $l = k + O(\log n)$. But first of all we prove a basic lemma which holds for any (k,l,n)-bunch.

Lemma 2. *There exists an algorithm $\mathbb{A}$ that on input $\langle k,l,n\rangle$ enumerates a list of some (k,l,n)-bunches $U_0,\ldots,U_q$ such that*

- *for any (k,l,n)-bunch U there exists a number $i \le q$ such that $|U \cap U_i| \ge 2^{l-\delta}$, where $\delta = 2(l-k)+O(1)$;*
- $q < 2^{n+l-2k+O(1)}$.

Proof. It is easy to construct an algorithm that gets integers k, l, n as an input and prints the list of all (k, l, n)-bunches as an output[4]. Let us call it *the standard enumerator.* This enumeration process can give us much more than $2^{n+l-2k+O(1)}$ bunches. We shall correct it and construct another enumerator.

Suppose s bunches $U_0, \dots, U_{s-1}$ are already printed by the 'corrected' enumerator. Let us explain how to get the next, $(s+1)$-th bunch. First of all we are waiting until the standard enumerator gives us a new (k, l, n)-bunch U. Further we check all the intersections $U \cap U_i$ for $i < s$. If the bunch U has a large intersection with some of the previous bunches U_i, i.e., $|U \cap U_i| \geq 2^{l-\delta}$ for some $i < s$, we drop U and keep waiting for another bunch from the standard enumerator. Otherwise (none of U_i, $i = 0, \dots, (s-1)$ has a large enough intersection with U), we set $U_s = U$ and go to the next iteration.

We have described an algorithm, which prints some bunches $U_0, \dots, U_q$ on the input $\langle k, l, n\rangle$ (we shall call them *standard bunches* for given (k, l, n)). Obviously, for any (k, l, n)-bunch U there exists a standard U_i $(i \leq q)$ such that $|U \cap U_i| \geq 2^{l-\delta}$. It remains to prove that the number of all standard bunches is not too large. More precisely, we must prove that this number is less than $2^{n+l-2k+O(1)}$.

It is enough to prove that any string x belongs to less than $2^{l-k+O(1)}$ standard bunches. Indeed, there are less than 2^n strings x such that $K(x) < n$. If any x belongs to at most $2^{l-k+O(1)}$ standard bunches and every bunch contains 2^k strings, then the number of standard bunches is not larger than

$$\frac{2^n \cdot 2^{l-k+O(1)}}{2^k} = 2^{n+l-2k+O(1)}.$$

Thus it remains to check that that the number of standard bunches covering a fixed string x is less than $2^{l-k+O(1)}$.

Fix an x (such that $K(x) < n$) and assume there are more than 2^{l-k+1} standard bunches U_i such that $x \in U_i$. Let us choose any $N = 2^{l-k+1}$ of these bunches. From the definition of a bunch it follows that all these bunches U_i are included in the set $S = \{y | K(y|x) < l\}$. So we have

$$\left|\bigcup U_i\right| \leq |S| < 2^l$$

(here we take the union of N chosen standard bunches). The size of any U_i is 2^k. Therefore we have

$$N \cdot 2^k - \sum_{i<j} |U_i \cap U_j| \leq \left|\bigcup U_i\right| < 2^l.$$

Further, for any two standard bunches $|U_i \cap U_j| < 2^{l-\delta}$. Hence,

$$N \cdot 2^k - N^2 \cdot 2^{l-\delta} < 2^l,$$

and we get a contradiction provided $\delta > 2(l-k+1)$. This means that less than $2^{l-k+O(1)}$ standard bunches contain the string x, and we are done. □

Below we prove an important property of (k, l, n)-bunches that holds in the case $l \approx k$. Namely, we state that for any (k, l, n)-bunch U there exists a string z such that for each $x \in U$ the string z is very simple conditional to x and $K(x|z)$ is not much larger then l. Loosely speaking, z is a "common part" of all $x \in U$. More precisely, the following theorem holds (it is most interesting in the case $k = l + O(\log n)$).

[4] Of course, there is no way to determine when *all* the list of (k, l, n)-bunches is printed by the algorithm. This enumeration algorithm never stops, though it prints only a finite number of bunches.

