# Monotone complexity of a pair

Pavel Karpovich*

August 5, 2009

**Abstract**

We define monotone complexity $KM(x,y)$ of a pair of binary strings $x,y$ in a natural way and show that $KM(x,y)$ may exceed the sum of the lengths of $x$ and $y$ (and therefore the a priori complexity of a pair) by $\alpha\log(|x|+|y|)$ for every $\alpha<1$ (but not for $\alpha>1$).

We also show that decision complexity of a pair or triple of strings does not exceed the sum of its lengths.

# 1 Introduction

There are different versions of Kolmogorov complexity: plain complexity ($C$), prefix complexity ($K$), decision complexity ($KR$), monotone complexity ($KM$), etc. Let us recall the definitions of plain, monotone and decision complexities in a form suitable for generalizations (see [7, 8]).

## 1.1 Plain complexity

Kolmogorov complexity $C_F(x)$ of a binary string $x$ with respect to a computable function $F$ (a decompressor) is defined as

$$C_F(x)=\min\{|p|:F(p)=x\},$$

where $|p|$ stands for the length of a binary string $p$. There exists an optimal decompressor $U$ such that $C_U$ is minimal up to $O(1)$; $C_U(x)$ is then called (plain) Kolmogorov complexity of $x$.

Let us reformulate this definition in a way that is parallel to the definition of monotone complexity. Instead of a function $F$ let us consider its graph. A *description mode* is an enumerable set $W$ of pairs of binary strings that is a graph of a function, i.e.,

$$\langle p,x\rangle\in W,\ \langle p',x'\rangle\in W,\ p=p'\ \Rightarrow\ x=x',$$

If $\langle p,x\rangle\in W$, then $p$ is called a *description for $x$ with respect to $W$*. The complexity $C_W(x)$ of a binary string $x$ is the minimal length of a description for $x$ with respect to $W$. There is an optimal description mode $S$ such that for every description mode $W$ there exists $c_W$ such that

$$C_S(x)\leqslant C_W(x)+c_W$$

for every binary string $x$. The corresponding function $C_S$ is plain Kolmogorov complexity.

*Moscow State Lomonosov University, pkarpovich@mail.ru. The work was performed while visiting LIF Marseille (CNRS & Univ. Aix–Marseille); the visit was made possible by the CNRS France–Russia exchange program; preparation of the final text was supported also by NAFIT ANR 008-01 grant.

## 1.2 Monotone complexity

We use the definition of monotone complexity $KM(x)$ suggested by L. A. Levin. (Levin [2] gave a criterion of Martin-Löf randomness in its terms: a binary sequence $\omega$ is Martin-Löf random if and only if $|x| - KM(x) \leqslant c$ for some constant $c$ and all prefixes $x$ of sequence $\omega$; here $|x|$ denotes the length of a string $x$. Earlier a similar criterion was proven by Schnorr who used a different version of complexity, called "process complexity".) Let us recall the definition of monotone complexity in terms of binary relations. A *monotone description mode* is an enumerable set $W$ of pairs of binary strings such that:

- if $\langle p,x\rangle \in W$ and $p \preceq p'$, then $\langle p',x\rangle \in W$.
- if $\langle p,x\rangle \in W$ and $x' \preceq x$, then $\langle p,x'\rangle \in W$.
- if $\langle p,x\rangle \in W$ and $\langle p,x'\rangle \in W$, then $x \preceq x'$ or $x' \preceq x$.

Here $x \preceq x'$ means that $x$ is a prefix of $x'$ (or $x = x'$). The intuition behind this definition: a binary string $u$ is considered as partial information about an infinite sequence that has prefix $u$; then $p \preceq p'$ means that $p'$ is a refinement of $p$, so if $p$ describes $x$, every $p' \succeq p$ should also describe $x$, and so on.

If $\langle p,x\rangle \in W$, then $p$ is called a *description for $x$ with respect to $W$*. The monotone complexity $KM_W(x)$ of $x$ with respect to a monotone description mode $W$ is (again) the minimal length of a description for $x$. There is an optimal monotone description mode $S$ such that

$$KM_S(x) \leqslant KM_W(x) + c_W$$

for every monotone description mode $W$ and binary string $x$. The function $KM_S$ is called *monotone Kolmogorov complexity*. (It is indeed monotone: if $x$ is a prefix of $x'$, then $KM(x) \leqslant KM(x')$.)

