Transactions of the Institute for Informatics and Automation Problems of NAS RA. Collected reports of participants of JAF-23.

ON THE REPRESENTATION OF ARITHMETICAL AND STRING FUNCTIONS IN FORMAL LANGUAGES

M. H. Khachatryan

Formal languages LA and LW are introduced for representation of arithmetical primitive recursive functions in LA and primitive recursive string functions in LW. A natural correspondence between arithmetical functions and string functions is defined. Shannon's functions describing relations between the complexities of the representation of functions in LA and LW are introduced; the linear upper and lower bounds for Shannon's functions are obtained.

§1. **INTRODUCTION.** In this paper primitive recursive arithmetical functions and primitive recursive string functions are considered; a natural correspondence between arithmetical functions and string functions in a given alphabet is defined. Formal languages LA and LW for the representation of primitive recursive arithmetical functions and primitive recursive string functions are introduced. The following problem is investigated: what is the change of complexity of representation of a function in a formal language when we pass from a function expressed in one language to the corresponding function expressed in another language? Shannon's function describing these changes are defined; linear upper and lower bounds are established in main theorem proved below.

§2. Let us recall some definitions given in [5], [6], [7]. We consider a finite list of different symbols $A = \{a_1, a_2, ..., a_p\}$, where $p > 1$; such a list will be called *alphabet*, and the symbols $a_1, a_2, ..., a_p$ will be called *letters* of A. A *string* (or word) in A is any finite sequence $a_{i_1}a_{i_2}...a_{i_t}$ of letters in A. The empty string having no letter is also considered; it will be denoted by $\wedge$. The number t is said to be *length* of the string $a_{i_1}a_{i_2}...a_{i_t}$; the length of $\wedge$ is 0. The length of a string P will be defined by $|P|$. The set of all strings in A (including $\wedge$) will be denoted by A^*. The *alphabetic number* $\pi(P)$ of the string P in

A is denoted by the following equalities:

$$\pi(\Lambda) = 0;$$

$$\pi(a_{i_1}a_{i_2}...a_{i_t}) = i_t + i_{t-1}\cdot p + ... + i_1 \cdot p^{t-1}.$$

It is known (see, for example, [7]) that such enumeration of strings in A defines a one-to-one correspondence between A^* and the set $N = \{0, 1, 2\ldots\}$ of all natural numbers. By $a_p n$ (or, shortly, an) we denote the string in A having the alphabetic number n. The *concatenation* of strings $P = a_{i_1}a_{i_2}...a_{i_t}$ and $Q = a_{j_1}a_{j_2}...a_{j_s}$ is the string $a_{i_1}a_{i_2}...a_{i_t}a_{j_1}a_{j_2}...a_{j_s}$; it will be denoted by PQ.

n-dimensional arithmetical function (or, shortly, arithmetical function) is defined as any mapping of N^n into N (where N^n is the n-th Cartesian degree of the set $N = \{0, 1, 2...\}$); n-dimensional string function in the alphabet A (or, shortly, string function) is defined as any mapping of $(A^*)^n$ into A^*. The notion of *primitive recursive function* (or primitive recursive arithmetical function) is defined in a usual way ([5], [6], [7]).

Let us recall this definition. Primitive recursive functions are functions obtained from the *basic functions* 0, $I_l^k(x_1, x_2...x_k) = x_l$ (where $k \geq 1, 1 \leq l \leq k$), $s(x) = x + 1$, by the operators of *superposition* and *primitive recursion* ([5], [6], [7]). The corresponding definitions are as follows. We say that a k-dimensional arithmetical function f (i.e. depending on k variables) is obtained from arithmetical functions $g, h_1, h_2, ..., h_m$ by the operator S of superposition and denote this statement by $f = S(g, h_1, h_2, ..., h_m)$, if the functions $g, h_1, h_2, ..., h_m$ depend on, correspondingly $m, k, k, , k$ variables, and the following equality holds for all natural numbers $x_1, x_2, ..., x_k$:

$$f(x_1, x_2, ..., x_k) = g(h_1(x_1, x_2, ..., x_k), h_2(x_1, x_2, ..., x_k), ..., h_m(x_1, x_2, ..., x_k)).$$

We say that an $(m+1)$-dimensional arithmetical function f is obtained from arithmetical functions α and β by the operator R of primitive recursion and denote this statement by $f = R(\alpha, \beta)$, if α and β depend on, correspondingly m and $m + 2$ variables, and the following equalities hold for all natural numbers $x_1, x_2, ..., x_m, y$:

$$f(x_1, x_2, ..., x_m, 0) = \alpha(x_1, x_2, ..., x_m);$$

$$f(x_1, x_2, ..., x_m, y + 1) = \beta(x_1, x_2...x_m, y, f(x_1, x_2, ..., x_m, y)).$$

Below we shall consider some special forms of superposition and primitive recursion, namely, operators of *Sbl*-superposition, *Sbr*-superposition, *Sel*-superposition, *Ser*-superposition (or, shortly, operators *Sbl*, *Sbr*, *Sel*, *Ser*) as well as a generalized form of the operator R of primitive recursion. The mentioned operators of superposition are defined as follows. Every operator of $Sbl-, Sbr-, Sel-, Ser$-superposition is applied to any functions $g, h_1, h_2, ..., h_m$, depending, correspondingly on $v, k_1, k_2, ..., k_m$ variables, where $m \leq v$; its value is a function f depending on $v - m + k$ variables, where $k = max(k_1, k_2...k_m)$, and satisfying the conditions which are given below for every mentioned operator:

(1) The function $f^{(1)} = Sbl(g, h_1, h_2, ..., h_m)$ given by the operator of Sbl-superposition satisfies the following equality:

$f^{(1)}(x_1, x_2, ..., x_{v-m+k}) =$

$g(h_1(x_{k-k_1+1}, x_{k-k_1+2}, ..., x_k), h_2(x_{k-k_1+1}, x_{k-k_1+2}, ..., x_k),$

