KYBERNETIKA — VOLUME 15 (1979), NUMBER 2

Memory Complexity of Countable Functions

MILOSLAV NEKVINDA

In the article, countable functions are studied with respect to memory requirements. The obtained results have an immediate relation to the classification of machine-operators which are computable in real time.

1. INTRODUCTION

The basic theorems on classification of operators real time computable with respect to memory complexity were formulated in [5]. It was shown that the classification hierarchy is a consequence of existence of sufficiently rich class of simple i-functions. In this article, theorems on construction of countable functions are given, the main attention being concentrated on needed memory at their generating. As a consequence we obtain analogical theorems for i-functions. These questions were reffered by author on Symposium MFCS (1973), see [6]. The countable functions were carefully being studied yet in [2], [3], no attention, however, being paid to needed memory. A little other approach we see in [4], where the main interest is concentrated upon needed time.

In the following, N denotes the set of natural numbers, N_0 the set of nonnegative integers and R the set of real numbers. Recall briefly some notions used in [5]. A Y-automaton (automaton of Yamada's type) is a many-tape autonomous automaton (i.e., without input) with output alphabet $\Pi = \{0, 1\}$. Every Y-automaton A generates thus an infinite sequence

$$(1.1) \qquad \alpha \equiv \alpha_1\alpha_2 \ldots \alpha_n \ldots, \quad \alpha_i \in \{0, 1\}, \quad i \in N,$$

where the symbol α_i appears on the output in i-th tact, $i \in N$. Let α be any sequence of form (1.1) with infinite number of 1's. Then, we order to it a function $f = F(\alpha)$: : $N \to N$ defined by

$$(1.2) \qquad f(n) = \min \{p; \, p \in N, \sum_{i=1}^{p} \alpha_i = n\}\,, \quad n \in N\,,$$

and a function $\varphi = F_1(\alpha) : N \to N_0$ defined by

$$(1.3) \qquad \varphi(n) = \sum_{i=1}^{n} \alpha_i\,, \quad n \in N\,.$$

We call a function $f : N \to N$ countable, if there exists a Y-automaton A which generates a sequence α of form (1.1) such that $f = F(\alpha)$. If f is countable, then the related function $\varphi = F_1(F^{-1}(f))$ we call an i-function. In this case, we shall also say that the automaton A generates the function f as well as the function φ, respectively. We denote by $\boldsymbol{C}$ the class of all countable functions and by $\boldsymbol{I}$ the corresponding class of all i-functions. Any nonnegative nondecreasing function $L: N \to R$ we shall call a complexity function. We shall say that an automaton A works with space limitation L if for almost all $n \in N$ (i.e., for all natural numbers from a certain one), the number of cells which A needs on any its tape during the first n tacts, is less or equal to $L(n)$. We say that a function $f \in \boldsymbol{C}$ is generable with space limitation L if there is a Y-automaton A which generates f and works with space limitation L. The class of all such functions we denote by $\boldsymbol{C}(L)$. Since we use the same terminology in the case of i-functions, we denote by $\boldsymbol{I}(L)$ the class of all i-functions which correspond to those of $\boldsymbol{C}(L)$.

2. CONSTRUCTIONS OF COUNTABLE FUNCTIONS

This part contains theorems of constructing character which show how to construct another countable function in the case some of countable functions are given. The special attention is paid to the memory needed by generating automata. Theorems 2.1 – 2.5 are proved yet in [2], [3] without attention to needed memory. The given proofs, however, can be modified for our purpose. Thus, because of illustration, we shall present only the proofs of theorems 2.1 and 2.5.

We begin with two following lemmas which serve as a technical means at proving.

Lemma 2.1. There exists a one-tape automaton A which codes nonnegative integers in such a way that

1. the length of code of number n is at most $\log n$ ($\log = \log_2$) for almost all $n \in N$;

2. from the code of $n \in N$, A can transfer to code of number $n - 1$ or $n + 1$ exactly in one tact. (The demand of transfering is supposed to be given by the input of A.)

