

**THE COMPLEXITY OF  
PROGRAMS  
RECOGNIZING  
MEMBERSHIP OF  
NATURAL NUMBERS  
NOT EXCEEDING  $\binom{n}{k}$   
IN A RECURSIVELY  
ENUMERABLE SET**

MATHEMATICS

1968

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196801.48685>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

**Abstract**

**Full Text**

UDC 51.01:518.5

**MATHEMATICS**

Ya. M. BARZDIN'

**THE COMPLEXITY OF PROGRAMS RECOGNIZING MEMBERSHIP OF NATURAL NUMBERS NOT EXCEEDING  $n$  IN A RECURSIVELY ENUMERABLE SET**

*(Presented by Academician A. N. Kolmogorov, 17 VII 1968)*

Let  $M$  be a set of natural numbers. A. Kolmogorov <sup>(1)</sup> introduced the concept of the complexity of numbered objects. Using the same idea, we introduce the concept of the complexity of programs recognizing membership, in the set  $M$ , of natural numbers not exceeding  $n$ . By this we shall mean, informally, the minimal length of a "program"  $p$  which, for any natural number  $x \leq n$ , makes it possible to say whether  $x$  belongs to the set  $M$  or not. A precise definition of this concept is connected with specifying the "programming method." In our case, by a programming method we shall mean a partial recursive numerical function  $\varphi(p, x)$ . We pass to the exact definitions. By the length  $l(p)$  of the binary number  $p$  we shall mean the length of its binary notation. Define the quantity  $K_\varphi(M; n)$  as the minimal length of numbers  $p$  such that, for all natural  $x \leq n$ ,

$$\varphi(p, x) = \begin{cases} 1, & \text{if } x \in M, \\ 0, & \text{if } x \notin M. \end{cases}$$

If no such  $p$  exists, put  $K_\varphi(M; n) = +\infty$ .

It is not difficult to show that there exists a partial recursive function  $A(p, x)$  (as such a function one may take, for example, the function  $A(p, x)$  from <sup>(1)</sup>) such that for any other partial recursive function  $\varphi(p, x)$  one has

$$K_A(M; n) \leq K_\varphi(M; n) + C_\varphi,$$

where  $C_\varphi$  is a constant independent of  $M$  and  $n$ .

We fix such a function  $A(p, x)$ , and henceforth by the complexity of programs recognizing membership, in the set  $M$ , of natural numbers not exceeding  $n$  (abbreviated notation:  $K(M; n)$ ) we shall mean  $K_A(M; n)$ .

P. Martin-Löf <sup>(2)</sup> showed that almost all infinite binary sequences  $x_1, x_2, \dots, x_n, \dots$  have the property that their initial segments  $x_1, \dots, x_n$  of length  $n$  have complexity asymptotically equal to  $n$ . It follows from this result that almost all sets  $M$  of natural numbers\* have the property that  $K(M; n) \sim n$ .

If  $M$  is a recursive set, then, obviously,

$$K(M; n) \leq C_M,$$

where  $C_M$  is a constant independent of  $n$ .

We shall consider recursively enumerable sets. Let us note that most algorithmically unsolvable mass problems can be formulated as the problem of determining the membership of arbitrary natural numbers  $x$  in some recursively enumerable set  $M$ . In this case  $K(M; n)$  meaningfully denotes the amount of initial information that we must possess in order to solve the first  $n$  individual problems of the given mass problem.

\* Here it is taken into account that between sets  $M$  of natural numbers and infinite binary sequences there exists a one-to-one correspondence

$$h : hM = x_1, x_2, \dots, x_n, \dots,$$

where  $x_n = 1$  if  $n \in M$ , and  $x_n = 0$  if  $n \notin M$ .

**Theorem 1 (upper bound).** For any recursively enumerable set  $M$  and any natural number  $n$ ,

$$K(M; n) \leq \log_2 n + C_M,$$

where  $C_M$  is a constant independent of  $n$ .

**Proof.** Let  $M$  be a recursively enumerable set. As is known [3], for every such set  $M$  there exists a partial recursive function  $\pi_M(x)$ , equal to 1 if  $x \in M$ , and undefined if  $x \notin M$ . Denote by  $T_M$  the Turing machine that computes  $\pi_M$ , and by  $T_M^*(x)$  the number of steps of the machine  $T_M$  required to compute  $\pi_M(x)$ . Let  $t_n$  be an element of the set  $M$ , not exceeding  $n$ , such that for all other elements  $x$  of the set  $M$ , not exceeding  $n$ , the inequality  $T_M^*(x) \leq T_M^*(t_n)$  holds. Obviously,  $l(t_n) \leq \log_2 n + 1$ .

