# KOLMOGOROFF ALGORITHMS ARE STRONGER THAN TURING MACHINES

D. Yu. Grigor'ev

UDC 518.5

A predicate is constructed which is recognizable in real time by a Kolmogoroff algorithm but which is not recognizable in real time by a machine with polynomial accessibility.

The purpose of this note is the construction of a predicate recognizable by some Kolmogoroff algorithm (see [1]) in real time, but not recognizable in real time by a machine with polynomial accessibility (see [2], [3]) — in particular by a multidimensional Turing machine.

When added to the main result of [4] it follows that the set of predicates recognizable in real time on a multidimensional Turing machine is strictly contained in the set of predicates recognizable in real time by a Kolmogoroff algorithm.

We will use without explanation the definitions and notation of [2], [3], and [5].

In Sec. 1 we introduce some preliminary material from graph theory. In Sec. 2, the locally complex functions are defined and it is proved that it is impossible to compute such functions in real time on a machine with polynomial accessibility. In Sec. 3 the desired predicate P is constructed as a locally complex function, and a Kolmogoroff algorithm recognizing P is real time is given.

## 1. Uniform Trees.

A binary tree will be called directed (see [5]) if there are either two edges or no edges coming from each vertex, and except for one vertex called the root, there is exactly one edge entering each vertex.

We will also assume that each edge of a binary tree is labeled by 0 or 1 (we will call this a labeling of the edges) so that two edges leaving the same vertex are labeled differently. If moreover each vertex of such a binary tree, except the root, is labeled with 0 or 1 (wee will call this a labeling of the vertices) then we will call the tree a labeled tree.

To each directed branch (see [5]) of a labeled tree starting from the root corresponds a word on the binary alphabet as follows: we write the label of the edges occurring in the branch beginning with the edge leaving the root. The word corresponding to a given branch has length equal to the number of edges in the branch. Distinct branches correspond to distinct words, and thus one may construct an inverse mapping on some set of words on a binary alphabet taking a word $A$ on the alphabet to a branch $\alpha_A$ of the labeled tree starting at the root.

Translated from Zapiski Nauchnykh Seminarov Leningradskogo Otdeleniya Mathematicheskogo Instituta im. V. A. Steklova Akad. Nauk SSSR, Vol. 60, pp. 29-37, 1976. Results presented September 25, 1975.

0090-4104/80/1405-1445$07.50

Let $\Gamma$ be a labeled tree and let $A$ be a word on the binary alphabet. We denote by $D_\Gamma(A)$ the word whose k-th letter is the label of the vertex which is the end of the k-th edge of the branch $\alpha_A$. Note that $|D_\Gamma(A)|=|A|$, where $|X|$ denotes the length of the word $X$

A binary tree is called regular if all of the directed branches starting from the root and finishing at a leaf (see [5]) have the same length, called the depth of the tree.

A regular labeled tree $\Gamma$ of depth $2n$ will be called semiuniform if for any words $A_1$, $A_2$, $B$ of length n on the binary alphabet such that $A_1 \neq A_2$, the right halves of the words $D_\Gamma(A_1B)$ and $D_\Gamma(A_2B)$ are distinct.

A regular labeled tree $\Gamma$ of depth $2^L$ will be called uniform if the following condition is satisfied for any natural numbers $n$ and $k$ such that $(n+1)2^{k+1} \leq 2^L$: any regular subtree of $\Gamma$ of depth $2^{k+1}$ whose root, $\Gamma$ located at depth $n \cdot 2^{k+1}$ is semiuniform.

We will identify two labeled trees $\Gamma$ and $\Delta$ if for each word $A$ on the binary alphabet, the words $D_\Gamma(A)$ and $D_\Delta(A)$ either are both undefined or are both defined and equal.

Note that a labeled tree $\Gamma$ of depth $n$ is defined by the values of the function $\text{Д}_\Gamma(A)$ for $|A|=n$.

We call the labeled tree $\Delta$ an initial subtree of $\Gamma$ if $\Delta$ is a subtree of $\Gamma$ whose root is the root of $\Gamma$.

LEMMA 1. There is an infinite sequence of uniform trees $\Gamma(0), \Gamma(1), \ldots$ of depths $2^0, 2^I, \ldots$ respectively such that for $k \leq n$, $\Gamma(k)$ is an initial subtree of depth $2^k$ of $\Gamma(n)$.