$...,$

$h_m(x_{k-k_1+1}, x_{k-k_1+2}, ..., x_k), x_{k+1}, x_{k+2}, ..., x_{v-m+k}).$

(2) The function $f^{(2)} = Sbr(g, h_1, h_2, ..., h_m)$ given by the operator of Sbr-superposition satisfies the following equality:

$f^{(2)}(x_1, x_2, ..., x_{v_m+k}) =$

$g(h_1(x_1, x_2...x_{k_1}), h_2(x_1, x_2...x_{k_2}), ..., h_m(x_1, x_2...x_{k_m}), x_{k+1}, x_{k+2}, ..., x_{v-m+k}).$

(3) The function $f^{(3)} = Sel(g, h_1, h_2, ..., h_m)$ given by the operator of Sel-superposition satisfies the following equality:

$f^{(3)}(x_1, x_2, ..., x_{v-m+k}) =$

$g(x_1, x_2...x_{v-m}, h_1(x_{v-m+k-k_1+1}, x_{v-m+k-k_1+2}, ..., x_{v-m+k}),$

$h_2(x_{v-m+k-k_2+1}, x_{v-m+k-k_2+2}, ..., x_{v-m+k}),$

$...,$

$h_m(x_{v-m+k-k_m+1}, x_{v-m+k-k_m+2}, ..., x_{v-m+k})).$

(4) The function $f^{(4)} = Ser(g, h_1, h_2, ..., h_m)$ given by the operator of Ser-superposition satisfies the following equality:

$f^{(4)}(x_1, x_2, ..., x_{v_m+k}) =$

$g(x_1, x_2...x_{v-m}, h_1(x_{v-m+1}, x_{v-m+2}, ..., x_{v-m+k_1}), h_2(x_{v-m+1}, x_{v-m+2}, ..., x_{v-m+k_2}),$

$...,$

$h_m(x_{v-m+1}, x_{v-m+2}, ..., x_{v-m+k_m})).$

Clearly, the operator S can be considered as partial case of operators Sbl, Sbr, Sel, Ser; it is obtained from every of mentioned operators when $v = m$ and $k_1 = k_2 = ... = k_m$.

The generalized form of the operator R is defined as follows. Let α be an arithmetical function depending on l variables, and β be an arithmetical function depending on k variables, where $k \geq l+2$. Then the function $f = R(\alpha, \beta)$ obtained from α and β by the generalized form of the operator R depends on $(k+1)$ variables and is defined by the following equalities:

$$f(x_1, x_2, ..., x_{k-2}, 0) = \alpha(x_1, x_2, ..., x_l);$$

$$f(x_1, x_2, ..., x_{k-2}, y+1) = \beta(x_1, x_2, ..., x_{k-2}, y, f(x_1, x_2, ..., x_{k-2}, y)).$$

Clearly, the usual form of the operator R is obtained from the generalized form when $k = l+2$.

Now let us define some relations between arithmetical functions and string functions in a given alphabet $A = \{a_1, a_2, ..., a_p\}$, where $p > 1$. Let f be an m-dimensional arithmetical function, and F be an m-dimensional string function. We say that f is the function *representing* F (or that F is the function *representable* by f) if the following equality holds for all natural numbers $x_1, x_2, ..., x_m$:

$$F(\alpha x_1, \alpha x_2, ..., \alpha x_m) = \alpha f(x_1, x_2, ..., x_m).$$

If a string function F is representable by an arithmetical function f, then we say also that the functions F and f *correspond* one to another.

A string function F is said to be *primitive recursive* if and only if the arithmetical function representing F is a primitive recursive arithmetical function (cf. [7]).

Let us define now *basic string functions* in the alphabet A as well as operators of *superposition* and *alphabetic primitive recursion* (cf. [7]) on string functions. These definitions are similar to corresponding definitions in the theory of arithmetical primitive recursive functions.

Basic string functions in A are following functions:

(1) The 0-dimentional function such that its value is the empty string Λ; such a function will be denoted by Λ.

(2) The function $S_i(P) = Pa_i$, where $1 \leq i \leq p$.

(3) The function $I_l^k(P_1, P_2, ..., P_k) = P_l$, where $k \geq 1$, and $1 \leq l \leq k$.

Clearly, all these string functions are primitive recursive. For example, every function $S_i(P)$ is representable by arithmetical primitive recursive function $s_i(x) = px + i$.

The operators $\boldsymbol{S}, \boldsymbol{Sbl}, \boldsymbol{Sbr}, \boldsymbol{Sel}, \boldsymbol{Ser}$ on string functions are defined similarly to the operators S, Sbl, Sbr, Sel, Ser on arithmetical functions.

The operator $\boldsymbol{R}$ of *alphabetic (string) primitive recursion* is defined as follows. Let $G, H_1, H_2, ..., H_p$ be string functions depending, correspondingly on $l, k_1, k_2, ..., k_p$ variables, where $k_i \geq 2$ for all i, $1 \leq i \leq p$, and $max(k_1, k_2, ..., k_p) \geq l+2$. Then the function $F = \boldsymbol{R}(G, H_1, H_2, ..., H_p)$ obtained by the operator $\boldsymbol{R}$ from $G, H_1, H_2, ..., H_p$ depends on $(k-1)$ variables, where $k = max(k_1, k_2, ..., k_p)$, and is defined by the following equalities, where $P_1, P_2, ..., P_{k-2}, Q$ are any strings belonging to A^*:

$$F(P_1, P_2, ..., P_{k-2}, \wedge) = G(P_1, P_2, ..., P_l);$$

$$F(P_1, P_2, ..., P_{k-2}, Qa_i) = H_i(P_{k-k_i+1}, P_{k-k_i+2}, ...P_{k-2}, Q, F(P_1, P_2, ..., P_{k-2}, Q)),$$

for all i, $1 \leq i \leq p$.

Let us note that the operator of alphabetic primitive recursion defined in [7] may be considered as a partial case of the operator $\boldsymbol{R}$; it is obtained when $k_1 = k_2 = ... = k_p = l+2$.