Proof. Obviously, the starting point of the proof is the binary coding of nonnegative integers. Unfortunately, it is not possible to transfer from the binary code of number n to the binary code of number $n + 1$ in one tact only. To avoid the difficulty, each number $n \in N_0$ will be coded by the configuration (i.e., instantaneous description), in which a one-tape automaton coding subsequently all nonnegative integers transfer in n-th tact. We can, e.g., choose a sequence of configurations in the following way:

$$p\mathbf{B},\quad z\mathbf{1},\quad p\mathbf{1},\quad q\mathbf{0},\quad p\mathbf{B}0,\quad z\mathbf{1}0,\quad z1\mathbf{0},\quad p1\mathbf{0},\quad z1\mathbf{1},$$
$$p1\mathbf{1},\quad q1\mathbf{0},\quad p\mathbf{1}1,\quad q\mathbf{0}0,\quad p\mathbf{B}00,\quad z\mathbf{1}00,\quad z1\mathbf{0}0,$$
$$z10\mathbf{0},\quad p10\mathbf{0},\quad z10\mathbf{1},\quad p10\mathbf{1},\quad q10\mathbf{0},\quad p1\mathbf{0}0,\quad \dots .$$

In the sequence, B denotes a blanc symbol, 0, 1 are digits of the binary system and bold symbols show the position of the read-write head. The symbols p, q, z are internal states of the automaton, their meaning being as follows: p – the number 1 shall be added at bold type place in the next tact; q – the number 1 shall be transfered into the higher order (to the left); the symbol z denotes that a binary code of some integer m has been already made, so that the automaton seeks for the right end of the code in order to manage the binary code of the following integer $m + 1$.

Of course, such an automaton must be provided, in addition, with the ability of recognizing the leftmost as well as the rightmost symbols of the written word. Obviously, the proposed automaton fulfils the conditions of our lemma.

Lemma 2.2. Let A be a Y-automaton which works with space limitation L, and let $r \in N$. Then there exists an automaton B such that

1. *B works r-times faster than A; the output symbols of B are, of course, r-tuples of output symbols of A;*

2. *B works with space limitation L_1, where $L_1(n) = L(rn)$ for almost all $n \in N$.*

Proof. The first assertion of the lemma is well-known, see, e.g., [2]. A rather mote detailed analysis of the proof shows that B may be construct in such a way that the second condition is also true.

Theorem 2.1. Let L_1, L_2 be two complexity functions and let $f_i \in C(L_i)$, $i = 1, 2$. Define f by

$$f(n) = f_1(n) + f_2(n), \quad n \in N .$$

Then $f \in C(L)$, where

$$L(n) = \max(L_1(n), L_2(n)), \quad n \in N .$$

Proof. Let A_i be a Y-automaton which generates f_i with space limitation L_i, $i = 1, 2$. Now, we shall construct a compound automaton A by appropriate composition of A_1, A_2 (the details on construction we shall omit here as well as in all following proofs, they follow from what A is to act) which is to act as follows: At first, e.g.,

A_1 will work until it prints its first 1. Then A_1 stops and A_2 begins to work. Whenever A_2 prints its first 1 it stops and A_1 will continue, in the same configuration which it had reached before. As soon as A_1 prints its following 1 its stops and A_2 will run again. In this manner, A_1 will always interchange with A_2 in acting. In addition, A will print 1's just in tacts in which A_2 will. Clearly, A generates f. To complete the proof we estimate the space limitation of A. Assume that A is performing n-th tact. Then A_i is in k_i-th configuration, $i = 1, 2$, where $k_1 + k_2 = n$. Using the monotony of functions L_i, we have $L_i(k_i) \leqq L_i(n)$, $i = 1, 2$. Thus A works with space limitation L, where $L(n) = \max(L_1(n), L_2(n))$, $n \in N$.

Theorem 2.2. Let L_1, L_2 be two complexity functions and let $f_i \in C(L_i)$, $i = 1, 2$. Define f by

$$f(n) = f_2(n) - f_1(n), \quad n \in N.$$

Suppose that

1. f is increasing, $f(1) \geqq 1$.
2. There are constants $C > 1$ and $m \in N$ such that

$$f_2(n + m) - f_2(n) \geqq C(f_1(n + m) - f_1(n)), \quad n \in N.$$

Let $r \in N$ be such that $r(C - 1) > C$.
Then $f \in C(L)$, where

$$L(n) = \max(L_1(rn), L_2(rn), \log n), \quad n \in N.$$

Proof. By modification of [2].