Now suppose  $t_n$  is given. It is easy to see that for any natural  $x \leq n$  the question whether  $x$  belongs to the set  $M$  can be decided as follows. Using the machine  $T_M$ , find  $T_M^*(t_n)$ . Then “run” the machine  $T_M$ , on whose tape  $x$  is written (as in the computation of  $\pi_M(x)$ ), and follow its operation for  $T_M^*(t_n)$  steps. If during this time the machine  $T_M$  halts, then  $\pi_M(x) = 1$  and, consequently,  $x \in M$ ; if it does not halt, then  $x \notin M$ . From the effectiveness of the procedure described it follows that there exists a partial recursive function  $\varphi(p, x)$  such that for any  $n$  and any  $x \leq n$

$$\varphi(t_n, x) = \begin{cases} 1, & \text{if } x \in M, \\ 0, & \text{if } x \notin M. \end{cases}$$

Obviously,  $K_\varphi(M; n) \leq l(t_n) \leq \log_2 n + 1$ . Since  $K(M; n) \leq K_\varphi(M; n) + C_\varphi$ , the assertion of the theorem follows.

**Theorem 2 (lower bound).** There exists a recursively enumerable set  $M$  such that for every natural number  $n$

$$K(M; n) > \log_2 n.$$

**Proof.** By an  $n$ -characteristic function of a set  $M$  we shall mean any function  $\chi(x)$  such that for  $x \leq n$ ,  $\chi(x) = 1$  if  $x \in M$ , and  $\chi(x) = 0$  if  $x \notin M$ . Obviously,  $K(M; n) = \min l(p)$ , where the minimum is taken over all  $p$  such that  $A(p, x)$  is an  $n$ -characteristic function of the set  $M$ .

We now define the desired set  $M$  as follows:  $x \in M$  if and only if  $A(x, x) = 0$ . In that case any  $n$ -characteristic function  $\chi_n(x)$  of the set  $M$  will have the property that for  $x \leq n$ ,  $\chi_n(x) = 1$  if  $A(x, x) = 0$ , and  $\chi_n(x) = 0$  if  $A(x, x) \neq 0$ . It is not difficult to verify (by contradiction) that for any  $p \leq n$ ,  $A(p, x) \neq \chi_n(x)$ . This means that for any  $p$  such that  $A(p, x) = \chi_n(x)$ ,  $l(p) \geq l(n) > \log_2 n$ . The theorem is proved.

We now introduce the notion of the complexity of programs recognizing membership of natural numbers  $x \leq n$  in a set  $M$ , under a restriction on the admissible difficulty of transforming programs. At the same time we shall restrict ourselves to considering transformations of programs only on Turing machines, and as the criterion of the difficulty of such a transformation we shall take the number of steps. We pass to precise definitions. Let  $\mathcal{A}$  be an arbitrary Turing machine that computes  $A(p, x)$ . Denote by  $\mathcal{A}^*(p, x)$  the number of steps of the machine  $\mathcal{A}$  used in computing  $A(p, x)$ . Let  $t$  be some numerical function of one argument. By the complexity  $K_{\mathcal{A}}^t(M; n)$  of programs recognizing membership of natural numbers  $x \leq n$  in the set  $M$ , we shall under-

take the minimal length of numbers  $p$  such that for all natural  $x \leq n$

a)

$$A(p, x) = \begin{cases} 1, & \text{if } x \in M, \\ 0, & \text{if } x \notin M, \end{cases}$$

b)  $\mathcal{A}^*(p, x) \leq t(x)$ .

If such a  $p$  does not exist, we put  $K_{\mathcal{A}}^t(M; n) = +\infty$ .

The following theorem shows that even under very weak restrictions on the admissible difficulty of processing programs, the encouraging result of Theorem 1 no longer holds.

**Theorem 3 (lower bound).** *There exists a recursively enumerable set  $M$  such that, for any general recursive function  $t(x)$  and any natural  $n$ ,*

$$K_{\mathcal{A}}^t(M; n) \geq C_t n,$$

where  $C_t$  is a constant independent of  $n$  (but depending on  $t$ ).

In the proof of this theorem the following two lemmas are used.

**Lemma 1.** *For any general recursive function  $t(x)$  there exists a recursively enumerable set  $M$  such that for any natural  $n$ ,  $K_{\mathcal{A}}^t(M; n) \geq C_0 n$ .*

We shall say that a set  $M'$  is densely reducible to a set  $M$  if there exists a general recursive function  $h(x)$  which reduces  $M'$  to  $M$  (i.e.,  $x \in M'$  if and only if  $h(x) \in M$ ) and, moreover, there exists a constant  $C$  such that for every  $x$ ,  $l(h(x)) \leq l(x) + C$ .

**Lemma 2.** *There exists a recursively enumerable set  $M$  to which every recursively enumerable set is densely reducible.*

Using the fixed-point theorem (3), one can show that as the set  $M$  for Theorem 3 one may take the set  $M$  from Lemma 2.

