



---

Soviet-era science, translated into English

# COMPLEXITY OF ALGORITHMS AND THEIR RUNNING TIME

1969

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196901.96719>

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

**Abstract**

**Full Text**

UDC 517.11

**MATHEMATICS**

**N. V. PETRI**

## **COMPLEXITY OF ALGORITHMS AND THEIR RUNNING TIME**

*(Presented by Academician A. N. Kolmogorov on 10 X 1968)*

The article is devoted to estimates of the complexity of normal algorithms whose running time is bounded.

1. An essential estimate of an algorithm that performs a prescribed task on a finite set is the greatest of the durations of its application to the objects of this set. Bounding this quantity affects another important estimate of algorithms—their complexity. From the point of view of the relation between the complexity of algorithms and the duration of their operation, we shall consider the following problems: a) recognition of the applicability of normal algorithms to words of bounded length; b) recognition of the  $M$ -computability of Boolean functions and the construction of minimizing algorithms <sup>(2)</sup>.
2. We use here the terminology and concepts introduced in <sup>(1,2)</sup>. Thus, the length of the representation of a normal algorithm  $\mathfrak{A}$  will be called the complexity of the algorithm  $\mathfrak{A}$ ; normal algorithms in the alphabet  $o|abc$  will be called  $\Phi$ -algorithms.
3. Without essentially narrowing problem a), we shall restrict ourselves to the case when the initial and resolving algorithms are  $\Phi$ -algorithms, and the words whose applicability to the initial algorithms is required to be recognized are words in the alphabet  $o|$ .

Let  $\mathfrak{A}$  be a  $\Phi$ -algorithm, and let  $n$  and  $l$  be natural numbers. By the symbol  $\rho(n, l, \mathfrak{A})$  we shall denote the following assertion: there exists a  $\Phi$ -algorithm  $\mathfrak{B}$  of complexity not greater than  $l$ , which is  $n$ -resolving for  $\mathfrak{A}$  with respect to the alphabet  $o|$  <sup>(4)</sup>.

By the symbol  $\rho(n, l, t, \mathfrak{A})$  we shall denote the following assertion: there exists a  $\Phi$ -algorithm  $\mathfrak{B}$  of complexity not greater than  $l$ , which is  $n$ -resolving for  $\mathfrak{A}$  with respect to the alphabet  $o|$ , and moreover the number of steps in the process of applying the algorithm  $\mathfrak{B}$  to each word of length not greater than  $n$  in the alphabet  $o|$  is bounded by the natural number  $t$ .

4. We shall be interested in such time bounds for  $n$ -resolution that are given by general recursive functions. In the six propositions formulated below, the letter  $\mathfrak{A}$  will serve as a variable denoting  $\Phi$ -algorithms, the letter  $f$  as a variable for general recursive functions of one argument, and the letters  $m, n, C$  as variables for natural numbers.

The following proposition is comparatively easy to prove.

$$4.1. \quad \exists f C \forall \mathfrak{A} n \neg \rho(n, 2 \cdot 2^n + C, f(n), \mathfrak{A}).$$

From Theorem 3.3 of <sup>(3)</sup> it follows that the following proposition is also true:

$$4.2. \quad \exists \mathfrak{A} \forall f m \exists n \left( n \geq m \ \& \ \neg \rho \left( n, \frac{2^n}{3}, f(n), \mathfrak{A} \right) \right).$$

We shall formulate two more propositions supplementing the existing picture.

$$4.3. \quad \exists \mathfrak{A} \forall f \exists C \forall n \neg \rho \left( n, \frac{2^n}{C}, f(n), \mathfrak{A} \right).$$

$$4.4. \quad \forall \mathfrak{A} C \neg \neg \exists f \forall m \exists n \left( n \geq m \ \& \ \neg \rho \left( n, \frac{2^n}{C}, f(n), \mathfrak{A} \right) \right).$$

From Theorem 3.1 of <sup>(4)</sup> one may conclude that

$$\forall \mathfrak{A} \exists C \forall n \neg \rho(n, n + C, \mathfrak{A}).$$

Comparison of this estimate with Theorems 4.2 and 4.3 shows that limiting the running time of deciding algorithms leads, generally speaking, to a considerable increase in the complexity of the decision. The following proposition shows that this is not always so.