It is known (see, for example, [7]), that a string function is primitive recursive if and only if it can be obtained from basic string functions using the operators $\boldsymbol{S}$ and $\boldsymbol{R}$. It is easily seen that the operators Sbl, Sbr, Sel, Ser (correspondingly, $\boldsymbol{Sbl}, \boldsymbol{Sbr}, \boldsymbol{Sel}, \boldsymbol{Ser}$) can be expressed by the operator S (correspondingly $\boldsymbol{S}$) using arithmetical basic functions (correspondingly, basic string functions). Similarly the operator $\boldsymbol{R}$ can be expressed using the operator of alphabetic primitive recursion defined in [7], the operator $\boldsymbol{S}$ of superposition, and basic string functions. So, the introducing of the mentioned additional operators is not essential for the describing of the classes of arithmetical primitive recursive functions and primitive recursive string functions (though it is essential for the estimates of complexity of corresponding formal expressions).

Now let us define the formal languages LA and LW for the representation of, correspondingly, arithmetical primitive recursive functions and primitive recursive string functions. Their definitions will be based on the notion of *term*, i.e. formal expression (having a given *dimension*) in the mentioned languages. In the definitions of terms in LA and LW we shall denote by D_i the unary notation of the natural number i; so it is the string $\underbrace{aa...a}_{i\ items}$, where a is a fixed letter in the alphabet of a corresponding language.

The notion of *term* in the language LA is defined inductively as follows.

1. 0 is a 0-dimensional term (this term expresses the 0-dimensional function 0).

2. s is a 1-dimensional term (this term expresses the 1-dimensional function $s(x) = x + 1$).

3. $I(D_k, D_l)$, where $k \geq 1$, $1 \leq l \leq k$, is a k-dimensional term (every such term expresses the k-dimensional function I_l^k; it will be denoted below by $\bar{I}_l^k$).

4. If g is a v-dimensional term, and $h_1, h_2, ..., h_m$, where $m \leq v$, are, correspondingly $k_1-, k_2-, ..., k_m-$ dimensional terms, then $Sbl(g, h_1, h_2, ..., h_m)$, $Sbr(g, h_1, h_2, ..., h_m)$, $Sel(g, h_1, h_2, ..., h_m)$, $Ser(g, h_1, h_2, ..., h_m)$, are $(v - m + k)$-dimensional terms, where $k = max(k_1, k_2, ..., k_m)$ (these terms express arithmetical functions obtained by the corresponding operators of superposition from functions expressed by $g, h_1, h_2, ..., h_m$).

5. If g is an l-dimensional term, h is a k-dimensional term, where $k \geq l+2$, then $R(g, h)$ is a $(k - 1)$-dimensional term (this term expresses an arithmetical function obtained by the operator R from functions expressed by g and h).

6. Some shortened expressions for terms will be introduced below.

There is no term except those obtained by 1-5 and their shortened variants.

The notion of *term* in the language LW is defined inductively as follows.

1. Λ is a 0-dimensional term (this term expresses the 0-dimensional basic function Λ).

2. $S(D_i)$, where $1 \leq i \leq p$, are 1-dimensional terms (every such term expresses the 1-dimensional basic string function $S_i(P)$; it will be denoted below by $\bar{S}_i$).

3. $I(D_k, D_l)$, where $k \geq 1$, $1 \leq l \leq k$, are k-dimensional terms (every such term expresses the k-dimensional basic string function I_l^k; it will be denoted below by $\bar{I}_l^k$.

4. If G is a v-dimensional term, and $H_1, H_2, ..., H_m$, where $m \leq v$, are, correspondingly, $k_1-, k_2-, ..., k_m$-dimensional terms, then $\boldsymbol{Sbl}(G, H_1, H_2, ..., H_m)$, $\boldsymbol{Sbr}(G, H_1, H_2, ..., H_m)$, $\boldsymbol{Sel}(G, H_1, H_2, ..., H_m)$, $\boldsymbol{Ser}(G, H_1, H_2, ..., H_m)$, are $(v-m+k)$-dimensional terms, where $k = max(k_1, k_2...k_m)$ (these terms express string functions obtained by the corresponding operators of superposition from functions expressed by $G, H_1, H_2, ..., H_m$).

5. If G is an l-dimensional term, $H_1, H_2, ..., H_p$ are, correspondingly, $k_1-, k_2 - ...k_p$-dimensional terms, where $k_i \geq 2$, $1 \leq i \leq p$, and $max(k_1, k_2, ..., k_p) \geq l + 2$, then $\boldsymbol{R}(G, H_1, H_2, ..., H_p)$ is a $(k - 1)$-dimensional term, where $k = max(k_1, k_2, ..., k_p)$ (this term expresses the string function obtained by the operator $\boldsymbol{R}$ from $G, H_1, H_2, ..., H_p$).

6. Some shortened expressions for terms will be introduced below.

There is no term except those obtained by 1-5 and their shortened variants.

Let us introduce also the following shortened expressions: S, Sb, Se, Sl, Sr (correspondingly, $\boldsymbol{S}$, $\boldsymbol{Sb}$, $\boldsymbol{Se}$, $\boldsymbol{Sl}$, $\boldsymbol{Sr}$) in the language LA (correspondingly, LW). These expressions are used in LA (correspondingly, LW) in the case when some letters in the expressions Sbl, Sbr, Sel, Ser (correspondingly, $\boldsymbol{Sbl}$, $\boldsymbol{Sbr}$, $\boldsymbol{Sel}$, $\boldsymbol{Ser}$) may be omitted without introducing a misunderstanding in the interpretation of the considered operator of superposition. Indeed, let an operator of superposition be applied to functions $g, h_1, h_2, ..., h_m$ depending, correspondingly, on $v, k_1, k_2, ..., k_m$ variables. Then it is easily seen, that the letters b and e may be omitted in the expression of the considered operator if $m = v$; the letters l and r may be omitted if $k_1 = k_2 = ... = k_m$. Every shortened expression is interpreted in the same way as an expression obtained from it by adding omitted letters.