Proof. We define $0 \oplus 0 = 0, 1 \oplus 0 = 1, 0 \oplus 1 = 1, 1 \oplus 1 = 0$. Let $A = a_1 \ldots a_s$, $B = b_1 \ldots b_s$ be twowords of length $s$ on the alphabet $\{0, I\}$. Let $B \oplus A$ denote the word $b_1 \oplus a_1 \ldots b_s \oplus a_s$. Let $\Gamma$ be a labeled tree of depth $s$ and let $A$ be a word of length $s$ on $\{0, I\}$. We will define the tree $\Gamma \oplus A$. In order to do this it is sufficient to define the word $D_{\Gamma \oplus A}(B)$ for all $B$ of length $s$. For any word $B$ on $\{0, I\}$ of length $s$ we define $D_{\Gamma \oplus A}(B) = (D_\Gamma(B)) \oplus A$.

We will prove the following statement:

(*) if $A_1, A_2, B$ are words of length $2^L$ such that $A_1 \neq A_2$ and $\Gamma$ is a uniform tree of depth $2^L$, then:

(*1) $B \oplus A_1 \neq B \oplus A_2$;

(*2) $A_1 \oplus B \neq A_2 \oplus B$;

(*3) $\Gamma \oplus B$ is a uniform tree.

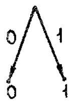
Since (*1) and (*2) are obvious, we will prove (*3). Consider arbitrary words $C$, $E_1$, $E_2$, $H$ on $\{0, I\}$ such that $E_1 \neq E_2$ and such thatthe conditions hold for some $|C| = n \cdot 2^{k+1}$; $|E_1| = |E\ | = |H| = 2^k$; $(n+1)2^{k+1} \leq 2^L$. Suppose $D_{\Gamma \oplus B}(CE_1H) = C'E_1'H'$ and $D_{\Gamma \oplus B}(CE_2H) = C'E_2'H''$ for some words $C', E_1', E_2'$, $H'$, $H''$ such that $|C'| = |C|$, $|H'| = |H''| = |H|$. In order to show $\Gamma \oplus B$ is uniform, we must show that $H' \neq H''$ for such $C, E_1, E_2, H$.

We represent the word $B$ in the following form: $B = B_1 B_2 B_3 B_4$, where $|B_1| = |C|$, $|B_2| = |B_3| = 2^k$. Let $D_\Gamma(CE_1H) = XY_1Z_1$ and $D_\Gamma(CE_2H) = XY_2Z_2$, where $|X| = |C|$ and $|Z_1| = |Z_2| = |H|$. Then $D_{\Gamma \oplus B}(CE_1H)$

$=(X\oplus B_1)(Y_1\oplus B_2)(Z_1\oplus B_3)$ and $D_{\Gamma\oplus B}(CE_2H)=(X\oplus B_1)(Y_2\oplus B_2)(Z_2\oplus B_3)$ . But $Z_1\neq Z_2$ since $\Gamma$ is uniform. Thus by (*2), $Z_1\oplus B_3\neq Z_2\oplus B_3$. On the other hand, $H'=Z_1\oplus B_3$ and $H''=Z_2\oplus B_3$, so that $H'\neq H''$ and $\Gamma\oplus B$ is uniform. The statement (*) is proved.

Now by induction on $L$ we will prove the existence of the tree $\Gamma(L)$ with the properties given in Lemma 1.

Basis Step. Suppose that $L=0$ . We let $\Gamma(0)$ be the labeled tree



Induction Step. Suppose we have constructed the tree $\Gamma(L-1)$ . Let $A$ and $B$ be words of length $2^{L-1}$ on $\{0,I\}$ . We define $D_{\Gamma(L)}(AB)=D_{\Gamma(L-1)}(A)\ D_{\Gamma(L-1)\oplus A}(B)$ . It is not hard to see that $\Gamma(L-1)$ is an initial subtree of $\Gamma(L)$.

We will show $\Gamma(L)$ is uniform. We will show that for any nonnegative integers $n,k$ with $(n+1)2^{k+1}\leqslant 2^L$ a subtree $\Delta$ of depth $2^{k+1}$ of $\Gamma(L)$ whose root is a vertex of $\Gamma(L)$ at depth $n\cdot 2^{k+1}$ is semiuniform.