## 1.3 Decision complexity

Decision complexity was defined by D.W. Loveland [4]. As before, we reformulate the definition in terms of binary relations. (Here description is treated as an isolated binary string while described object is treated as information about an infinite sequence.)

Formally, a *decision description mode* is an enumerable set $W$ of pairs of binary strings such that:

- if $\langle p,x\rangle \in W$ and $x' \preceq x$, then $\langle p,x'\rangle \in W$.
- if $\langle p,x\rangle \in W$ and $\langle p,x'\rangle \in W$, then $x \preceq x'$ or $x' \preceq x$.

If $\langle p,x\rangle \in W$, then $p$ is called a *description for $x$ with respect to $W$*. The decision complexity $KR_W(x)$ of $x$ is the minimal length of a description for $x$ with respect to $W$. There is an optimal decision description mode $S$ such that

$$KR_S(x) \leqslant KR_W(x) + c_W$$

for every decision description mode $W$ and binary string $x$. $KR_S(x)$ is called *decision Kolmogorov complexity*.

The notions of monotone complexity and decision complexity can be naturally generalized to tuples of strings. (Monotone complexity for tuples was considered also by H. Takahashi, cf. [6].)

## 2 Monotone complexity of a pair.

A *monotone description mode for pairs* is a pair of enumerable sets $W_1$ and $W_2$; each of them is a monotone description mode (as defined earlier).

The monotone complexity $KM_{W_1,W_2}(x,y)$ of a pair of binary strings $x$ and $y$ is the minimal length of a string $p$ such that $\langle p,x\rangle \in W_1$ and $\langle p,y\rangle \in W_2$ (i.e., $p$ describes $x$ with respect to $W_1$ and describes $y$ with respect to $W_2$). There is an optimal monotone description mode for pairs and we can define monotone complexity of a pair, denoted by $KM(x,y)$.

Monotone complexity of pairs is a monotone function: $x \preceq x'$ and $y \preceq y'$ implies $KM(x,y) \leqslant KM(x',y')$. Monotone complexity of pairs $\langle x,x\rangle$, $\langle x,\Lambda\rangle$ and $\langle \Lambda,x\rangle$ (here $\Lambda$ stands for an empty string) equals $KM(x)$ (up to $O(1)$ additive term).

Monotone complexity of a string $x$ is bounded by its length:

$$KM(x) \leqslant |x| + c$$

(for some $c$ and all $x$). It is easy to prove that monotone complexity of a pair $\langle x,y\rangle$ is bounded by sum of lengths of strings $x$ and $y$ with additional logarithmic term. For every $\alpha > 1$ we have

$$KM(x,y) \leqslant |x| + |y| + \alpha \log(|x| + |y|) + O(1).$$

(all the logarithms have base 2). Indeed, a pair $\langle x,y\rangle$ can be (monotonically) described by the concatenation of a self-delimited code for $x$ (of size $|x| + \alpha \log |x|$) and string $y$. The following theorem shows that this bound cannot be significantly improved.

**Theorem 1**. *For every $\alpha < 1$ and every $c \in \mathbb{N}$ there exists a pair of binary strings $\langle x,y\rangle$ such that*

$$KM(x,y) > |x| + |y| + \alpha \log(|x| + |y|) + c.$$

**Proof**. We fix some universal monotone description mode $W$ of pairs. By way of contradiction, let us suppose the inequality

$$KM(x,y) \leqslant |x| + |y| + \alpha \log(|x| + |y|) + c$$

holds for some $\alpha < 1$, some $c \in \mathbb{N}$ and for all pairs $\langle x,y\rangle$. Then every pair $\langle x,y\rangle$ has description of length $f(|x| + |y|)$ where

$$f(n) = n + \lfloor \alpha \log n \rfloor + c$$

(and $f(0) = c$). (Note that if $p$ is a description for a string $x$, then every $p' \succeq p$ is also description for $x$).