The *complexity* of a term t in the languages LA and LW is defined as the length $|t|$ of the string t.

Let us note that *the dimension of any term t in LA and LW is less or equal to the length of t*. This statement is easily proved using the induction on the process of constructing of terms in LA or LW.

Now let $t \in LW$ and $r \in LA$ be some terms. We say that the terms t and r *correspond* one to another if the string function expressed by the term t is representable by the arithmetical function expressed by the term r. The sets of terms $LA(t)$ and $LW(r)$ are defined as follows for any $t \in LW$, and any $r \in LA$:

$LA(t) = \{r | r \in LA, \text{ and } r \text{ corresponds to } t\}$;

$LW(r) = \{t | t \in LW, \text{ and } t \text{ corresponds to } r\}$.

Shannon's functions $SH_{WA}(n)$ and $SH_{AW}(n)$ are defined as follows:

$SH_{WA}(n) = \max_{(t \in LW) \& (|t| \leq n)} (\min_{r \in LA(t)} |r|)$;

$SH_{AW}(n) = \max_{(t \in LA) \& (|t| \leq n)} (\min_{r \in LW(t)} |r|)$.

Clearly, these functions describe the comparative complexity of terms functions which correspond one to another in the languages LW and LA. So, they describe, what is the change of complexity when we pass from one language to another and from some function to the function corresponding to it.

Main theorem. For every natural number $p > 1$ (where p is the quantity of letters in the alphabet A) there exist real constants $c_1^p, d_1^p, c_2^p, d_2^p, c_3^p, d_3^p, c_4^p, d_4^p$ such that for every natural number n

$c_1^p n + d_1^p \leq SH_{WA}(n) \leq c_2^p n + d_2^p$;

$c_3^p n + d_3^p \leq SH_{AW}(n) \leq c_4^p n + d_4^p$;

Proof: Let a number $p > 1$ be fixed. Below the upper indexes p in the expressions c_1^p, d_1^p, c_2^p, d_2^p, c_3^p, d_3^p, c_4^p, d_4^p will be omitted. Let us prove the estimate $c_1 n + d_1 = SH_{WA}(n)$. It is sufficient to establish this estimate (as all others) for $n \geq n_0$, where n_0 is some constant. Indeed, if this estimate holds for $n \geq n_0$, then similar estimate in the general case can be obtained by changing the additive constant in the considered inequality.

Let n be any natural number. For any k and any k-tuple of natural numbers $(i_1, i_2, ..., i_k)$, where $1 \leq i_l \leq p$ for $1 \leq l \leq k$, let us consider a term

$$t_{i_1,i_2,...,i_k} = \boldsymbol{S}(\bar{S}_{i_k}, \boldsymbol{S}(\bar{S}_{i_{k-1}} \ldots \boldsymbol{S}(\bar{S}_{i_2}, \boldsymbol{S}(\bar{S}_{i_1}, \wedge))...))$$

(as it is noted above, $\bar{S}_i$ denotes the term $S(D_i)$). Obviously, every term $t_{i_1,i_2,...,i_k}$ expresses in LW a 0-dimensional function such that its value is the string $a_{i_1}, a_{i_2}, ..., a_{i_k}$. It is easily seen that there exist some constants A and B depending only on p such that $|t_{i_1,i_2,...,i_k}| \leq Ak + B$. Now let us take such k that $Ak + B \leq n < A(k+1) + B$ (such a k exists for every $n \geq n_0$, where n_0 is some constant). The quantity of functions expressed in LW by the terms $t_{i_1,i_2,...,i_k}$ is equal to p^k, and all these functions are different. Let us consider the minimal terms $r_{i_1,i_2,...,i_k}$ in LA expressing the arithmetical functions which represent the mentioned functions in LA; obviously $r_{i_1,i_2,...,i_k} \in LA(t_{i_1,i_2,...,i_k})$ and $(t_{i_1,i_2,...,i_k} \in LW) \& (|t_{i_1,i_2,...,i_k}| \leq n)$. Now let us denote the quantity of letters in the alphabet of the language LA by q.

For every natural number w the quantity of strings in LA having the length $\leq w$ is less or equal to $1 + q + q^2 + ... + q^w = \frac{q^{w+1} - 1}{q - 1}$. But the quantity of the terms $r_{i_1,i_2,...,i_k}$ is equal to p^k and all these terms are different. Hence the maximum w of the lengths of these terms satisfies the inequality $\frac{q^{w+1} - 1}{q - 1} \geq p^k$.

Let us denote the value $SH_{WA}(n)$ by m. Using the definition of SH_{WA} we can conclude that $m \geq w$; so, we have $\frac{q^{m+1} - 1}{q \;\; 1} \geq p^k$ and $q^{m+1} > p^k(q-1)$, hence $(m+1)\log q > k \log p + log(q-1)$.

Using the inequality $k > \frac{n - A - B}{A}$, we obtain that $m \log q + \log q > \frac{n - A - B}{A} \log p + log(q-1)$.

So, we have $SH_{WA}(n) = m > c_1 n + d_1$, where $c_1 = \dfrac{n \log p}{A \log q}, d_1 = \dfrac{log(q-1)}{\log q} - \dfrac{\log p}{A \log q}(A+B) - 1$, and the lower estimate for SH_{WA} is proved.