Theorem 2.3. Let L be a complexity function and let $f \in C(L)$. Define g by

$$g(n) = \sum_{i=1}^{n} f(i), \quad n \in N.$$

Then $f \in C(L_1)$, where

$$L_1(n) = \max(L(n), \log n), \quad n \in N.$$

Proof. By modification of [2].

Theorem 2.4. Let L_1, L_2 be two complexity functions and let $f_i \in C(L_i)$, $i = 1. 2$. Define f by

$$f(n) = f_1(n) f_2(n), \quad n \in N.$$

Then $f \in C(L)$, where

$$L(n) = \max(L_1(n), L_2(n), \log n), \quad n \in N.$$

Proof. By modification of [2].

Theorem 2.5. Let L_1, L_2 be two complexity functions and let $f_i \in C(L_i)$, $i = 1, 2$. Let φ_2 denotes the i-function related to f_2. i.e., $\varphi_2 = F_1(F^{-1}(f_2))$. Define f by

$$f(n) = f_2(f_1(n)), \quad n \in N .$$

Then $f \in C(L)$, where

$$L(n) = \max(L_1(\varphi_2(n)), \quad L_2(n)), \quad n \in N .$$

Proof. We shall construct a Y-automaton A that will generate f. Let A_1, A_2 be a Y-automaton which generates f_1, f_2 with space limitation L_1, L_2, respectively. A will consist of A_1 and A_2, the organization of its program being as follows: A_2 will run continuously without stopping generationg thus the function f_2 (i.e., the related binary sequence). A_1 will perform just one tact of its own program whenever A_2 prints 1, the remaining tacts being stopped. Moreover, A will print 1's just in tacts in which A_1 will. Obviously, A fulfils all conditions of the theorem.

Remark. Using obvious relation $\varphi_2(n) \leqq n$, $n \in N$ we obtain for L a more simple estimate

$$L(n) = \max(L_1(n), L_2(n)), \quad n \in N .$$

Before the next theorem, we introduce the following notations. Let $f : N \to N$, $g : N \to N$ be any two functions. Setting $f(0) = 0$, $g(0) = 0$ we define functions $u : N_0 \to N$, $v : N_0 \to N$ by

$$u(n + 1) = \min(f(n + 1) - f(n); \ g(n + 1) - g(n)), \quad n \in N_0 ;$$
$$v(n + 1) = \max(f(n + 1) - f(n); g(n + 1) - g(n)), \quad n \in N_0 .$$

In addition, we define functions $U : N_0 \to N_0$, $V : N_0 \to N_0$ by

$$(2.1) \qquad U(n) = \sum_{i=1}^{n} u(i), \quad n \in N, \quad U(0) = 0 ;$$
$$V(n) = \sum_{i=1}^{n} v(i), \quad n \in N, \quad V(0) = 0 .$$

Theorem 2.6. Let L_1, L_2, be two complexity functions, let $f \in C(L_1)$, $g \in C(L_2)$. Using (2.1) we define a function $r : N \to N$ as follows:

1. $r(U(n)) = V(n)$, $n \in N$;
2. $r(U(n) + i) = V(n) + i$, $i = 1, 2, \ldots, u(n + 1) - 1$; $n \in N_0$.