We get the desired contradiction by counting how many objects can a description serve and how many descriptions and objects we have. First of all, note that we have about $n2^n$ pairs where sum of lengths is $n$ but only $2^n$ descriptions of length $n$. This is not enough for us, because the same string can be a description of many pairs: if $p$ is a description of some pair $\langle x,y\rangle$ with long $x$ and $y$, it is a description of all pairs $\langle x',y'\rangle$ where $x' \preceq x$ and $y' \preceq y$, and $n+1$ pairs among them have $|x'| + |y'| = n$. So we get the same factor $n$ here as before. The crucial observation is that if some short $p$ is a description of a pair $\langle x,y\rangle$ with long $x$ and $y$, then all extensions of $p$ describe the same pair and therefore we waste a lot of descriptions. To make this argument formal, we need to consider at the same time descriptions of different lengths.

It is done in the following way. Let $S$ be a set of binary strings. We define the *gain* of the set $S$, denoted by $G(S)$, as follows: each pair $\langle x,y\rangle$ that has a description $p$ in $S$ with $|p| = f(|x|+|y|)$, adds $2^{-(|x|+|y|)}$ to the gain.

$$G(S) = \sum_{\langle x,y\rangle \text{ has a description } p \text{ in } S \text{ with } |p| = f(|x|+|y|)} 2^{-(|x|+|y|)}.$$

Let $S_n$ be a set of all strings of length at most $f(n)$. By assumption, $S_n$ contains descriptions of length $f(k)$ for all pairs $\langle x,y\rangle$ such that $k = |x|+|y| \leqslant n$. Therefore the gain of $S_n$ is at least

$$\sum_{k\leqslant n}(k+1) \simeq n^2/2$$

At the other hand, we prove the following lemma (and get the desired contradiction):

**Lemma**. *The gain of the set of all strings of length at most $f(n)$ does not exceed $O(n^{1+\alpha})$.*

**Proof.** Let $p$ be a string of length $f(l)$ for some $l \leqslant n$. We prove the following upper bound on the gain of set $S_{p,n}$ of all binary strings of length at most $f(n)$ that have prefix $p$:

$$2^{f(l)}G(S_{p,n}) \leqslant (n+1)2^{f(n)-n} + \sum_{k=l}^{n-1} 2^{f(k)-k+1} + \sum_{k=\lceil (n-1)/2\rceil}^{n-1} (2k+1-n)2^{f(k)-n} \qquad (1)$$

(Note that in the last term the factor $(2k+1-n)$ is non-negative if and only if $k \geqslant \lceil (n-1)/2\rceil$.) Using the fact that $f(k)$ is less than $k+\alpha\log(k)+c$ we get an upper bound for the gain of $S_{b,n}$ when $|b| = c$ (we let $l = 0$):

$$2^c G(S_{b,n}) \leqslant 2^c(n+1)n^\alpha + 2^c\cdot 2\cdot \sum_{k=0}^{n-1} k^\alpha + 2^c\cdot 2\cdot \sum_{k=\lceil (n-1)/2\rceil}^{n-1} (2k+1-n)2^{k-n}k^\alpha$$

The left-hand side is an upper bound for $G(S_n)$ since the gain is achieved on strings of size $b$ or more, and all three terms in the right hand side (both sums and the additional term $2^c(n+1)n^\alpha$) are bounded by $O(n^{1+\alpha})$ values.

It remains to prove the inequality (1) by a backward induction on the length of string $p$. There are $2^{f(k)}$ different subsets $S_{p,n}$ with $|p| = f(k)$, and our bound is valid for each of them. The right side of the inequality (1) depends only on the length of $p$.

**Induction base** ($l = n$). For strings $p$ with $|p| = f(n)$ we need to show that the following inequality holds:

$$G(S_{p,n}) \leqslant (n+1)2^{-n}$$

Indeed, the set $S_{p,n}$ consists of only one string of length $f(n)$ that can be a description for $n+1$ (or less) pairs $\langle x,y\rangle$ with $|x|+|y| = n$, and the pairs with smaller sum of lengths do not give any gain.

**Induction step.** Suppose the inequality (1) is valid for all sets $S_{p',n}$ with $|p'| = f(l)$ and $l > k$. We will prove the bound on the gain $G(S_{p,n})$ with $|p| = f(k)$. At first we consider a case when $f(k+1) = f(k)+1$. (It can also happen that $f(k+1) = f(k)+2$, but we consider this case later.) The set $S_{p,n}$ consists of the root $p$ and two subtrees $S_{p0,n}$ and $S_{p1,n}$. A simple bound for $G(S_{p,n})$ is the sum of gains of this subtrees and the root's gain, but it is not enough. We should use the fact that if the root ($p$) is a description for many pairs of total length $k$ then there

should be of lot of pairs (of greater total length) that have descriptions in both subtrees $S_{p0,n}$ and $S_{p1,n}$, and we should take into account descriptions for each of this pairs only once.