Theorem 5. *For any (k,l,n)-bunch U there exist a string z such that $K(x|z) \le l+O(|l-k|+\log n)$ and $K(z|x) = O(|l-k|+\log n)$ for any $x \in U$.*

Proof of theorem. For brevity we shall omit the terms $O(|l-k|+\log n)$ in the proof. We shall write

$$\Phi \lesssim \Psi, \text{ if } \Phi \le \Psi + O(|l-k|+\log n),$$
$$\Phi \gtrsim \Psi, \text{ if } \Phi \ge \Psi - O(|l-k|+\log n),$$
$$\Phi \simeq \Psi, \text{ if } |\Phi - \Psi| = O(|l-k|+\log n)$$

(for any values Φ, Ψ). For example, we shall say that for any $x_1, x_2 \in U$ we have $K(x_1|x_2) \lesssim k$ (instead of $K(x_1|x_2) < l$).

Without loss of generality it can be assumed that $n = \max_{x\in U} K(x)$.

Let us prove that for most of $x \in U$ the complexity $K(x)$ is close to n. Fix any string $x_0 \in U$ such that $K(x_0) = n$. For all $x \in U$ we have $K(x|x_0) \lesssim k$. On the other hand, for the majority of $x \in U$ the complexity $K(x|x_0)$ is close to k because $|U| = 2^k$. Hence for the majority of x complexity $K(x_0, x)$ is about $(n+k)$. At the same time

$$K(x_0, x) = K(x) + K(x_0|x) \lesssim K(x) + k.$$

Therefore for the majority of x we have $K(x) \simeq n$.

Further, let us note that for the majority of $x_1, x_2 \in U$

$$I(x_1 : x_2) \simeq K(x_1) - K(x_1|x_2) \simeq n - k.$$

By lemma 2 we have a standard (k,l,n)-bunch U_j such that $\log|U \cap U_j| \gtrsim k$. Denote by z the binary representation of the integer j. From lemma 2 the number j is not much larger than 2^{n-k}; hence, $K(z) \lesssim n-k$. Now let us fix any string $x \in U$ and prove that $I(x : z) \gtrsim n-k$.

We choose at random a triple x_1, x_2, x_3 from $|U \cap U_j|$ and apply a non-Shannon type inequality (1):

$$\begin{aligned} I(x_1 : x_2) \;\lesssim\; & I(x_1 : x_2|x) + I(x_1 : x_2|z) + I(x : z) + \\ & I(x_1 : x_2|x_3) + I(x_1 : x_3|x_2) + I(x_2 : x_3|x_1). \end{aligned}$$

In the left-hand side of this inequality we have $I(x_1 : x_2) \simeq n-k$ for most of x_1, x_2. In the right-hand side all terms except for $I(x : z)$ are negligible for most of x_1, x_2, x_3. Indeed, consider the value

$$I(x_1 : x_2|x) \simeq K(x_1|x) + K(x_2|x) - K(x_1, x_2|x).$$

We have $K(x_i|x) \lesssim k$ for $i = 1, 2$ because x, x_1, x_2 are elements of a (k,l,n)-bunch U. On the other hand, $K(x_1, x_2|x) \gtrsim 2k$ for most of x_1, x_2 since $|U \cap U_j|$ is not far less than 2^k. Consequently, $I(x_1 : x_2|x) \simeq 0$.

For the same reason the values $I(x_1 : x_2|x_3)$, $I(x_1 : x_3|x_2)$, $I(x_2 : x_3|x_1)$ are negligible for most of x_1, x_2.

Also for most of x_1, x_2 the value $I(x_1 : x_2|z)$ is negligible. Indeed, given z and the numbers k, l, n we can enumerate the list of all strings in U_j. Hence, $K(x_i|z) \lesssim \log|U_j| \simeq k$ for $i = 1, 2$. On the other hand, for most of $x_1, x_2 \in U \cap U_j$ we have $K(x_1, x_2|z) \ge \log|U \cap U_j| \gtrsim 2k$. So we get $I(x_1 : x_2|z) \simeq 0$.

Thus we have $K(z) \lesssim n-k$ and $I(x : z) \gtrsim n-k$. These two conditions imply $K(z|x) \simeq 0$ and $K(x|z) \lesssim k$. $\square$

In the next lemma we prove that for any triple a, b, c the typization set $Type(a|b, c)$ is a bunch.

Lemma 3. *For any strings a, b, c the typization set $Type(a|b,c)$ is a bunch with parameters*

$$(K(a|bc) - O(\log m), K(a|bc) + 2W(a,b,c) + O(\log m), K(a) + 1),$$

where $m = K(a,b,c)$.