Then $r \in C(L)$, where L is a complexity function defined by

$$L(n) = \max(L_1(n), L_2(n)), \quad n \in N .$$

Proof. Let A_1, A_2 be a Y-automaton generating f, g with space limitation L_1, L_2 respectively. We construct A as a compound automaton consisting of A_1, A_2. We divide its acting into stages, the n-th stage, $n \in N$, being related to tacts $V(n-1)+1$, $V(n-1)+2, \ldots, V(n)$. Suppose that, at the beginning of n-th stage, A_1 occurs in its own $f(n-1)$-th configuration (just after having printed its $(n-1)$-th 1) and that, analogously, A_2 occurs in its own $g(n-1)$-th configuration. Now, both A_1 and A_2 will run independently (but synchronously) until one of them prints its following 1. In all these tacts except the last one A prints 1's on its output. In the following tacts, this of A_1, A_2 which had just printed its 1 stops waiting for the moment in which the other prints also its next own 1. In all these tacts but the last one A prints 0's. In the last tact of this time interval A prints 1 having ended the n-th stage. Clearly, A generates the function r with desirable space limitation.

Remark. In order to indicate the dependence on functions f, g we write $r = r(f, g)$. Define e, the identity function by $e(n) = n$, $n \in N$. It is easy to see that

1. $r(f, f) = e$;
2. $r(f, g) = r(g, f)$;
3. $r(f, e) = f$.

Now, we shall state a special important case of previous theorem.

Theorem 2.7. Let f, g be two countable functions such that

$$f(n+1) - f(n) \geqq g(n+1) - g(n), \quad n \in N.$$

Then the function $r = r(f, g)$ (also countable) fulfils the following conditions:

1. $r(g(n)) = f(n), \quad n \in N$;
2. $r(g(n) + i) = f(n) + i, \quad i = 1, 2, \ldots, g(n+1) - g(n) - 1; \quad n \in N_0$.

Proof. It follows immediately from the previous theorem. Note that, in this case, the function $r = r(f, g)$ is, roughly speaking, equal to the function $f(\psi(.))$, where ψ is the i-function related to g. More precisely, they have the same values at points $g(n)$, $n \in N$.

In previous theorems we constructed new countable functions from given ones using algebraical or other operations. In further theorem we shall operate with functions that are inverse to countable functions, i.e., with i-functions.

Definition. Let φ be an i-function. We shall call it almost concave, if there is a constant $C (C > 1)$ such that for almost all $n \in N$

$$\varphi(2n) \leqq C\,\varphi(n).$$

Theorem 2.8. Let L_1, L_2 be two complexity functions. Let $\varphi \in I(L_1)$, $\psi \in I(L_2)$ and let φ, ψ be almost concave. Let for almost all $n \in N$

$$\varphi(n)\,\psi(n) \leqq n.$$

Define the complexity function L by

$$L(n) = \max\left(L_1(n), L_2(n), \log n\right), \quad n \in N .$$

Then there exists an i-function $\eta \in I(L)$ such that

1. For almost all $n \in N$

$$C_1\, \varphi(n)\, \psi(n) \leqq \eta(n) \leqq C_2\, \varphi(n)\, \psi(n),$$

where $C_1 > 0$, $C_2 > 0$ are certain constants;

2. η is almost concave.

Proof. I. By hypotheses of the theorem, a constant $C > 1$ (common for both φ and ψ) and $\hat{n} \in N$ exist such that

$$(2.2) \qquad \begin{gathered} \varphi(n)\, \psi(n) \leqq n, \\ \varphi(2n) \leqq C\, \varphi(n), \quad \psi(2n) \leqq C\, \psi(n), \quad n \geqq \hat{n}. \end{gathered}$$

Define a sequence $n_1, n_2, \ldots$ of positive integers by

$$(2.3) \qquad n_1 = 4\hat{n}; \quad n_i = 2^{i-1} n_1, \quad i \in N .$$

Let $\eta : N \to N_0$ be any nondecreasing function (we do not suppose it to be an i-function) such that

$$(2.4) \qquad \eta(n_i) = \varphi(n_i/4)\, \psi(n_i/4), \quad i \in N .$$

We shall show that, under this assumption, η fulfils both assertions of the theorem. Let $k \in N$, $k > n_1$. Take $i \in N$ such that $n_i + 1 \leqq k \leqq n_{i+1}$. Since each n_i in the sequence (2.3) is an integral multiple of 4, there is $n \in N$ such that $n_i = 4n$. Hence