We turn to the upper bound. Obviously, there exists a machine  $\mathcal{A}$  and a general recursive function  $t_0(x)$  such that for any set  $M$ ,

$$K_{\mathcal{A}}^{t_0}(M; n) \leq n + C.$$

In the case of recursively enumerable  $M$ , for infinitely many  $n$  a stronger assertion is true:

**Theorem 4.** *There exists a Turing machine  $\mathcal{A}$  such that for any recursively enumerable set  $M$  the following holds: for any constant  $C > 0$  there exists a general recursive function  $t(x)$  such that, for infinitely many natural  $n$ ,*

$$K_{\mathcal{A}}^t(M; n) \leq Cn.$$

Let us now consider Turing machines. We shall assume that they all use one and the same fixed external alphabet and that the number of letters of this alphabet is greater than two (so that in this class of machines all partial recursive functions are computable). Let  $\mathcal{M}$  be the set of such machines. We introduce the notion of the complexity of programs recognizing membership of Turing machines with at most  $s$  states in the set  $\mathcal{M}$ . To this end, fix the numbering of Turing machines described in (4)\* (strictly speaking, in (4) Turing machines are assigned only words (codes) in the alphabet  $\{0, 1\}$ , beginning with 1; in order to obtain from this a numbering of Turing machines by natural numbers, we must identify the assigned words with the numbers of which they are the binary notation). Denote the number of a machine  $T$  by  $\bar{T}$ . By the complexity  $H(\mathcal{M}; s)$  of programs,

---

\* Generally speaking, one could also have fixed some other numbering. For the validity of the subsequent assertions, it is important only that the fixed numbering possess the following property, which, with the introduction of the intuitive notion of effectiveness, can be formulated as follows: for any machine (starting, for example, from its diagram) one can effectively construct its number, and from the number—the machine itself (for example, in the form of a diagram).

recognizing membership, among Turing machines with no more than  $s$  states, in the set  $\mathcal{M}$ , we shall mean the minimal length of numbers  $p$  such that for all machines  $T$  with no more than  $s$  states

$$A(p, \bar{T}) = \begin{cases} 1, & \text{if } T \in \mathcal{M}, \\ 0, & \text{if } T \notin \mathcal{M}. \end{cases}$$

A set  $\mathcal{M}$  of Turing machines will be called recursively enumerable (recursive) if the set of numbers of its elements is recursively enumerable (recursive).

It is easy to see that for any recursively enumerable set  $\mathcal{M}$  of Turing machines and any natural number  $s$ ,

$$H(\mathcal{M}; s) \leq C_M s \log_2 s,$$

where  $C_M$  is a constant independent of  $s$  (but depending on  $\mathcal{M}$  and on the size of the fixed external alphabet).

We now consider one natural class of sets of Turing machines and show that, for them, the lower estimate of the quantity  $H(\mathcal{M}; s)$  is close to the upper estimate just mentioned.

Let  $E$  be some property defined for Turing machines and characterized by the fact that if two machines  $T_1$  and  $T_2$  realize identical functions (i.e., tapes with identical contents and identical head positions are transformed into tapes with identical contents), then they either both possess property  $E$  or both do not possess this property. Such a property will be called functional. We shall call the property  $E$  nontrivial if there exist both machines possessing property  $E$  and machines not possessing this property. The set of Turing machines possessing property  $E$  will be denoted by  $\mathcal{M}_E$ .

It follows from the Rice-Uspensky theorem (see Theorem 9, § 11 <sup>(5)</sup>) that if  $E$  is a nontrivial functional property of Turing machines, then  $\mathcal{M}_E$  is a nonrecursive set, i.e.,  $H(\mathcal{M}_E; s) \rightarrow \infty$  as  $s \rightarrow \infty$ . The following theorem sharpens this result.

**Theorem 5.** *Whatever nontrivial functional property  $E$  of Turing machines is considered, the set  $\mathcal{M}_E$  has the property that for any natural number  $s$*

$$H(\mathcal{M}_E; s) \geq C_E s \log_2 s,$$

where  $C_E$  is a constant independent of  $s$  (but depending on  $E$ ).

By analogy with  $K_{\mathfrak{A}}^t(M; n)$ , one can introduce  $H_{\mathfrak{A}}^t(\mathcal{M}; s)$ , where  $t$  is a general recursive function.

**Theorem 6.** *Whatever nontrivial functional property  $E$  is considered, the set  $\mathcal{M}_E$  has the property that, whatever general recursive function  $t(x)$  we take, for any natural number  $s$*

$$H_{\mathfrak{A}}^t(\mathcal{M}_E; s) \geq s^{C_{E,t}},$$

where  $C_{E,t}$  is a constant independent of  $s$  (but depending on  $E$  and  $t$ ).

Computing Center  
of the Latvian State  
P. Stučka University

Received  
7 VI 1968

## CITED LITERATURE

- <sup>1</sup> A. N. Kolmogorov, Problems of Information Transmission, **1**, no. 1 (1965).
- <sup>2</sup> P. Martin-Löf, Theory of Probability, **11**, No. 1 (1966).
- <sup>3</sup> A. I. Mal' tsev, *Algorithms and Recursive Functions*, Nauka, 1965.
- <sup>4</sup> B. A. Trakhtenbrot, *Algorithms and Automatic Solution of Problems*, Moscow, 1960.
- <sup>5</sup> V. A. Uspensky, *Lectures on Computable Functions*, Moscow, 1960.

*Note: Figure translations are in progress. See original paper for figures.*

*Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.*