There is some maximal pair of binary strings $\langle x,y\rangle$ such that $p$ is a description of $\langle x,y\rangle$ (in the following sense: if $p$ is a description of some other pair $\langle x',y'\rangle$, then $x' \preceq x$ and $y' \preceq y$). The total length $|x|+|y|$ of this pair may be greater than $k$; in this case $p$ provides gain for several pairs. Let $r$ be a number of those pairs. (Obviously, $0 \leqslant r \leqslant k+1$). Then (by definition) the root $p$ itself provides gain $r2^{-k}$. If $r > 1$, the root $p$ is also a description for at least $r-1$ pairs with total length $k+1$. These pairs are already taken into account in both $G(S_{p0,n})$ and $G(S_{p1,n})$ since they have descriptions of length $f(k+1)$ in these subtrees. Therefore, we may subtract this overlap of size $(r-1)2^{-(k+1)}$ from the sum $G(S_{p0,n})+G(S_{p1,n})$. Continuing this line of reasoning, we note that the root $p$ is a description for $r-2$ pairs with total length $k+2$, for $r-3$ pairs with total length $k+3$ and so on. We should take into account the overlap for these pairs too. Gains and penalties should be taken only for pairs with total length at most $n$. Thus we get the following bound on the gain of $S_{p,n}$:

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+r2^{-k}-(r-1)2^{-(k+1)}-(r-2)2^{-(k+2)}-\ldots-(r-i)2^{-(k+i)}.$$

Here $i$ is the maximal integer such that $r-i \geqslant 1$ and $k+i \leqslant n$, i.e., $i=\min(r-1,n-k)$. Transforming the right hand side (splitting one term into two), we get

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2^{-k}+(r-1)2^{-k}-(r-1)2^{-(k+1)}- \\ -(r-2)2^{-(k+2)}-\ldots-(r-i)2^{-(k+i)},$$

then (combining terms that contain $r-1$)

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2^{-k}+(r-1)2^{-(k+1)}- \\ -(r-2)2^{-(k+2)}-\ldots-(r-i)2^{-(k+i)},$$

then (splitting again)

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2^{-k}+2^{-(k+1)}+(r-2)2^{-(k+1)}- \\ -(r-2)2^{-(k+2)}-\ldots-(r-i)2^{-(k+i)},$$

then (combining terms that contain $r-2$)

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2^{-k}+2^{-(k+1)}+(r-2)2^{-(k+2)}-\ldots-(r-i)2^{-(k+i)},$$

and so on until we get

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2^{-k}+2^{-(k+1)}+\ldots+2^{-(k+i-1)}+(r-i)2^{-(k+i)},$$

Recall that we have two cases: $i=\min(r-1,n-k)$ and minimum can be equal to the first or the second term. If $r-1<n-k$, the first term matters. Then $i=r-1$, so $r-i=1$ and in the right hand side we have a geometric sequence that can be bounded by twice its first term, i.e.,

$$G(S_{p,n}) \leqslant G(S_{p0,n})+G(S_{p1,n})+2\cdot 2^{-k}.$$

If $r-1 \geqslant n-k$, then $i = n-k$ and (in addition to geometric sequence) we have the last term:

$$G(S_{p,n}) \leqslant G(S_{p0,n}) + G(S_{p1,n}) + 2 \cdot 2^{-k} + (r-n+k)2^{-n}.$$

The maximal value for this last term is achieved when $r$ is maximal, i.e., $r = k+1$. So in any case we have the following bound:

$$G(S_{p,n}) \leqslant G(S_{p0,n}) + G(S_{p1,n}) + 2 \cdot 2^{-k} + (2k+1-n)2^{-n}.$$

At the end we have the expression:

$$G(S_p) < 2\max(G(S_{p0}), G(S_{p1})) + 2^{-k+1} + \max(0, 2k+1-n)2^{-n}$$

Remember, $f(k+1) - f(k)$ is equal to 1 by assumption. Then the inequality may be rewritten as:

$$G(S_p) < 2^{f(k+1)-f(k)}\max(G(S_{p0}), G(S_{p1})) + 2^{-k+1} + \max(0, 2k+1-n)2^{-n} \qquad (2)$$

Now we can multiply both sides of inequality (2) by $2^{f(k)}$ and use the induction assumption. The max-operation in the last terms restricts the sum to its non-negative terms, i.e., for $k \geqslant \lceil (n-1)/2 \rceil$.