We consider three cases:

(1) $k<L-1$, $n<2^{(L-1)-(k+1)}$;

(2) $k<L-1$, $n\geqslant 2^{(L-1)-(k+1)}$;

(3) $k=L-1$, $n=0$.

In case (1), $\Delta$ is a subtree of $\Gamma(L-1)$ and the semiuniformity of $\Delta$ follows from the uniformity of $\Gamma(L-1)$. In case (2), $\Delta$ is a subtree of the tree $\Gamma'$ , of depth $2^{L-1}$ , which is the subtree of $\Gamma(L)$ whose root is the end of a branch $\alpha_B$ for some word $B$ of length $2^{L-1}$. Thus $\Gamma'=\Gamma(L-1)\oplus B$, and from (*3) if follows that $\Gamma'$ is uniform. Hence $\Delta$ is semiuniform. Only case (3) is left.

Thus suppose $C_1$, $C_2$, $E$ are words of length $2^{L-1}$ on $\{0,I\}$ with $C_1\neq C_2$ . Suppose that $D_{\Gamma(L)}(C_1E)=H_1B_1$ and $D_{\Gamma(L)}(C_2E)=H_2B_2$ for some words $H_1$, $H_2$, $B_1$, $B_2$ with $|B_1|=|B_2|=2^{L-1}$. We will show $B_1\neq B_2$ . According to the construction of $\Gamma(L)$, $D_{\Gamma(L)}(C_1E)=D_{\Gamma(L-1)}(C_1)\ (D_{\Gamma(L-1)}(E)\oplus C_1)$ and $D_{\Gamma(L)}(C_2E)=D_{\Gamma(L-1)}(C_2)(D_{\Gamma(L-1)}(E)\oplus C_2)$. Hence $B_1=D_{\Gamma(L-1)}(E)\oplus C_1$ and $B_2=D_{\Gamma(L-1)}(E)\oplus C_2$ . By (*1) it follows that $B_1\neq B_2$ . The uniformity of $\Gamma(L)$ and Lemma 1 are proved.

## 2. Locally Complex Functions

We will consider functions whose arguments are words in an alphabet $Q$ and whose values are letters in some alphabet.

A function $g$ is called almost complex if there exist numbers $\ell_0$ and $\theta\ (0\leqslant\theta<1)$ such that for any natural numbers $n$, $k$ and for any words $A$, $C$ in the alphabet $Q$ with $|A|=n\cdot 2^{k+1}$ and $\ell_0\leqslant|C|=2^k$, there is no equivalence class of the relation $\equiv_{A,C}$ containing more than $b^{\theta\cdot 2^k}$ members, where $b$ is the number of elements in $Q$ . (The relation $\equiv_{A,C}$ introduced in [2] is defined as follows: $X_1\equiv_{A,C}X_2$, if $|X_1|=|X_2|=|C|$ and for all $\ell$ such that $1\leqslant\ell\leqslant 2^k$ the identity $g(AX_1c_1\ldots c_\ell)=g(AX_2c_1\ldots c_2)$ holds, where $C=c_1\ldots c_{2^k}$.)

We call a function $f$ locally complex if there is an almost complex function $g$ and two sequences of natural numbers $\{a_k\}_{k=1}^{\infty}$ and $\{b_k\}_{k=1}^{\infty}$ such that (1) $\overline{\lim_{k\to\infty}} b_k=\infty$, (2) for all words $A$ and $B$ for which

$$|A|=\Sigma_{i=1}^{k}a_i+\Sigma_{j=1}^{k-1}b_j \text{ and } |B|\leq b_k$$

the equality $f(AB)=g(B)$ holds.

Systems with polynomial accessible memory were introduced in [2].

LEMMA 2. No locally complex function $f$ is recognizable in real time by a system with polynomial accessible memory.

A proof of this lemma may be constructed following the proof of the main result of [3] for each set of words whose length is between the bounds

$$\Sigma_{i=1}^{k}a_i+\Sigma_{j=1}^{k-1}b_j \text{ and } \Sigma_{i=1}^{k}a_i+\Sigma_{j=1}^{k}b_j \ .$$

We denote this set of words by $I_k$.

