

# SOME THEOREMS ON THE COMPLEXITY OF NORMAL ALGORITHMS AND COMPUTATIONS

MATHEMATICS

1969

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196901.78424>

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

**Abstract**

**Full Text**

UDC 517.11

**MATHEMATICS**

**M. I. KANOVICH, N. V. PETRI**

## **SOME THEOREMS ON THE COMPLEXITY OF NORMAL ALGORITHMS AND COMPUTATIONS**

*(Presented by Academician P. S. Novikov, 18 VI 1968)*

The article considers two problems: a) the complexity of normal algorithms constructively generated by words of a certain alphabet; b) the complexity of deciding enumerable sets under a restriction on the running time of the deciding algorithms.

We shall use the terminology and concepts introduced in the works <sup>(1,2,4)</sup>. In particular, the length of a description of an algorithm  $\mathfrak{A}$  will be called the **complexity** of the algorithm  $\mathfrak{A}$  and denoted by the symbol  $\mathfrak{A}\mathfrak{J}$ .

1. **Theorem 1.** *Let  $A$  and  $B$  be alphabets each containing at least two letters. Let  $\mathfrak{A}$  be a normal algorithm over the alphabet  $B$  such that, for any word  $Q$  in the alphabet  $B$ , from  $!\mathfrak{A}_{\perp}Q_{\perp}$  it follows that  $\mathfrak{A}_{\perp}Q_{\perp}$  is a description of a normal algorithm over the alphabet  $A$ . Then one can specify a normal algorithm  $\mathfrak{D}$  over the alphabet  $B$  and a natural number  $C$  such that, for any word  $Q$  in the alphabet  $B$ , the following three conditions are satisfied:*

- a) *the algorithm  $\mathfrak{D}$  is applicable to the word  $Q$ ;*
- b) *the word  $\mathfrak{D}_{\perp}Q_{\perp}$  is a description of a normal algorithm in a two-letter extension of the alphabet  $A$ , of complexity not exceeding*

$$\frac{\log_2 |\partial|}{\log_2 |A^{\partial}|} \cdot |Q^{\partial}| + C;$$

- c) *if  $!\mathfrak{A}_{\perp}Q_{\perp}$ , then the words  $\mathfrak{A}_{\perp}Q_{\perp}$  and  $\mathfrak{D}_{\perp}Q_{\perp}$  are descriptions of normal algorithms equivalent with respect to the alphabet  $A$ .*

We note that in the case where one of the alphabets under consideration has one letter, analogous theorems hold.

Theorem 1 is very convenient in estimating the complexities of normal algorithms through the complexities of the algorithms participating in their construction. Thus, for example, there exists a natural number  $C$  such that, what-

ever the alphabet  $A$  and the normal algorithms  $\mathfrak{A}$  and  $\mathfrak{B}$  in this alphabet may be, one can construct a normal algorithm  $\mathfrak{C}$  in a two-letter extension of the alphabet  $A$ , equivalent with respect to the alphabet  $A$  to the normal composition of the algorithms  $\mathfrak{A}$  and  $\mathfrak{B}$ , whose complexity does not exceed the quantity  $2\mathfrak{A}\mathfrak{B} + 2\mathfrak{B}\mathfrak{A} + C|A^\theta|$ .

2. At present many results are known that are connected with estimates of the complexity of algorithms performing some work; however, the time characteristics of the computation process are usually not taken into account. We shall consider some questions connected with the computation of Boolean functions by normal algorithms under a restriction on the running time of the computing algorithms.

A Boolean function of  $n$  arguments will be called  $(M, T)$ -**computable** if there exists a  $\Phi$ -algorithm computing it, of complexity not exceeding  $M$ , and such that its running time on each  $n$ -dimensional Boolean vector is bounded by the number  $T$ . If a Boolean function is not  $(M, T)$ -computable, then we shall say that it is  $(M, T)$ -**noncomputable** (here  $M$  and  $T$  are natural numbers).

**Theorem 2.** One can construct an algorithm  $\mathfrak{A}$  over the alphabet  $0|$ , possessing the following properties:

- a) for every natural number  $n$ , if  $!\mathfrak{A}_{[n]}$ , then  $\mathfrak{A}_{[n]}$  is a record of a Boolean function of  $n$  arguments;
- b) for every general recursive function  $f$  and every natural number  $p$  one can specify a natural number  $m$  exceeding it such that  $!\mathfrak{A}_{[m]}$  and  $\mathfrak{A}_{[m]}$  is a record of a  $(2^m/3, f(m))$ -noncomputable Boolean function.

We note that any algorithm  $\mathfrak{A}$  for which properties a) and b) of Theorem 2 hold is incomplete.

3. Theorem 2 makes it possible to obtain a certain estimate of the complexity of deciding enumerable sets under a restriction on the running time of the deciding algorithms.

By the symbol  $\tau(n, \mathfrak{B}, \mathfrak{A}, t)$  we shall denote the following assertion: “the algorithm  $\mathfrak{B}$  is applicable to all words of length  $n$  in the alphabet  $A$  and annihilates exactly those of them to which the algorithm  $\mathfrak{A}$  is applicable; moreover, the running time of the algorithm  $\mathfrak{B}$  on each word of length  $n$  in the alphabet  $A$  is bounded by the number  $t$ .”

**Theorem 3.** Let  $A \supseteq 0|$ . There exists a general recursive function  $f$  such that, for every algorithm  $\mathfrak{A}$  over the alphabet  $0|$  and every natural number  $n$ , it is false that there is no  $\Phi$ -algorithm  $\mathfrak{B}$  such that  $\tau(n, \mathfrak{B}, \mathfrak{A}, f(n))$  and  $\mathfrak{B} \leq 2^n + 64$ .

The impossibility of substantially strengthening Theorem 3 follows from the following proposition.

**Theorem 4.** There exists an algorithm  $\mathfrak{A}$  over the alphabet  $0|$  such that, for every general recursive function  $f$  and every natural number  $p$ , one can specify

a natural number  $m$  exceeding it such that there is no  $\Phi$ -algorithm  $\mathfrak{B}$  for which  $\tau(m, \mathfrak{B}, \mathfrak{A}, f(m))$  and  $\mathfrak{B} \leq 2^m/3$ .

Theorems 2 and 4 remain valid when the temporal signaling function is replaced by any other signaling function (for the definition of signaling function, see <sup>(3)</sup>).

Moscow State University  
named after M. V. Lomonosov

Received  
13 VI 1968

### CITED LITERATURE

- <sup>1</sup> A. A. Markov, *Trudy Mat. Inst. im. V. A. Steklova AN SSSR*, **42** (1954).
- <sup>2</sup> A. A. Markov, *Izv. AN SSSR, Ser. Mat.*, **31**, 161 (1967).
- <sup>3</sup> B. A. Trakhtenbrot, *Complexity of Algorithms and Computations*. Novosibirsk, 1967.
- <sup>4</sup> N. A. Shanin, *Trudy Mat. Inst. im. V. A. Steklova AN SSSR*, **52**, 226 (1958).

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

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