Now it remains to consider the case $f(k+1) - f(k) = 2$. In this case we can also use the inequality (2) with term $\max(G(S_{p00}), G(S_{p10}), G(S_{p01}), G(S_{p11}))$ instead of $\max(G(S_{p0}), G(S_{p1}))$. (We have the sum of gains for four subtrees, and $2^{f(k+1)-f(k)} = 4$. Note that we do not use the overlap in full: the same pairs are served in all four subtrees, so we could subtract three times more, but it is not needed.) This is enough for our induction argument.

Lemma (and therefore Theorem 1) are proven.

# 3 Decision complexity of triples.

A *decision description mode for pairs* is a pair of enumerable sets $W_1$ and $W_2$; each of them is a decision description mode. The complexity $KR_{W_1,W_2}(x,y)$ of a pair of binary strings $x$ and $y$ is the minimal length of a string $p$ such that $\langle p,x\rangle \in W_1$ and $\langle p,y\rangle \in W_2$ (i.e., $p$ describes $x$ with respect to $W_1$ and $p$ describes $y$ with respect to $W_2$).

There is an optimal decision description mode for pairs and we can define decision complexity of a pair, denoted by $KR(x,y)$. We can also define decision complexity $KR(x,y,z)$ of a triple in the same way (as well as decision complexity of $k$-tuples for any fixed $k$).

It is easy to see that decision complexity of a binary string $x$ is bounded by $|x| + O(1)$. Indeed, we may consider the set $W$ of all pairs $\langle p,x\rangle$ where $x$ is a prefix of $p$. In this case every string $x$ is a description of itself; switching to the optimal description mode, we lose $O(1)$.

It is also easy to prove that decision complexity of a pair $\langle x,y\rangle$ is bounded by $|x|+|y|+O(1)$. Let the set $W_1$ from our definition be the set of all pairs $\langle p,x\rangle$ where $x$ is a prefix of $p$. Let the set $W_2$be the set of all pairs $\langle p,y\rangle$ where $y$ is a prefix of $p^R$ (the reversed string $p$). Then any pair $\langle x,y\rangle$ has a description $xy^R$, and its length is $|x|+|y|$.

It turns out (quite unexpectedly) that similar statement is true for triples (though we do not know the simple explanation why it happens and the argument we use looks artificial; we do not know whether it generalizes to $k$-tuples for $k > 3$):

**Theorem 2**.

$$KR(x,y,z) \leqslant |x| + |y| + |z| + c$$

*for some $c$ and all triples $\langle x,y,z\rangle$.*

**Proof**. The statement of Theorem 2 is a consequence of the following combinatorial statement:

**Lemma 1.** *For every $n \in \mathbb{N}$ there is a set $Z_n$ that contains $2^n$ triples of $n$-bit strings $\langle a,b,c\rangle$ such that for every triple $\langle x,y,z\rangle$ with $|x|+|y|+|z| = n$ there exists a triple $\langle a,b,c\rangle \in Z_n$ such that $x \preceq a$, $y \preceq b$ and $z \preceq c$.*

Note that we need at least $2^n$ elements in $Z_n$ since every triple $\langle a,b,c\rangle$ serves $\binom{n+2}{2}$ triples $\langle x,y,z\rangle$ and there are $2^n\binom{n+2}{2}$ triples $\langle x,y,z\rangle$ with $|x|+|y|+|z| = n$.

Lemma 1 implies the statement of Theorem 2. Indeed, we may assume that $Z_n$ is a computable function of $n$, and fix some bijection between elements of $Z_n$ and $n$-bit strings. Then a binary string $p$ of length $n$ that corresponds to a triple $\langle a,b,c\rangle \in Z_n$ is considered as a description for every triple $\langle x,y,z\rangle$ where $x \preceq a$, $y \prec b$, $z \preceq c$. It remains to prove Lemma 1.