$$4n + 1 \leqq k \leqq 8n .$$

Since η is nondecreasing, then, using (2.4), we have $\eta(k) \leqq \eta(8n) = \varphi(2n)\, \psi(2n)$. Since φ, ψ are also nondecreasing, we obtain $\varphi(2n)\, \psi(2n) \leqq \varphi(4n+1)\, \psi(4n+1) \leqq \leqq \varphi(k)\, \psi(k)$, which gives

$$(2.5) \qquad \eta(k) \leqq \varphi(k)\, \psi(k) .$$

Now, with respect to (2.2), we have $\varphi(n) \geqq C^{-3}\, \varphi(8n)$, $\psi(n) \geqq C^{-3}\, \psi(8n)$, so that $C^{-3}\, \varphi(8n)\, \psi(8n) \leqq \psi(n)\, \psi(n) = \eta(4n) \leqq \eta(k)$. Since $\varphi(k)\, \psi(k) \leqq \varphi(8n)\, \psi(8n)$, we thus obtain

$$(2.6) \qquad C^{-6}\, \varphi(k)\, \psi(k) \leqq \eta(k) .$$

Obviously, the relations (2.5), (2.6) prove the first assertion of the theorem. In addition, applying (2.2), (2.5) and (2.6) we have

$$\eta(2k) \leqq \varphi(2k)\,\psi(2k) \leqq C^2\,\varphi(k)\,\psi(k) \leqq C^8\,\eta(k)\,,$$

which proves that η is almost concave.

II. In the following, we shall try to construct an automaton A which generates an i-function η with the property (2.4). According to the previous part of proof, η will then fulfil both assertions of our theorem. We divide the acting of A into stages. The preliminary stage is given by tacts $1, 2, \ldots, n_1$, the i-th stage as a time interval from tact $n_i + 1$ to $2n_i = n_{i+1}$, $i \in N$. As we have mentioned above, for a given $i \in N$, there is an integer $n \in N$ such that $n_i = 4n$. The i-th stage is thus given as a time interval from $(4n + 1)$-th to $8n$-th tacts. Obviously

$$\begin{aligned}(2.7)\qquad & \varphi(2n)\,\psi(2n) - \varphi(n)\,\psi(n) = \\ & = \varphi(2n)\,(\psi(2n) - \psi(n)) + \psi(n)\,(\varphi(2n) - \varphi(n))\,.\end{aligned}$$

Denoting $d_n = \varphi(2n)\,\psi(2n) - \varphi(n)\,\psi(n)$, $e_n = \varphi(2n)\,(\psi(2n) - \psi(n))$, $f_n = \psi(n)\,.$ $.\,(\varphi(2n) - \varphi(n))$, we can rewrite (2.7) in the form

$$(2.8)\qquad d_n = e_n + f_n\,.$$

Because of definition of η (see (2.4)), A must print d_n 1's during the investigated i-th stage. Using (2.2) we have $d_n \leqq \varphi(2n)\,\psi(2n) \leqq 2n$ and, consequently

$$(2.9)\qquad e_n \leqq 2n\,,\quad f_n \leqq 2n\,.$$

Thus, we suggest a construction of A as a compound automaton consisting of three blocks B_1, B_2, D, with the following properties. According to (2.8), (2.9), the block B_1 will produce e_n 1's during the interval $\langle 4n + 1, 6n\rangle$. Analogously, B_2 will produce f_n 1's during the interval $\langle 6n + 1, 8n\rangle$. The block D is a device for signalizing the $4n$-th, $6n$-th and $8n$-th tacts, i.e. the beginning, middle and the end of each stage.

III. The construction of D. D will consist of three tapes, say T_1, T_2, T_3. Suppose that, at the beginning of the stage, the number $2n$ is encoded on tape T_1 in the sence of Lemma 2.1 and the tapes T_2, T_3 are empty. During the interval $\langle 4n + 1, 6n\rangle$, subtracting subsequently 1's, the code will be erased on tape T_1 being again reproduced, by adding 1's both on T_2 and T_3. In following $2n$ tacts, the tape T_2 will be erased by subtracting 1's per tact. At the same time, by adding 1's per tact, the code of number $4n$ will be formed on tape T_3. Thus, at the end of the stage, the tapes T_1, T_2 are empty and T_3 contains the code of number $4n$. Hence, all three tapes are ready for acting in the next stage, the roles of tapes T_1, T_3 being changed to each other. Obviously, D is able to signalize the needed tacts.

