ON ESTIMATING THE COMPLEXITY OF NORMAL ALGORITHMS

MATHEMATICS

1969

SovietRxiv

Abstract

Full Text

UDC 51.01:518.5

MATHEMATICS

D. A. OSTRUKHOV

ON ESTIMATING THE COMPLEXITY OF NORMAL ALGORITHMS

(Presented by Academician A. A. Dorodnitsyn, 24 VII 1967)

1. In considering questions connected with the construction of normal algorithms, it is often necessary to indicate a general method of construction, for every alphabet, of a normal algorithm associated with this alphabet by a certain relation. Such are, for example, the methods of constructing the reversing and doubling algorithms given in A. A. Markov' s monograph (^1) (II. § 4.12, II. § 4.13). The reversing and doubling algorithms for words in an n-letter alphabet constructed in accordance with these instructions have complexity (length of representation) of order n^2. The question arises: can every general method of constructing a normal algorithm over an n-letter alphabet be replaced by another method that gives the desired algorithm with complexity depending linearly on n; in particular, can one indicate a method of construction, for an n-letter alphabet, of a universal algorithm of complexity of order n (the universal algorithm $\mathfrak{U}_n$, constructed by E. S. Orlovskii in (^4), has complexity of order n^2)? Below this question is considered in a precise formulation.

2. Since we shall be speaking only about normal algorithms, we agree to regard the terms *algorithm* and *normal algorithm* as synonyms.

Following A. A. Markov (^2), by the representation of an algorithm $\mathfrak{A}$ in an alphabet not containing the letters α, β, γ, we shall mean the word $\mathfrak{A}^i$ which is obtained if, in the scheme of the algorithm $\mathfrak{A}$, in all simple substitution formulas the arrow is replaced by the letter α, in all final substitution formulas the word $\rightarrow \cdot$ is replaced by the letter β, at the end of each formula the letter γ is appended, and the resulting words are written one after another in the order in which the corresponding formulas occur in the scheme $\mathfrak{A}$.

By the complexity of the algorithm $\mathfrak{A}$ we shall mean the length of the word $\mathfrak{A}^i$. We shall denote the complexity of the algorithm $\mathfrak{A}$ by the symbol $\mathfrak{A}\mathfrak{Z}$.

We shall consider natural numbers as words in the one-letter alphabet $|$ ((^1), I. § 3.13).

By the symbol $\mathfrak{R}_{\sigma}^{\chi}$ we shall denote the algorithm whose scheme is obtained from the closure scheme of the algorithm $\mathfrak{R}$, if every final substitution formula of the form $U \to \cdot V$ is replaced by the formula $U \to \chi V$ and then every substitution formula of the form $\to V$ is replaced by the formula $\varepsilon \to \varepsilon V$ (($\frown$1), III. § 2.2, III. § 3.2.1).

3. Let us fix a sequence $a_1, a_2, \dots, a_n, \dots$ of pairwise distinct letters, and let the letters $a, b, \alpha, \beta, \gamma, \delta$ be pairwise distinct and distinct from the letters of this sequence. We form a sequence of alphabets A_n, putting $A_0 \rightleftarrows \Lambda$, $A_{n+1} \rightleftarrows A_n a_{n+1}$.

For every word X formed from the letters $\alpha, \beta, \gamma, a_1, \dots, a_n, \dots$, its translation $[X^\tau]$ is defined in the following way: $[\alpha^\tau \rightleftarrows aba$, $[\beta^\tau \rightleftarrows ab^2a$, $[\gamma^\tau \rightleftarrows ab^3a$, $[a_i^\tau \rightleftarrows ab^{3+i}a$ $(i \geqslant 1)$, $[X\xi^\tau \rightleftarrows [X^\tau][\xi^\tau$ (ξ is one of the letters $\alpha, \beta, \gamma, a_1, \dots, a_n, \dots$).

The notation of an algorithm $\mathfrak{A}$ in any one of the alphabets A_n can be defined as the translation of the word $\mathfrak{A}^i$. We shall denote the notation of the algorithm $\mathfrak{A}$ by $\{\mathfrak{A}\}$. Whatever the natural number n and the algorithm $\mathfrak{A}$ in the alphabet A_n may be, $\{\mathfrak{A}\}$ is a word in the alphabet ab.