We will point out only the the changes which one must make in Lemma 3 of [3]. For the restriction of $f$ to $I_k$ this lemma will be true for $M$ and $R$ such that

$$M=\Sigma_{i=1}^{k}a_i+\Sigma_{j=1}^{k-1}b_j+n\cdot 2^{l+1}, R=2^{l}$$

(see [3]), while in part (a) of Lemma 3 [3] in place of the value $\hat{T}$ one may take the time of processing words of length $b_k$ in $I_k$.

The rest of the proof proceeds without any essential change. As a result one obtains the fact that for sufficiently large $k$, the processing of some words of length $b_k$ in $I_k$ takes a time greater than $b_k$ (one uses the fact that the set of $b_k$ is unbounded), which contradicts the assumption that the recognizing system acts in real time.

Note that for the case of uniform machines (see [2]) this proof of Lemma 2 is not valid, since in part (a) of Lemma3 of [3] for the uniform case, the time of processing all input words exceeds $\Sigma_{i=1}^{k}a_i$, and this bound may be very large. Hence there is no way to obtain a sufficient time bound on the processing of words of length $b_k$ in $I_k$.

## 3. Construction of the Desired Predicate

We will construct a locally complex function $F$ whose arguments are words on the alphabet $\{0,I\}$, whose values are elements of this same alphabet, and which is computable by some Kolmogoroff algorithm in real time.

Suppose that it takes a fixed Kolmogoroff algorithm $T_L$ the time $K'$ to construct the tree $\Gamma(L)$ (see Sec. 1), and suppose that the sequence $\{T_L\}_{L=0}^{\infty}$ is monotone increasing in $L$.

We will define F(A) for any word $A$ in the binary alphabet. We set $F(A)=1$ if

$$\Sigma_{i=0}^{L}T_i+2^{L+1}\leq|A|<\Sigma_{i=0}^{L+1}T_i+2^{L+1}$$

for some $L$ ; if

$$\Sigma_{i=0}^{L-1}T_i+2^{L-1}\leq|A|<\Sigma_{i=0}^{L-1}T_i+2^{L}$$

for some $L$ then $F(A)$ is the last letter in the word $D_{\Gamma(L)}(B)$ where

$$A = CB \text{ and } |C| = \Sigma_{i=0}^{L-1} T_i + 2^{L-1} - 1$$

for some word $C$ on {0, 1}.

The desired predicate $P$ is given as follows: $P(A) \Longleftrightarrow F(A) = 0$, where $A$ is a word on {0,1}.

It is not hard to construct a Kolmogoroff algorithm $K$ recognizing $P$ in real time. For each $L$, from the moment of time $\Sigma_{i=0}^{L-1} T_i + 2^L$ until the moment $\Sigma_{i=0}^{L} T_i + 2^L - 1$ with the help of the algorithm $K'$ the algorithm $K$ constructs the tree $\Gamma(L)$. Then if the input supplies a word $B$ of length $2^L$, the algorithm $K$, avoids the construction of the tree $\Gamma(L)$ along the branch $\alpha_B$ and yields the word $D_{\Gamma(L)}(B)$. In the time interval $\Sigma_{i=0}^{L-1} T_i + 2^L$ to $\Sigma_{i=0}^{L} T_i + 2^L - 1$ the output of $K$ is 1.

LEMMA 3. The function $F$ is locally complex.

Proof. In the definition of locally complex function, we take $2^K$ as $b_K$ and $T_K$ as $a_K$ (see Sec. 2).

We take as the function $g$ (see Sec. 2) the function whose value on the word $A$ is the last letter of the word $D_{\Gamma(L)}(A)$, where $L$ is a number such that $|A| \leq 2^L$. (This definition does not depend on $L$ since each tree $\Gamma(K)$ is an initial subtree of $\Gamma(n)$ for $n \geq K$ and thus for any word $A$ on {0, 1} with $|A| \leq 2^K$, $D_{\Gamma(K)}(A) = D_{\Gamma(n)}(A)$.) We will show that this function is almost complex (see Sec. 2).

Let $A$ and $C$ be words on {0, 1} such that $|C| = 2^K$ and $|A| = n \cdot 2^{K+1}$. Take $L$ with $2^L \geq (n+1)2^{K+1}$. As $\theta$ and $\ell_0$ (in the definition of almost complex function in Sec. 2) we may take 0 and 1 respectively, i.e., as we will prove below, any $\equiv_{A,C}$ equivalence class does not contain more than one element. Suppose otherwise.