IV. Now, we describe a construction of B_1. Let F, G be automata generating the i-functions φ, ψ with space limitation L_1, L_2, respectively. B_1 will consist of two copies F_1, F_2 of F, one copy of G and five additional tapes $T_1, T_2. \ldots, T_5$. Suppose that, at the beginning of the stage, the following situation occurs:

(i) F_1 is in its $2n$-th configuration (this means that F_1 had just performed $2n$ tacts of its own program), F_2 is in initial configuration (with empty tapes) and G is in its $2n$-th configuration;

(ii) The number $\varphi(2n)$ is logarithmically encoded (i.e., in the sence of Lemma 2.1) on tape T_1, the tapes T_2, T_3 are empty;

(iii) the number $\psi(2n) - \psi(n)$ is logarithmically encoded on tape T_4, the tape T_5 is empty.

As we know, B_1 has to print $e_n = \varphi(2n)(\psi(2n) - \psi(n))$ of 1's during the time interval $\langle 4n + 1, 6n \rangle$. This may be done in $\psi(2n) - \psi(n)$ intervals, each of them will take just $\varphi(2n)$ tacts. During the first interval, by subtracting 1's per tact, the code of $\varphi(2n)$ will be erased on tape T_1, being at the same time reproduced on T_2. The end of this interval will be marked on T_4 by subtracting 1. The acting in the second interval proceeds in the same way as in the previous one, the role of T_1 being interchanged with T_2. This process passes again and again, until the tape T_4 ocurs empty. At this moment, the needed number of e_n tacts are just counted out.

V. Besides counting e_n tacts B_1 will have to get ready for acting in the next stage. So, F_1 will transfer into the initial configuration (all tapes will be erased). Since F_1 is, at the beginning of the stage, in $2n$-th configuration, the length of words on each its tape is at most $2n$. Hence, every tape of F_1 can be erased in $4n$ tacts, i.e., in time. The automaton F_2 will run in all tacts of the stage, so that it will print $\varphi(4n)$ 1's which will be subsequently encoded on tape T_3.

The automaton G will perform just $2n$ tacts of its own program (in the first half of the stage or in the second one, it does not matter), producing thus $\varphi(4n) - \varphi(2n)$ 1's which will be encoded on T_5.

At last, both T_1 and T_2 will become empty during the second half of the stage (as a matter of fact, one of them will be already empty at $6n$-th tact).

Thus, B_2 is able to get ready for acting in the next stage in time. We shall not describe the construction of the block B_2, for it may be done in analogical way.

Summarizing we see that the i-function η generated by A fulfils the needed condition (2.4).

VI. It remains to estimate the needed memory space. Consider again the time interval $\langle 4n + 1, 8n \rangle$. Let $k \in N$, $4n + 1 \leqq k \leqq 8n$. Since numbers coded on tapes of D equal at most $4n$, the number of needed cells on each tape of D is at most $\log 4n \leqq \log k$. Because of inequalities $\varphi(j) \leqq j$, $\psi(j) \leqq j$, $j \in N$, the same estimate is true for tapes $T_1 - T_5$ of B_1.

Since each copy of automata F, G in block B_1 as well as in B_2 reaches in this time interval at most its $4n$-th configuration, the number of needed cells on every its tape is bounded by $\max(L_1(k), L_2(k))$.

On the whole, the space limitation L of A fulfils the desirable condition which completes the proof.