The lower estimate $c_3 n + d_3 \leq SH_{AW}(n)$ is obtained in a similar way. Let us consider 1-dimensional arithmetical functions $f_i(x) = px + i$ for $1 \leq i \leq p$; let r_i be terms expressing these functions in LA. Let n be any natural number. For any natural number k and any k-tuple $(i_1, i_2, ..., i_k)$, where $l \leq i_l \leq p$ for $1 \leq l \leq k$, let us consider the term $r_{i_1,i_2,...,i_k} = S(r_{i_k}, ...S(r_{i_2}, S(r_{i_1}, 0))...)$ in LA. Obviously, the term $r_{i_1,i_2,...,i_k}$; expresses in LA an 0-dimensional arithmetical function such that its value is the alphabetic number of the string $a_{i_1}, a_{i_2}, ..., a_{i_k}$. All these functions are different. It is easily seen that there exist constants A_1 and B_1 depending only on p, such that $|r_{i_1,i_2,...,i_k}| \leq A_1 k + B_1$ for any k-tuple $(i_1, i_2, ..., i_k)$. The quantity of the terms $r_{i_1,i_2,...,i_k}$ is equal to p^k, and all these terms are different. Now by conclusions similar to those, which are used in the proof of the lower estimate for SH_{WA}, we obtain the required inequality for the function SH_{AW}.

Let us prove now the upper estimate $SH_{WA}(n) \leq c_2^n + d_2$. We shall prove this inequality using course-of-values induction on n (cf. [5]). Namely, we shall prove that if c_2 and d_2 satisfy some conditions then this inequality holds for a given n if it holds for all natural numbers less than n.

It is sufficient to prove that for any natural number n and for every term t in LW such that $|t| = n$ there exists a term Φ in LA such that $\Phi \in LA(t)$ and $|\Phi| \leq c_2 n + d_2$.

Let us suppose that $|t| = n$, and the term t is obtained by one of the points 1-3 in the definition of the term in the language LW. Obviously, in this case the required inequality holds for every $c_2 \geq 1$ and for every d_2 which is greater or equal to all the lengths of terms expressing the functions $px + i$ for $1 \leq i \leq p$ in LA. In particular, we can take d_2 as the maximal length of the mentioned terms, and $c_2 = 1$.

Now let us suppose that $|t| = n$, and the term t is obtained using the operator ***Sbl*** by the point 4 in the definition of the term in LW. In this case we have

$t = \boldsymbol{Sbl}(G, H_1, H_2, ..., H_m)$;

$n = |t| = |G| + |H_1| + ... + |H_m| + m + 5.$

Let us consider string functions expressed by the terms $G, H_1, H_2 ... H_m$; let us denote them by $\boldsymbol{G}, \boldsymbol{H_1}, \boldsymbol{H_2}, ..., \boldsymbol{H_m}$. Let us denote arithmetical functions representing these string functions by $g(y_1, y_2 ... y_v), h_1(x_1, x_2 ... x_{k_1}), h_2(x_1, x_2 ... x_{k_2}) h_m(x_1, x_2 ... x_{k_m})$.

Arithmetical function representing the string function F expressed by the term t, we denote by $f(x_1, x_2 ... x_{v-m+k})$, where $k = max(k_1, k_2 ... k_m)$.

Clearly, we have

$$f(x_1, x_2 ... x_{v-m+k}) = g(h_1(x_{k-k_1+1}, x_{k-k_1+2} ... x_k), h_2(x_{k-k_2+1}, x_{k-k_2+2} ... x_k), ...$$
$$h_m(x_{k-k_m+1}, x_{k-k_m+2} ... x_k), x_{k+1}, x_{k+2} ... x_{v-m+k})$$

By induction we conclude that the arithmetical functions $g(y_1, y_2 ... y_v)$, $h_1(x_1, x_2 ... x_{k_1})$, $h_2(x_1, x_2 ... x_{k_2})$ ... $h_m(x_1, x_2 ... x_{k_m})$ can be expressed in the language LA by the terms $r, r_1, r_2 ... r_m$ such that $|r| \leq c_2|G| + d_2$, $|r_1| \leq c_2|H_1| + d_2, ...,$ $|r_m| \leq c_2|H_m| + d_2$.

Arithmetical functionfis expressed in the language LA by the term Φ = $Sbl(r, r_1, r_2 ... r_k)$ the complexity of this term is less or equal to

$$c_2|G| + d_2 + c_2|H_1| + d_2 + ... + c_2|H_m| + d_2 + m + 5$$

It is easily seen that the inequality $Sbl(r, r_1, r_2 ... r_k) \leq c_2 n + d_2$ holds if the positive numbers c_2 and d_2 satisfy the condition $c_2 \geq d_2 + 1$.

The case when the term t is obtained by operators ***Sbr***, ***Sel***, ***Ser*** are considered in the same way.

Let us suppose that $|t| = n$, and the term t in LW is obtained by the point 5 in the definition of the term in LW. In this case we have $t = \boldsymbol{R}(G, H_1, H_2 ... H_p)$, where the dimensions of terms $G, H_1, H_2 ... H_p$ are equal, correspondingly, to $l, k_1, k_2 ... k_p$; these numbers satisfy the conditions $k \geq l + 2$, where $k = max(k_1, k_2 ... k_m)$, and $k_i \geq 2$ for all $i, 1 \leq i \leq p$.

We introduce the notations similar to those introduce in the preceding case. Namely, let us denote by F the string function expressed by the term t in LW; clearly, F is obtained by the operator $\boldsymbol{R}$ from the functions expressed by the terms $G, H_1, H_2 ... H_p$. By $g, h_1, h_2 ... h_p$ we denote arithmetical functions representing the functions $\boldsymbol{G}$,$\boldsymbol{H_1}$, $\boldsymbol{H_2}$,, $\boldsymbol{H_p}$. By induction we conclude that there exist terms $r, r_1, r_2 ... r_p$ in LA expressing the functions $g, h_1, h_2 ... h_p$ and satisfying the conditions

$$|r| \leq c_2|G| + d_2, \ |r_1| \leq c_2|H_1| + d_2, ..., \ |r_m| \leq c_2|H_m| + d_2.$$

By f we denote the arithmetical function representing the function F. Obviously, the function f satisfies the following conditions for all natural numbers $x_1, x_2 ... x_{k-2}, y$:

$$f(x_1, x_2 ... x_{k-2}, 0) = g(x_1, x_2 ... x_l);$$
$$f(x_1, x_2 ... x_{k-2}, py + i) = h_i(x_{k-k_i+1}, x_{k-k_i+2} ... x_{k-2}, y, f(x_1, x_2 ... x_{k-2}, y)),$$

where $1 \leq i \leq p$.