Let $X_1$ and $X_2$ be words such that $X_1 \equiv_{A.C} X_2$ and $X_1 \neq X_2$. This means that if $D_{\Gamma(L)}(AX_1C) = A'X_1'C'$ and $D_{\Gamma(L)}(AX_2C) = A'X_2'C''$ for some words $A'$, $X_1'$, $X_2'$, $C'$, $C''$ such that $|A'| = |A|$ and $|C'| = |C''| = 2^K$, then $C'' = C'$. But this contradicts the semiuniformity of the subtree of depth $2^{K+1}$ whose root is the root of the branch $\alpha_A$. The semiuniformity of this subtree follows from the uniformity of the tree $\Gamma(L)$.

Hence we have proved that the function serving as the function $g$ in the definition of locally complex function is almost complex. Thus it follows that $F$ is locally complex. Lemma 3 is proved.

THEOREM. There is a predicate not recognizable in real time by any system with polynomial accessible memory but recognizable in real time by some Kolmogoroff algorithm.

For the proof we take the predicate $P$ and use Lemmas 2 and 3 and also use the Kolmogoroff algorithm $K$ constructed in this section.

An analogous assertion may be proved for the predicate $A$ in [6] by using an extension of the same method.

The author does not know whether one can construct an almost complex or complex function computable in real time by some Kolmogoroff algorithm.

The author would like to acknowledge the advice of A. O. Slisenko and the valuable comments of S. V. Pakhomova, whichsubstantially improved the presentation.

## LITERATURE CITED

1. A. N. Kolmogorov, "Onthe concept of an algorithm," Usp. Mat. Nauk, 8, No. 4(56), 175-176 (1953).
2. S. A. Cook and S. O. Aanderaa, "On the minimum computation time of functions," Trans. Am. Math. Soc., 142, 291-314 (1969).
3. M. S. Paterson, M. J. Fisher, and A. R. Meyer, "An improved overlap argument for on-line multiplication," SIAM-AMS Proc., Soc., 142, 291-314 (1969).
4. A. Schönhage, "Real time simulation of multidimensional Turing machines by storage modification machines," Proj. MAC Tech. Memo., No. 37 (1973).
5. F. Harary, Graph Theory, Addison-Wesley (1969).
6. F. C. Hennie, "On-line Turing machine computation," IEEE Trans. Electrical Comp., EC-15, 1, 35-44 (1966).

# APPLICATION OF SEPARABILITY AND INDEPENDENCE NOTIONS FOR PROVING LOWER BOUNDS OF CIRCUIT COMPLEXITY

D. Yu. Grigor'ev

UDC 518.5:519.1

This note consists of two independent parts. In the first part the concept of an ( $m,c$ )-system for a set of linear forms is introduced, and a lower bound is obtained for the algebraic complexity of the computation of $(m,c)$ -systems on algebraic circuits of a special form. In the second part, the notion of an $\ell$ -independent set of boolean functions is introduced and a lower bound is obtained for a certain complexity measure for circuits of boolean functions computing $\ell$ -independent sets. As a corollary it is shown that the standard algorithm for multiplying matrices or polynomials may be realized by a circuit of boolean functions in a way that is optimal with respect to a selected complexity measure.

In our paper two lower bounds on the complexity of computation of algebraic circuits (defined in [1], [2]) are obtained.

In Sec. 1 a lower bound is found for the computational complexity of a set of linear forms (Theorem 1). The second bound is given in Theorem 2 in Sec. 2. It follows from this theorem that the standard procedures formultiplying multiple-digit numbers and multiplying matrices modulo 2 are optimal in a certain sense.

## 1. Bounds for $(m,c)$ -Systems of Linear Forms

1. In thissection wewill consider thequestion of the complexity of algebraic circuits for the simultaneous computation of a set of linear forms with complex coefficients in the variables $x_1,\dots,x_n$. A set of linear forms may be represented by the matrix of their coefficients, denoted $A$ below, and the problem reduces to the problem of constructing a circuit for the calculation of the product $AX$ where $X$ is the vector of variables $x_1,\dots,x_n$.

Translated from Zapiski Nauchykh Seminarov Leningradskogo Otdeleniya Mathematicheskogo Instituta im. V. A. Steklova Akad. Nauk SSSR, Vol. 60, pp. 38-48, 1976. Main results presented December 12, 1974 and May 29, 1975.