Proof. Let us fix three strings a, b, c and consider the typization set $Type(a|b,c)$. From lemma 1 it follows that $|Type(a|b,c)| = 2^{K(a|bc)-O(\log m)}$ and complexity of any $x \in Type(a|b,c)$ is not larger than $K(a)$. Let us estimate the conditional complexity $K(x_1|x_2)$ for any $x_1, x_2 \in Type(a|b,c)$. We apply a non-Shannon type inequality (1):

$$\begin{array}{rcl} I(b:c) & \leq & I(b:c|x_1) + I(b:c|x_2) + I(x_1:x_2) + \\ & & I(b:c|a) + I(a:b|c) + I(a:c|b) + O(\log K(a,b,c,x_1,x_2)). \end{array}$$

Since $x_i \in Type(a|b,c)$, the values $I(b:c|x_i)$ are close to $I(b:c|a)$. Hence we have

$$I(x_1:x_2) \geq I(b:c) - 2 \cdot I(b:c|a) - W(a,b,c) - O(\log m) = I(a:bc) - 2W(a,b,c) - O(\log m).$$

Therefore, $K(x_1|x_2) \leq K(a|bc) + 2W(a,b,c) + O(\log m)$, and we are done. $\square$

4 Extraction of the Mutual Information for a Triple of Strings

At first we prove the following technical result.

Lemma 4. *For any binary strings a, b, c there exists a string d such that*

- $K(d) \leq I(a:bc) + 2W(a,b,c) + O(\log m)$,
- $K(b|d) \leq K(b|a) + 2W(a,b,c) + O(\log m)$,
- $K(c|d) \leq K(c|a) + 2W(a,b,c) + O(\log m)$,

where $m = K(a,b,c)$.

Proof. Let us consider the typization set for a string a with respect to b, c. From lemma 3 it follows that the typization set $Type(a|b,c)$ is a bunch with parameters

$$(K(a|bc) - O(\log m), K(a|bc) + 2W(a,b,c) + O(\log m), K(a) + 1).$$

Further, by lemma 2 we get a *standard* bunch U_j (with the same parameters) such that $|Type(a|b,c) \cap U_j| \geq 2^{K(a|bc)-2W(a,b,c)-O(\log m)}$. Let d be the binary representation of the integer j. By lemma 2 we have $j < 2^{I(a:bc)+2W(a,b,c)+O(\log m)}$; hence, $K(d) < I(a:bc) + 2W(a,b,c) + O(\log m)$.

Let us estimate the conditional complexity $K(b|d)$. Recall that for any $x \in Type(a|b,c) \cap U_j$ we have $K(b|x) = K(b|a) + O(\log m)$ (by definition of $Type(a|b,c)$) and $K(x|d) \leq K(a|bc) + O(\log m)$ (since U_j is enumerable given the number j, and $|U_j| = 2^{K(a|bc)-O(\log m)}$). In other words, b belongs to the set

$$\begin{array}{rcl} B = \{b' & | & \text{there exist at least } 2^{K(a|bc)-2W(a,b,c)-C\log m} \text{ strings } x \text{ such that} \\ & & K(b'|x) \leq K(b|a) + C\log m \text{ and } K(x|d) \leq K(a|bc) + C\log m\}, \end{array}$$

where the constant C does not depend on a, b, c. It is clear that B can be enumerated given d and all the numbers $K(a|bc)$, $K(b|a)$, $W(a,b,c)$, and the value $C\log m$. Therefore, $K(b|d) \leq \log|B| + O(\log m)$. On the other hand, $|B|$ is not larger than

$$\frac{2^{K(a|bc)+O(\log m)} \cdot 2^{K(b|a)+O(\log m)}}{2^{K(a|bc)-2W(a,b,c)-O(\log m)}} = 2^{K(b|a)+2W(a,b,c)+O(\log m)},$$

and we get $K(b|d) \leq K(b|a) + 2W(a,b,c) + O(\log m)$.

By the same argument, $K(c|d) \leq K(c|a) + 2W(a,b,c) + O(\log m)$. □

Now we are ready to prove the main results.

Proof of theorem 2. Denote by m the value $K(a,b,c)$. Let us prove that $I(a:b|c) \leq C \cdot \log m$ for a large enough C (C does not depend on a,b,c).