Lemma 1 is a simple consequence of the following algebraic statement. Consider for every $n$ a $n$-dimensional vector space $\mathbb{F}_2^n$ over two-element field $\mathbb{F}_2$.

**Lemma 2.** *There is a family of $3n$ vectors $a_1,\ldots,a_n,b_1,\ldots,b_n,c_1,\ldots,c_n$ in this space such that for every non-negative $q,r,t$ with $q+r+t = n$ the vectors $a_1,\ldots,a_q,b_1,\ldots,b_r,c_1,\ldots,c_t$ are linearly independent.*

In other terms, we have three bases $(a_i)$, $(b_i)$ and $(c_i)$ in our space $\mathbb{F}_2^n$ with additional property: if we take in total $n$ vectors from these bases, and in each basis start from the beginning, we again get a basis in $\mathbb{F}_2^n$.

Let us show how Lemma 2 implies Lemma 1 (and therefore the statement of Theorem 2). There are $2^n$ different linear functions on $\mathbb{F}_2^n \to \mathbb{F}_2$. For each linear function $f$ we construct a triple of binary strings $\langle a,b,c\rangle$ (and these triples form $Z_n$):

$$a = f(a_1)\ldots f(a_n), \quad b = f(b_1)\ldots f(b_n), \quad c = f(c_1)\ldots f(c_n).$$

So we get $2^n$ triples. For any triple of binary strings $\langle x,y,z\rangle$ such that $|x|+|y|+|z| = n$, consider $q = |x|$, $r = |y|$, and $t = |z|$. Since $a_1,\ldots,a_q,b_1,\ldots,b_r,c_1,\ldots,c_t$ are independent, there exists a linear function that has values $x_1,\ldots,x_q$ on $a_1,\ldots,a_1$, has values $y_1,\ldots,y_r$ on $b_1,\ldots,b_r$ and $z_1,\ldots,z_t$ on $c_1,\ldots,c_t$. It remains to prove Lemma 2.

**Proof of lemma 2.** We will construct the required family by induction over $n$. The induction step moves us from $n$ to $n+3$, so we need to consider the cases $n = 1,2,3$ for a base.

**Induction base**. For $n = 1$, the space has dimension 1 and three vectors are $a,a,a$ where $a$ is the only nonzero vector in the space.

For $n = 2$ consider the basis $e,f$ in our (2-dimensional) space; six vectors could be, for example

$$\begin{array}{cc} e & f \\ f & e \\ e+f & e \end{array}$$

(the first row is $a_1,a_2$, the second is $b_1,b_2$, the third is $c_1,c_2$). Each row is evidently a basis, and if we take any two vectors from the first column, we also get a basis.

Finally, for $n = 3$ we take a basis $e,f,g$ in three-dimensional space and consider vectors

$$\begin{array}{ccc} e & f & g \\ g & f & e \\ f+e+g & f & e. \end{array}$$

Each row is evidently a basis; first column is a basis. If we take two first vectors of any row and complement them by a first element of some other row, we again get a basis. (Note that we can check either the linear independence or the fact that chosen vectors span the entire space.)

**Induction step**. By induction assumption, we have $3k$ vectors

$$\begin{array}{cccc} a_1 & a_2 & \ldots & a_k \\ b_1 & b_2 & \ldots & b_k \\ c_1 & c_2 & \ldots & c_k \end{array}$$

in a $k$-dimensional space. Now we add three new dimensions and corresponding vectors $a, b, c$ (that complement any basis in $k$-dimensional space giving a basis in $(k+3)$-dimensional space). We need to construct $3k+3$ vectors in this extended space. It can be done as follows:

$$\begin{array}{cccccc} a & a_1[+c] & a_2[+c] & \ldots & a_k[+c] & b+c \quad c \\ b & b_1[+a] & b_2[+a] & \ldots & b_k[+a] & c+a \quad a \\ c & c_1[+b] & c_2[+b] & \ldots & c_k[+b] & a+b \quad b \end{array}$$

(square brackets mean that we either add the term in brackets or not, the choice will be made later).

We need to check the every family of $k+3$ vectors (in each row we choose some vectors starting from the left) is independent. Let us start with simple cases where this can be checked independently of the terms in brackets.