Let us construct a term Φ expressing in LA the function f. We shall use the notations and constructions given in [5]. It is easily seen that the following equalities hold for all natural numbers $x_1, x_2 \dots x_{k-2}, z$:

$f(x_1, x_2 \dots x_{k-2}, 0) = g(x_1, x_2 \dots x_l)$;

$$f(x_1, x_2 \dots x_{k-2}, z+1) = \left\{ \sum_{\lambda=1}^{p-1} \overline{sg}(|rm(z+1,p) - \lambda|) \cdot \right.$$

$$\left. \cdot h_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, [(z+1)/p], f(x_1, x_2 \dots x_{k-2}, [(z+1)/p])) \right\} +$$

$$+ \overline{sg}(rm(z+1,p)) \cdot h_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, [(z+1)/p] \dot{-} 1, f(x_1, x_2 \dots x_{k-2}, [(z+1)/p] \dot{-} 1)).$$

The expressions $a^b, a+b, a \cdot b, \overline{sg}(c) \cdot a + sg(c) \cdot b$ we shall write below sometimes as, correspondingly, $(a \exp b)$, $\mathrm{sum}(a,b)$, $\mathrm{prod}(a,b)$, $\mathrm{cond}(a,b,c)$. Let us consider an auxiliary function f^* defined by the following equality:

$$f^*(x_1, x_2 \dots x_{k-2}, w) = \prod_{z<w} (p_z \exp f(x_1, x_2 \dots x_{k-2}, z)),$$

where p_z is the $(z+1)$-th prime number $(p_0 = 2, p_1 = 3, \dots)$.

Clearly, $f^*(x_1, x_2 \dots x_{k-2}, z) = (f^*(x_1, x_2 \dots x_{k-2}, v))_z$, for every $v > z$, (where $(y)_z$ is the degree of p_z in the expansion of y on prime divisors). We have the following equations of primitive recursion for f^*:

$$f^*(x_1, x_2 \dots x_{k-2}, 0) = \alpha(\,);$$

$$f^*(x_1, x_2 \dots x_{k-2}, z+1) = \beta(x_1, x_2 \dots x_{k-2}, z, f^*(x_1, x_2 \dots x_{k-2}, z)),$$

where α and β are defined as follows:

$$\alpha(\,) = 1;$$

$$\beta(x_1, x_2 \dots x_{k-2}, z, w) = w \cdot (p_z \exp\{cond(g(x_1, x_2 \dots x_l),$$

$$\{\{\sum_{\lambda=1}^{p-1} h_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, [z/p], (w)_{[z/p]}) \cdot \overline{sg}(|rm(z,p) - \lambda|)\} +$$

$$+ h_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, [z/p] \dot{-} 1, (w)_{[z/p] \dot{-} 1}) \cdot \overline{sg}(rm(z,p))\}, z\}).$$

Let us construct terms in LA expressing the functions α, β, f^*, f. The function α is expressed by the following term $\tilde{\alpha}$ (where s is a term expressing the basic function $s(x) = x+1$):

$$\tilde{\alpha} = S(s, 0).$$

For constructing a term $\bar{\beta}$ expressing β we consider auxiliary functions δ_1, δ_2, η_1, η_2, $\sigma_1, \sigma_2 \dots \sigma_p$, $\rho_1, \rho_2 \dots \rho_p$, $\psi_1, \psi_2 \dots \psi_p$, φ_1, φ_2, φ_3, φ_4, θ satisfying the following conditions:

$$\sigma_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, z, w) = h_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, \eta_1(z,w), \eta_2(z,w)),$$

where $1 \le \lambda \le p-1$, $\eta_1(z,w) = [z/p]$, $\eta_2(z,w) = (w)_{[z/p]}$;

$$\sigma_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, z, w) = h_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, \delta_1(z,w), \delta_2(z,w)),$$

where $\delta_1(z,w) = [z/p] \dot{-} 1$, $\delta_2(z,w) = (w)_{[z/p] \dot{-} 1}$;

$$\psi_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, z, w) = \sigma_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, z, w) \cdot \rho_\lambda(z,w),$$

where $1 \le \lambda \le p-1$, $\rho_\lambda(z,w) = \overline{sg}(|rm(z,p) - \lambda|)$;

$$\psi_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, z, w) = \sigma_p(x_{k-k_p+1}, x_{k-k_p+2} \dots x_{k-2}, z, w) \cdot \rho_p(z,w),$$

where $\rho_p(z,w) = \overline{sg}(rm(z,p))$;

$$\varphi_1(x_1, x_2 \dots x_{k-2}, z, w) = \sum_{\lambda=1}^{p} \psi_\lambda(x_{k-k_\lambda+1}, x_{k-k_\lambda+2} \dots x_{k-2}, z, w),$$

$$\varphi_2(x_1, x_2 \dots x_{k-2}, z, w, y) = cond(g(x_1, x_2 \dots x_l), \varphi_1(x_1, x_2 \dots x_{k-2}, z, w), y),$$

$$\varphi_3(x_1, x_2 \dots x_{k-2}, z, w) = \varphi_2(x_1, x_2 \dots x_{k-2}, I_1^2(z,w), I_2^2(z,w), I_1^2(z,w)),$$

$$\varphi_4(x_1, x_2 \dots x_{k-2}, z, w) = \theta(\varphi_3(x_1, x_2 \dots x_{k-2}, z, w), I_1^2(z,w), I_2^2(z,w)),$$

where $\theta(y, z, w) = (p_z \exp y)$;

$$\beta(x_1, x_2 \dots x_{k-2}, z, w) = w \cdot \varphi_4(x_1, x_2 \dots x_{k-2}, z, w).$$