For any triple a,b,c we have $K(abc|d) \leq K(ab|c) + K(c|d) + O(\log m)$. On the other hand, we are given the inequality

$$K(a|d) + K(b|d) + K(c|d) \leq K(abc|d) + O(\log m).$$

Combining the last two inequalities, we get

$$K(a|d) + K(b|d) \leq K(ab|c) + O(\log m).$$

Since the string d is simple conditional to c, we have $K(a|c) \leq K(a|d) + O(\log m)$ and $K(b|c) \leq K(b|d) + O(\log m)$. Thus we have

$$K(a|c) + K(b|c) \leq K(ab|c) + O(\log m),$$

i.e., $I(a:b|c) = O(\log m)$.

In the same way we get $I(a:c|b) = O(\log m)$ and $I(b:c|a) = O(\log m)$. □

Proof of theorem 3. Denote by m the value $K(a,b,c)$. We must prove that there exists a string d such that

1. the complexities $K(d|a)$, $K(d|b)$, $K(d|c)$ are less than $C \cdot \log m$,
2. $K(a,b,c|d) \leq K(a|d) + K(b|d) + K(c|d) + C \cdot \log m$

for a large enough number C (C does not depend on a,b,c).

From lemma 4 we have a string d such that

$$\begin{aligned} K(d) &\leq I(a:bc) + O(\log m) = I(a:b:c) + O(\log m), && (3) \\ K(b|d) &\leq K(b|a) + O(\log m) = K(b) - I(a:b:c) + O(\log m), && (4) \\ K(c|d) &\leq K(c|a) + O(\log m) = K(c) - I(a:b:c) + O(\log m). && (5) \end{aligned}$$

From (3) and (4) we get $K(d) + K(b|d) = K(b) + O(\log m)$; hence, $K(d|b) = O(\log m)$ and $K(d) = I(a:b:c) - O(\log m)$. From (5) we have $K(d|c) = O(\log m)$.

Let us estimate the conditional complexity $K(d|a)$. We use a simple Shannon-type inequality

$$K(d|a) \leq K(d|b) + K(d|c) + I(b:c|a) + O(\log K(a,b,c,d)) \tag{6}$$

(see, e.g., [4]). Since all values in the right-hand side of (6) are equal to $O(\log m)$, we get $K(d|a) = O(\log m)$.

Now we must prove that a, b, c are independent given d, i.e.,

$$K(a,b,c|d) \leq K(a|d) + K(b|d) + K(c|d) + O(\log m).$$

We have

$$K(a|d) + K(b|d) + K(c|d) = K(a) + K(b) + K(c) - 3I(a:b:c) + O(\log m).$$

On the other hand,

$$\begin{aligned} K(abc) &= K(a) + K(b|a) + K(c|ab) + O(\log m) \\ &= K(a) + K(b|a) + K(c|a) - I(c:b|a) + O(\log m) \\ &= K(a) + K(b) + K(c) - I(a:b) - I(a:c) + O(\log m) \\ &= K(a) + K(b) + K(c) - 2I(a:b:c) + O(\log m). \end{aligned}$$

Since $K(abc|d) = K(abc) - K(d) + O(\log m)$, we get

$$K(abc|d) = K(a|d) + K(b|d) + K(c|d) + O(\log m).$$

This concludes the proof. □

5 Relativized Extraction of the Mutual Information for a Pair of Strings.

In this section we prove theorem 4 (which is a special case of conjecture 1). Our proof is based on theorem 3.

Proof of theorem 4. We must prove that for some d'

$$K(d'|a) = O(\log m),\ K(d'|b) = O(\log m),\ K(a|d') + K(b|d') \leq K(ab|d') + O(\log m).$$

Let us consider the triple $a, b, \langle c, d\rangle$ and check that $W(a, b, \langle c, d\rangle) = O(\log m)$. At first we estimate the value $I(a : b|\langle c, d\rangle)$. Omitting the terms $O(\log m)$ we get

$$\begin{aligned} K(a|\langle c, d\rangle) &= K(a) - I(a : \langle c, d\rangle) \\ &= K(a) - I(a : c) - I(a : d|c) \\ &= K(a) - K(d|c) \\ &= K(a) - I(a : b). \end{aligned}$$

By the same argument we have $K(b|\langle c, d\rangle) = K(b) - I(a : b)$ and $K(ab|\langle c, d\rangle) = K(ab) - I(a : b)$. Hence, $I(a : b|\langle c, d\rangle) = K(a|\langle c, d\rangle) + K(b|\langle c, d\rangle) - K(ab|\langle c, d\rangle) = 0$.