Each row forms a basis: first vector and two last vectors generate all three vectors $a, b, c$, after that the square brackets terms do not matter and we use the induction assumption.

If selection involves all three rows, then vectors $a$, $b$, and $c$ are there, and the rest of the selection is $k$ vectors taken from old family, so we get a basis (induction assumption).

It remains to consider the case when selection involves exactly two rows, say, two first rows. Then it includes vectors $a$ and $b$. Therefore, the terms $[+a]$ in the second row do not matter (since we can add $b$ without changing linear independence). There are several possibilities starting from

$$a, b, b_1, b_2, \ldots, b_k, a+c$$

(all the terms except the first one are taken from the second row) and ending with

$$a, a_1[+c], a_2[+c], \ldots, a_k[+c], b+c, b$$

(all the terms, except the last one, are taken from the first row). These two extreme cases are easy (we have $a, b, c$ and vectors from the old basis), but intermediate cases require more attention. Let us start with selection

$$a, a_1[+c], b, b_1, b_2, \ldots, b_k$$

(two vectors from the first row and the rest from the second row). We have here $b_1, \ldots, b_k$ that form a basis in the old space; vector $a_1$ is a combination of them, so if we add $c$, we get a basis in the new space (all three new vectors $a, b, c$ are now accessible). Then we move to the selection

$$a, a_1 + c, a_2[+c], b, b_1, \ldots, b_{k-1}.$$

Here (by induction) the vectors $a_1, b_1, \ldots, b_{k-1}$ form a basis, therefore $a_2$ is a combination of them. Using $a_1 + c$ instead of $a_1$ in this combination, we may get $a_2 + c$ instead of $a_2$. If this

is the case, we do not add $c$ to $a_2$ and get a basis in the new space; if we still get $a_2$, not $a_2+c$ (this happens if $a_1$ was not involved in the expression for $a_2$), we use $a_2+c$. Then we consider the next selection

$$a, a_1+c, a_2[+c], a_3[+c], b, b_1, b_2, \ldots, b_{k-2},$$

recall that $a_1, a_2, b_1, \ldots, b_{k-2}$ form a basis, take an expression for $a_3$ in this basis, look whether $c$ appears if we use $a_1+c$ and $a_2[+c]$ instead of $a_1$ and $a_2$, and so on.

The case when selection does not involve first or second row is similar (circular permutation of rows).

Lemma 2 is proven.

**Question**: is an algebraic statement similar to Lemma 2 true for quadruples (or $k$-tuples) instead of triples? If not, is the combinatorial statement similar to Lemma 1 true? If not, is the decision complexity bounded by the sum of lengths?

**Remark**: If it is not the case for $k$-tuple for some fixed $k$, then get a new proof of Theorem 1 in a weak form saying that $KM(x,y)$ is not bounded by $|x|+|y|+O(1)$. indeed, it is easy to see that if such a bound were true, this would imply similar bound for $k$-tuples for any $k$, and this would imply Theorem 2 for any $k$.

# References


[1] Gács, P., On the relation between descriptional complexity and algorithmic probability, FOCS 1981. Journal version: *Theoretical Computer Science*, **22**:71–93 (1983).

[2] Levin L.A., On the notion of a random sequence, *Soviet Math. Dokl.*, **14**:1413–1416 (1973).

[3] Li M., Vitányi P., *An Introduction to Kolmogorov Complexity and Its Applications*, Second Edition, Springer, 1997. (638 pp.)

[4] Loveland, D.W., A Variant of the Kolmogorov Concept of Complexity, *Information and Control*, **15**:510–526 (1969).

[5] A. Shen, *Algorithmic Information Theory and Kolmogorov Complexity*. Lecture notes of an introductory course. Uppsala University Technical Report 2000-034. (2000)

[6] Takahashi, H., On a definition of random sequences with respect to conditional probability, *Information and Computation*, **206**:1375–1382 (2008).

[7] Shen, A., Algorithmic variants of the notion of entropy, *Soviet Math. Dokl.*, **29**(3):569–573 (1984).

[8] Uspensky, V.A., Shen, A., Relation between varieties of Kolmogorov complexities, *Math. Systems Theory*, **29**(3):271–192 (1996).

[9] Zvonkin, A.K., Levin, L.A., The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Math. Surveys*, **25**(6):83–124 (1970).