Clearly, the functions sum, prod, cond, θ can be expressed in LA by the terms $\bar{sum}$, $\bar{prod}$, $\bar{cond}$, $\bar{\theta}$ having constant complexities; the functions δ_1, δ_2, η_1, η_2, ρ_1, $\rho_2 \dots \rho_p$ can be expressed in LA by the terms $\bar{\delta}_1$, $\bar{\delta}_2$, $\bar{\eta}_1$, $\bar{\eta}_2$, $\bar{\rho}_1$, $\bar{\rho}_2 \dots \bar{\rho}_p$ having constant complexities depending only on p. The term $\bar{\beta}$ expressing β is obtained by following constructions (where by $\bar{\sigma}_1, \bar{\sigma}_2 \dots \bar{\sigma}_p$, $\bar{\psi}_1, \bar{\psi}_2 \dots \bar{\psi}_p$, $\bar{\varphi}_1, \bar{\varphi}_2, \bar{\varphi}_3, \bar{\varphi}_4$ we denote terms expressing the functions $\sigma_1, \sigma_2 \dots \sigma_p$, $\psi_1, \psi_2 \dots \psi_p$, $\varphi_1, \varphi_2, \varphi_3, \varphi_4$):

$\bar{\sigma}_\lambda = Se(r_\lambda, \bar{\eta}_1, \bar{\eta}_2)$, where $1 \le \lambda \le p-1$,

$\bar{\sigma}_p = Se(r_p, \bar{\delta}_1, \bar{\delta}_2)$,

$\bar{\psi}_\lambda = Sl(\bar{p}rod, \bar{\sigma}_\lambda, \bar{\rho}_\lambda)$, where $1 \leq \lambda \leq p-1$,

$\bar{\psi}_p = Sl(\bar{p}rod, \bar{\sigma}_p, \bar{\rho}_p)$,

$\bar{\varphi}_1 = Sl(\bar{s}um, \bar{\psi}_1, Sl(\bar{s}um, \bar{\psi}_2 \ldots Sl(\bar{s}um, \bar{\psi}_{p-1}, \bar{\psi}_p) \ldots))$,

$\bar{\varphi}_2 = Sbr(\bar{c}ond, \tau, \bar{\varphi}_1)$,

$\bar{\varphi}_3 = Se(\bar{\varphi}_2, \bar{I}^2_1, \bar{I}^2_2, \bar{I}^2_1)$,

$\bar{\varphi}_4 = Sl(\bar{\theta}, \bar{\varphi}_3, \bar{I}^2_1, \bar{I}^2_2)$,

$\bar{\beta} = Sl(\bar{p}rod, \bar{\varphi}_1, \bar{I}^1_1)$,

Now the term Ω expressing f^* is obtained as $R(\bar{\alpha}, \bar{\beta})$. The term Φ expressing f is obtained using the following equality:

$$f(x_1, x_2 \ldots x_{k-2}, z) = (f^*(x_1, x_2 \ldots x_{k-2}, z+1))_z$$

by the following construction:

$\Phi = Sl(\bar{\tau}, Sel(\Omega, s), \bar{I}^1_1)$,

where 2-dimensional term $\bar{\tau}$ expresses a function τ such that $\tau(y, z) = (y)_z$.

We can conclude considering the structure of the term Φ that the following equality holds: $|\Phi| = |r| + |r_1| + |r_2| + \ldots + |r_p| + a$, where a is a constant depending only on p. It can be easily proved that the condition $c_2 \geq d_2 + a$ is sufficient for the considered step of induction. Now if we take d_2 as the maximal length of terms expressing in LA the functions $px + i$ for $1 \leq i \leq p$, and define c_2 as $d_2 + a + 1$, then the step of induction is ensured in all cases. So, the estimate $SH_{WA} \leq c_2 n + d_2$ is proved.

Now let us prove the estimate $SH_{AW} \leq c_4 n + d_4$. We shall prove three lemmas for the establishing of this estimate. Let us introduce a function ν such that for every natural number n (cf. [7]):

$$\nu(n) = a_1^n = \underbrace{a_1 a_1 \ldots a_1}_{n\ times};$$

$$\nu(0) = a_1^0 = \Lambda,$$

where a_1 is the first letter in A.

Lemma 1. (cf. [7],p. 216, lemma 1) Constants c' and d' can be found such that for every term $t \in LA$ there exists a term $\Phi \in LW$ satisfying the following conditions:

(1) for all natural numbers $x_1, x_2 \ldots x_m$ the equality

$\varphi(\nu(x_1), \nu(x_2) \ldots \nu(x_m))) = \nu(\tau(x_1, x_2 \ldots x_m))$

holds for the functions φ and τ expressed, correspondingly, by the terms Φ and t;

(2) $|\Phi| \leq c'|t| + d'$.

The proof is similar to that of the estimate $SH_{AW} \leq c_3 n + d_3$. If t is one of the basic terms 0, $\bar{I}^k_n$, s, then the term Φ can be constructed as $\wedge$, $\bar{I}^k_n$, $\bar{S}_1$, in this case every $c' \geq 1$ and $d' \geq 1$ satisfy the conditions of lemma. If t is obtained using the operator Sbl, Sbr, Sel or Ser, then Φ is obtained in a similar way using the operator, correspondingly $\boldsymbol{Sbl}$, $\boldsymbol{Sbr}$, $\boldsymbol{Sel}$ or $\boldsymbol{Ser}$. Finally, if t is obtained using the operator R as $R(t_1, t_2)$ then the corresponding term Φ is constructed as $\boldsymbol{R}(\Phi_1, \Phi_2, \underbrace{\bar{\varepsilon}, \bar{\varepsilon}, \ldots, \bar{\varepsilon}}_{(p-1)\ times})$, where Φ_1 and Φ_2 are terms in LW obtained by induction for, correspondingly, t_1 and t_2; $\bar{\varepsilon}$ is a two-dimensional term $\boldsymbol{Sb}(\bar{I}^4_1, \wedge, \wedge)$ (this term expresses the function $\varepsilon(x, y) = \wedge$). It is easily seen that

$$|\Phi| = |\Phi_1| + |\Phi_2| + a$$

where a is a constant depending only on p. The remaining details of the proof are similar to the proof of the estimate $SH_{AW} \leq c_3 n + d_3$.