Further we estimate the value $I(a : \langle c, d\rangle|b)$.

$$\begin{aligned} I(a : \langle c, d\rangle|b) &= K(a|b) - K(a|b, \langle c, d\rangle) \\ &= K(a|b) - K(ab|\langle c, d\rangle) + K(b|\langle c, d\rangle) \\ &= K(a|b) - (K(ab) - I(a : b)) + (K(b) - I(a : b)) = 0. \end{aligned}$$

Likewise, $I(b : \langle c, d\rangle|a) = 0$ (all the equalities above are valid up to an additive term $O(\log m)$).

Since $W(a, b, \langle c, d\rangle) = O(\log m)$, from theorem 3 we get a string d' such that the conditional complexities $K(d'|a)$ and $K(d'|b)$ and the mutual information $I(a : b|d')$ are not larger than $O(\log m)$ (the constant in O-term depends on C_1 but not on a, b, c). □

6 Conclusion and Open Problems

In the present paper we proved a criterion of extractability of the mutual information for a triple of strings a, b, c. The technique used in the proof may be interesting on its own right. We suppose that 'bunches of strings' can be applied in other problems of algorithmic information theory. Below we formulate two conjectures, which are generalizations of results proved in section 4 and 5.

We suppose the following refinement of lemma 4 holds:

Conjecture 2. *For any binary strings a, b, c there exists a string d such that*

1. $K(d) \leq I(a : bc) + O(\log m)$,
2. $K(b|d) \leq K(b|a) + O(\log m)$,
3. $K(c|d) \leq K(c|a) + O(\log m)$,
4. $K(bc|d) \leq K(bc|a) + O(\log m)$,

where $m = K(a, b, c)$.

Also we believe that the following generalization of theorem 4 is true:

Conjecture 3. *For any $C_1 > 0$ there exists a $C_2 > 0$ such that the following condition holds. Assume that for some strings a, b, c the pair $\langle a, b\rangle$ and c are independent, i.e., $I(ab : c) \leq C_1 \cdot \log m$, where $m = K(a, b, c)$. Suppose γ bits of the mutual information between a and b can be extracted (to an accuracy of $C_1 \log m$) given c, i.e., there exists a string d such that*

$$K(d|c) \geq \gamma,\ K(d|bc) \leq C_1 \cdot \log m,\ K(d|ac) \leq C_1 \cdot \log m.$$

Then γ bits of the mutual information between a and b can be extracted (without relativization) to an accuracy of $C_2 \log m$, i.e., there exists a string d' such that

$$K(d') \geq \gamma,\ K(d'|b) \leq C_2 \cdot \log m,\ K(d'|a) \leq C_2 \cdot \log m.$$

Remark 1. Obviously, conjecture 3 is a special case of conjecture 1. It can be shown that conjecture 2 implies conjecture 3 (in the same way as lemma 4 implies theorem 4). We do not know if conjecture 2 implies conjecture 1.

Remark 2. It was proved in [6] that conjecture 2 implies inequality (1). Other generalizations of lemma 4 may imply new non-Shannon-type information inequalities.

Acknowledgment

I would like to thank Andrei Muchnik for very helpful discussions on the topic of this paper.

References

[1] A. Chernov, An. A. Muchnik, A. Shen, A. Romashchenko, and N. Vereshchagin. Upper semi-lattice of binary strings with the relation x is simple conditional to y. *Theoretical Computer Science*, 271:69–95, 2002.

[2] I. Csiszár and J. Körner. *Information Theory. Coding Theorems for Discrete Memoryless Systems.* Akadémiai Kiadó, Budapest, Hungary, 1981.

[3] P. Gács and J. Körner. Common information is far less then mutual information. *Probl. Contr. Inform. Theory*, 2(2):149–162, 1973.

[4] D. Hammer, A. Romashchenko, A. Shen, and N. Vereshchagin. Inequalities for Shannon entropy and Kolmogorov complexity. *Journal of Computer and System Sciences*, 60:442–464, 2000.

[5] A. N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, 1(1):1–7, 1965.