4.5.

$$\exists \mathfrak{A} C f \forall n (\neg \rho(n, 2n + C, f(n), \mathfrak{A}) \ \& \ \neg \rho(n, n/6 - C, \mathfrak{A})).$$

5. The upper estimates for the complexity of algorithms recognizing the  $M$ -noncomputability of Boolean functions of  $N$  arguments, and of algorithms minimizing for  $N$  arguments, proved in paper <sup>(2)</sup>, can be substantially improved <sup>(4)</sup>. However, in the presence of time restrictions they turn out to be close to optimal, as will be seen from what follows.
6. By the symbol  $\omega^+(l, t, M, N)$  we shall denote the following assertion: there exists a  $\Phi$ -algorithm  $\mathfrak{A}$  of complexity not greater than  $l$ , which recognizes the  $M$ -noncomputability of Boolean functions of  $N$  arguments <sup>(2)</sup>, and the time of its work on the record of each Boolean function of  $N$  arguments does not exceed  $t$ .

By the symbol  $\omega(l, t, M, N)$  we shall denote the following assertion: there exists a  $\Phi$ -algorithm  $\mathfrak{A}$  of complexity not greater than  $l$ , which recognizes the  $M$ -noncomputability of Boolean functions of  $n$  arguments for every  $n \leq N$ , and the time of its work on the record of each Boolean function of  $n$  arguments ( $n \leq N$ ) does not exceed  $t$ .

In the following six propositions the letter  $f$  serves as a variable for general recursive functions of one argument, and the letters  $m, n, M, N$  and  $C$  for natural numbers.

It is comparatively simple to establish the validity of the following proposition:

6.1.

$$\exists f C \forall M N \neg \omega(2^{M+3N} + C, f(N), M, N).$$

The next three propositions are analogues of Propositions 4.2 and 4.3.

6.2.

$$\exists m C \forall f M (M \geq m \supset \forall n \exists N (N \geq n \ \& \ \neg \omega^+(2^N/C, f(N), M, N))).$$

6.3.

$$\forall f \exists m C \forall M N (m \leq M < 2^N/3 \supset \neg \omega^+(2^{N+M}/C, f(N), M, N)).$$

6.4.

$$\exists m \forall f \exists n C \forall M N (m \leq M < 2^{N-n}/3 \supset \neg \omega(2^{M+N}/C, f(N), M, N)).$$

It is also possible to prove analogues of Theorem 4.4.

6.5.

$$\exists C \forall M \neg \neg \exists f \forall n \exists N (N \geq n \ \& \ \omega^+(3M + C, f(N), M, N)).$$

6.6.

$$\forall M C \neg \neg \exists f \forall n \exists N (N \geq n \ \& \ \neg \omega(2^N/C, f(N), M, N)).$$

7. We formulate three more propositions on the complexity of algorithms minimizing for  $N$  arguments <sup>(2)</sup> under a restriction on running time.

By the symbol  $\mu^+(l, t, N)$  we shall denote the following assertion: there exists an  $M$ -algorithm (see <sup>(2)</sup>) of complexity not greater than  $l$ , which minimizes for  $N$  arguments, and the time of its work on the record of each Boolean function of  $N$  arguments does not exceed  $t$ .

7.1.

$$\exists f C \forall N \neg \neg \mu^+(2^{2^N+N} + C, f(N), N).$$

7.2.

$$\exists C \forall f n \exists N (N \geq n \ \& \ \neg \mu^+(2^{2^N/3}/C, f(N), N)).$$

7.3.

$$\forall f \exists C \forall N \neg \mu^+(2^{2^N/3}/C, f(N), N).$$

Moscow State University  
named after M. V. Lomonosov

Received  
10 X 1968

## REFERENCES

- <sup>1</sup> A. A. Markov, *Tr. Mat. Inst. im. V. A. Steklova*, **42** (1954).
- <sup>2</sup> A. A. Markov, *Izv. Akad. Nauk SSSR, Ser. Mat.*, **31**, 161 (1967).
- <sup>3</sup> M. I. Kanovich, N. V. Petri, *DAN*, **184**, No. 6 (1969).
- <sup>4</sup> N. V. Petri, *DAN*, **185**, No. 1 (1969).

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

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