Lemma 2. There exists a primitive recursive string function G such that $G(\nu(m)) = \alpha m$ for every natural number m.

The proof is given in [7] (see [7], p. 217). Indeed, if ψ is a string function defined by the following alphabetic primitive recursion:

$$\psi(\wedge) = a_1;$$

$$\psi(Qa_i) = Qa_{i+1}, \text{ where } 1 \leq i \leq p-1;$$

$$\psi(Qa_p) = \psi(Q)a_1,$$

then G is given by the alphabetic primitive recursion:

$$G(\wedge) = \wedge;$$

$$G(Qa_i) = G(Q), \text{ where } 2 \leq i \leq p;$$

$$G(Qa_1) = \psi(G(Q)).$$

Lemma 3. The one-dimensional string function $\gamma(Q) = \nu(\pi(Q))$ is primitive recursive.

The proof is given in [7] (see [7], p. 217). Indeed, using lemma 1 we obtain that there exist one-dimensional primitive recursive string functions $\omega_1, \omega_2, \ldots, \omega_p$ such that $\omega_i(\nu x) = \nu(px+i)$ for $1 \le i \le p$ and for any natural number x. The function γ is obtained from them by alphabetic primitive recursion:

$$\gamma(\wedge) = \wedge;$$

$$\gamma(Qa_i) = \omega_i(\gamma(Q)), \text{ where } 1 \le i \le p.$$

Now let us return to the estimate of $SH_{AW}(n)$. The inequality $SH_{AW} \le c_4 n + d_4$ will be established for some constants c_4 and d_4 if for every term $t \in LA$ we shall be able to construct a term $\Omega \in LW$ such that $\Omega \in LW(t)$ and $|\Omega| \le c_4|t| + d_4$.

Let $t \in LA$ be any term. Arithmetical function expressed by t let us denote by $\tau(x_1, x_2, \ldots, x_m)$; the string function represented by τ let us denote by $\psi(Q_1, Q_2 \ldots Q_m)$. It is sufficient to construct a term Ω expressing ψ in the language LW and such that $|\Omega| \le c_4|t| + d_4$, where c_4 and d_4 are some constants.

Using lemma 1 let us construct the string function φ such that $\varphi(\nu(x_1), \nu(x_2), \ldots, \nu(x_m)) = \nu(\tau(x_1, x_2 \ldots x_m))$ for any natural numbers $x_1, x_2 \ldots x_m$, and the function φ is expressed in LW by a term Φ satisfying the inequality $|\Phi| \le c'|t| + d'$. Using lemmas 1 and 2 we have

$$\psi(Q_1, Q_2 \ldots Q_m) = \alpha\tau(\pi(Q_1), \pi(Q_2) \ldots \pi(Q_m)) =$$

$$= G(\nu(\tau(\pi(Q_1), \pi(Q_2) \ldots \pi(Q_m)))) =$$

$$= G(\varphi(\nu(\pi(Q_1)), \nu(\pi(Q_2)) \ldots \nu(\pi(Q_m)))) =$$

$$= G(\varphi(\gamma(Q_1), \gamma(Q_2) \ldots \gamma(Q_m))).$$

The functions G and γ are fixed primitive recursive string functions. The function ψ is expressed in LW by the term

$$\Omega = S(\tilde{G}, S(\Phi, \underbrace{\tilde{\gamma}, \tilde{\gamma}, \ldots, \tilde{\gamma}}_{m\ times})),$$

where $\tilde{G}$ and $\tilde{\gamma}$ are terms in LW expressing G and γ. But $|\tilde{G}|$ and $|\tilde{\gamma}|$ are constants depending only on p, the inequality $|\Phi| \le c'|t| + d'$ holds, and the quantity of terms $|\tilde{\gamma}|$ in Ω is less or equal to $|t|$ (because the dimension of every term is less or equal to its length). Hence $|\Phi| \leq c_4|t| + d_4$, where c_4 and d_4 are constants depending only on p. This completes the proof of main theorem.

Theorems similar to main theorem of this paper are considered in [2] and [4]. However, the definitions of the languages LA and LW used in this paper are different from those given in [2] and [4]. The definitions given above are more convenient for further generalizations, in particular, for investigations of analogous problems connected with A. Grzegorczyk's classification of primitive recursive functions ([1], [3]).

ACKNOWLEDGEMENTS

The author is grateful to A. P. Beltiukov and A. O. Slissenko for useful discussions, and to P. Cegielski and I. D. Zaslavsky for the help in the formulation of the final version of this paper.

REFERENCES

1. Grzegorczyk A., Some classes of recursive functions. Rozprawy Matematyczne, N4, pp. 1-46 (1953).

2. Khachatryan M. H., On the systems of primitive recursive arithmetic based on the word recursion. Proc. of the Conference, "Computer Science and Information Technologies", Sept. 22-26, 2003, Yerevan, Armenia, CSIT-03, pp. 88-89 (2003) (in Russian).

3. Mikael Khachatryan. On the complexity of the representation of functions obtained by the word recursion. Intern. Conference "23rd Days of Weak Arithmetics", June 2-5, 2004, Yerevan, Armenia, Abstracts, p. 5 (2004).

4. Khachatryan M. H., The complexity of representation of the functions in the language of alphabetically recursive functions. Hayastani Chartaragitakan Akademiai Lraber, vol 1, N3, pp. 400-413 (2004) (in Armenian).

5. Kleene S.C., Introduction to Metamathematics. D. van Nostrand Comp., Inc., New York-Toronto (1952).

6. Lewis H. R., Papadimitriou C. H., Elements of the theory of Computation. Prentice-Hall, Upper Saddle River, New Jersey (1998).

7. Maltsev A. I., Algorithms and Recursive Functions. 2nd edition, Moscow, "Nauka" (1986) (in Russian).

17 January, 2006

Institute for Informatics and Automation Problems
National Academy of Sciences of Armenia
E-mail: mkhachatryan@seua.am