[6] K. Makarychev, Yu. Makarychev, A. Romashchenko, and N. Vereshchagin. A new class of non-Shannon type inequalities for entropies. *Communications in Information and Systems*, 2(2):147–166, 2002.

[7] An. A. Muchnik. On common information. *Theoretical Computer Science*, 207:319–328, 1998.

[8] A. E. Romashchenko. Pairs of words with nonmaterializable mutual information. *Problems of Information Transmission*, 36(1):1–18, 2000.

[9] Z. Zhang and R. W. Yeung. On characterization of entropy function via information inequalities. *IEEE Transactions on Information Theory*, 44:1440–1452, 1998.

[10] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the algorithmic concepts of information and randomness. *Russian Mathematical Survey*, 6(25):83–124, 1970.

7 Appendix

Proof of lemma 1. Let us fix three binary strings x, y, z and consider the set S of strings which are "similar" to x given y and z in the following sense. We define S as the set of all strings x' such that

1. $K(x') \leq K(x)$,
2. $K(x'|yz) \leq K(x|yz)$,
3. $K(y|x') \leq K(y|x)$,
4. $K(z|x') \leq K(z|x)$,
5. $K(y|x'z) \leq K(y|xz)$,
6. $K(z|x'y) \leq K(z|xy)$,
7. $K(yz|x') \leq K(yz|x)$.

Obviously, $x \in S$. Further, the set S can be enumerated given the strings y, z and all the numbers $K(x), K(x|yz), K(y|x), K(z|x), \ldots$ involved in the definition of S. Hence,

$$K(x|yz) \leq \log|S| + O(\log K(x, y, z)).$$

This implies that $|S|$ is quite large: $|S| \geq 2^{K(x|yz) - O(\log K(xyz))}$.

By the construction of S for any $x' \in S$ we have $K(x'|yz) \leq K(x|yz)$. On the other hand, $|S|$ is about $2^{K(x|yz)}$. Hence for the majority of $x' \in S$ the conditional complexity $K(x'|yz)$ is about $K(x|yz)$ (up to $O(\log K(xyz))$).

Therefore for a large enough constant C_1 the set

$$T = \{x' \in S | K(x'|yz) > K(x|yz) - C_1 \log K(xyz)\}$$

contains most of strings from S.

Let $Type(x|y, z)$ be the set T defined above. It remains to prove that for every $x' \in T$ the conditional complexities involving x', y, z are the same (up to logarithmic terms) as the corresponding conditional complexities for x, y, z, i.e.,

1. $K(x') \geq K(x) - C_2 \cdot \log K(xyz)$,
2. $K(y|x') \geq K(y|x) - C_2 \cdot \log K(xyz)$,
3. $K(z|x') \geq K(z|x) - C_2 \cdot \log K(xyz)$,
4. $K(y|x'z) \geq K(y|xz) - C_2 \cdot \log K(xyz)$,
5. $K(z|x'y) \geq K(z|xy) - C_2 \cdot \log K(xyz)$,
6. $K(yz|x') \geq K(yz|x) - C_2 \cdot \log K(xyz)$

for a large enough $C_2 > 0$.

For example, let us prove that $K(y|x') = K(y|x) - O(\log K(xyz))$. Note that

$$K(x'yz) \leq K(x')+K(y|x')+K(z|yx')+O(\log K(xyz)) \leq K(x)+K(y|x')+K(z|yx)+O(\log K(xyz)).$$

On the other hand,

$$K(xyz) = K(x) + K(y|x) + K(z|yx) + O(\log K(xyz)).$$

Thus, if $K(y|x') \ll K(y|x)$, we have $K(x'yz) \ll K(xyz)$. But this implies

$$K(x'|yz) = K(x'yz) - K(yz) + O(\log K(xyz)) \ll K(xyz) - K(yz) = K(x|yz) + O(\log K(xyz)),$$

and we get a contradiction with the definition of T. This yields that

$$K(y|x') \geq K(y|x) - C_2 \cdot \log K(xyz)$$

for a large enough C_2.

By the same argument the values $K(x')$, $K(z|x')$, $K(y|x'z)$, $K(z|x'y)$, and $K(yz|x')$ are equal (up to $O(\log K(xyz))$) to the complexities $K(x)$, $K(z|x)$, $K(y|xz)$, $K(z|xy)$, and $K(yz|x)$ respectively. $\square$