A word Q in the alphabet ab will be called a word of type $\mathfrak{z}$ if there can be specified a natural number n and an algorithm $\mathfrak{A}$ in the alphabet A_n such that

$$Q \simeq \xi \mathfrak{A} \mathfrak{z}.$$

For every word Q of type $\mathfrak{z}$, define the number $N(Q)$ to be the greatest of the numbers j ($j \geqslant 0$) such that the word b^{3+j} occurs in Q. For every word Q of type $\mathfrak{z}$ and for every natural number $j \geqslant N(Q)$, by $\langle Q, j \rangle$ we denote the algorithm in the alphabet A_j whose notation is the word Q.

By a constructive sequence of normal algorithms (c.s.n.a.) we shall mean any normal algorithm $\mathfrak{A}$ that transforms any natural number into some word of type $\mathfrak{z}$.

4. **Theorem 1.** For every natural number n an algorithm $\mathfrak{U}_n$ over the alphabet $A_n ab\alpha\beta\gamma\delta$ can be constructed satisfying the following conditions:

1) For any word P in the alphabet A_n and for any algorithm $\mathfrak{A}$ in A_n,

$$\mathfrak{U}_n(\mathfrak{A}\mathfrak{w}\delta P) \simeq \mathfrak{A}(P);$$

2) If P is a word in the alphabet A_n, $j \geqslant n$, and $\mathfrak{A}$ is an algorithm in the alphabet A_j that transforms every word in A_n to which it is applicable also into a word in the alphabet A_n, then

$$\mathfrak{U}_n(\xi \mathfrak{A} \mathfrak{z}\, \delta P) \simeq \mathfrak{A}(P).$$

3) If P is a word in the alphabet A_n, then

$$\mathfrak{U}_n : P \vdash [P^\tau \neg].$$

4) $\mathfrak{U}_n\mathfrak{z} = 10n + 515$.

Returning to the question posed at the beginning of the note, we propose to regard Theorem 1 and the following Theorem 2 as the answer to it.

Theorem 2. For every c.s.n.a. $\mathfrak{R}$ there can be constructed a c.s.n.a. $\mathfrak{S}$ such that, for every natural number n:

1) the algorithms

$$\langle \mathfrak{R}(n), \max(N(\mathfrak{R}(n)), n) \rangle \quad \text{and} \quad \langle \mathfrak{S}(n), \max(N(\mathfrak{S}(n)), n) \rangle$$

are equivalent with respect to the alphabet A_n;

2)

$$\langle \mathfrak{S}(n), \max(N(\mathfrak{S}(n)), n) \rangle \mathfrak{z} = 11n + 432 + \mathfrak{R}^x_\xi \mathfrak{z}.$$

The proof of Theorem 2 is based on the existence of a c.s.n.a. generating algorithms with the properties formulated in items 2), 3), 4) of Theorem 1.

Theorem 3. There can be constructed a c.s.n.a. $\mathfrak{R}$ satisfying the following condition: if n is a natural number and $\mathfrak{A}$ is an algorithm over the alphabet A_n equivalent to the algorithm

$$\langle \mathfrak{R}(n), \max(N(\mathfrak{R}(n)), n) \rangle$$

with respect to A_n, then

$$\mathfrak{A}\mathfrak{z} \geqslant 5n - 5.$$

Theorem 4. Whatever the natural numbers n and m, it is not true that an algorithm $\mathfrak{A}$ in the alphabet A_n satisfying the following condition is impossible: there is no algorithm $\mathfrak{B}$ over the alphabet A_n, equivalent to the algorithm $\mathfrak{A}$ with respect to the alphabet A_n, and such that $\mathfrak{B}\mathfrak{z} \leqslant m$.

5. The author expresses deep gratitude to A. A. Markov and N. M. Nagorny for their attention to the work and for advice which greatly influenced its final editing.

Received
23 VI 1967

REFERENCES

1. A. A. Markov, *Tr. Matem. inst. im. V. A. Steklova AN SSSR*, **42** (1954).

2. A. A. Markov, *DAN*, **157**, No. 2, 262 (1964).

3. N. A. Shanin, *Tr. Matem. inst. im. V. A. Steklova AN SSSR*, **52** (1958).

4. E. S. Orlovskii, *Tr. Matem. inst. im. V. A. Steklova AN SSSR*, **52** (1958).