Theorem 2.9. Let $p \in N$, let L_j be a complexity function, let $\varphi_j \in \boldsymbol{I}(L_j)$ be almost concave, $j = 1, 2, \ldots, p$. Let for almost all $n \in N$

$$\varphi_1(n)\, \varphi_2(n) \ldots \varphi_p(n) \leqq n\,.$$

Define the complexity function L by

$$L(n) = \max\,(L_1(n), L_2(n), \ldots, L_p(n), \log n)\,, \quad n \in N\,.$$

Then there exists an i-function $\eta \in \boldsymbol{I}(L)$ such that

1. For almost all $n \in N$

$$C_1\, \varphi_1(n)\, \varphi_2(n) \ldots \varphi_p(n) \leqq \eta(n) \leqq C_2\, \varphi_1(n)\, \varphi_2(n) \ldots \varphi_p(n)\,,$$

where $C_1 > 0$, $C_2 > 0$ are certain constants;

2. η is almost concave.

Proof. It follows from previous theorem by induction.

3. EXAMPLES OF SIMPLE i-FUNCTIONS

In following, any two function $\varphi : N_0 \to R_+$, $\psi : N_0 \to R_+$ we shall call equivalent (R_+ denotes the set of nonnegative real numbers), if there are constants $C_1 > 0$, $C_2 > 0$ such that for almost all $n \in N$

$$C_1\, \varphi(n) \leqq \psi(n) \leqq C_2\, \varphi(n)\,.$$

Lemma 3.1. Let s be a rational number, $0 < s \leqq 1$. Then there exists an almost concave i-function $\varphi \in \boldsymbol{I}(\log)$ equivalent to ψ, where

$$\psi(n) = n^s\,, \quad n \in N\,.$$

Proof. The assertion is trivial for $s = 1$. Suppose that $0 < s < 1$. The identity function e ($e(n) = n$, $n \in N$) is generable by finite automaton. It means that $e \in C(1)$. Let $m \in N$, define h by $h(n) = n^m = (e(n))^m$, $n \in N$. Using Theorem 2.4 we obtain $h \in \boldsymbol{C}(\log)$. Now, let $p, q \in N$ be such that $s = p/q$; thus $p < q$. Define f, g by $f(n) = n^q$, $g(n) = n^p$, $n \in N$, we have f, $g \in \boldsymbol{C}(\log)$. In accordance to Theorem 2.7, let $r = r(f, g)$ and let φ be the i-function which corresponds to r. Then $\varphi \in \boldsymbol{I}(\log)$. Moreover, by the first assertion of Theorem 2.7 we have

$$\varphi(n^q) = n^p \tag{3.1}$$

for all $n \in N$. Since φ is nondecreasing, it is therefore equivalent to ψ, where $\psi(n) = n^s$, $n \in N$. Besides, from (3.1) we can obtain

$$\lim_{n \to +\infty} \varphi(n)/n^s = 1\,.$$

Now, since ψ is concave in the interval $\langle 0, +\infty)$ and φ is equivalent to ψ, so φ must be almost concave which completes the proof.

Before the next lemma we introduce a notation for iterated logarithm. For each $k \in N$ we define a function $\lg_k : R \to R_+$ by recursion scheme:

$$\lg_1 x = \begin{cases} \log_2 x & \text{for} \quad x \geqq 1 \, ; \\ 0 & \text{for} \quad x \leqq 1 \, ; \end{cases}$$

$$\lg_{k+1} x = \lg_1 (\lg_k x) \quad \text{for all} \quad x \in R \, ; \quad k = 1, 2, \ldots \, .$$

Lemma 3.2. Let $k \in N$, let $r > 0$ be a rational number. Then there exists an almost concave i-function $\varphi \in I(\log)$ equivalent to ψ, where

$$\psi(n) = \lg_k^r n, \quad n \in N \, .$$

Proof. At first, it is easy to see that the function f, where $f(n) = 2^n$, $n \in N$, belongs to the class $C(\log)$. E.g. it was shown in [5] that $f_1(n) = 3{\cdot}2^{n-1} - n - 1$ belongs to $C(\log)$. Define $g_1(n) = n + 1$, then we have $f_2 = f_1 + g_1 \in C(\log)$, and again, $f_3 = f_2 + f_2 \in C(\log)$. Since $f_3(n) = 3{\cdot}2^n$, then, using speed-up Lemma 2.2 ($r = 3$) we obtain our result.

Now, define a sequence $f_1, f_2, \ldots$ of functions by

$$f_1(n) = 2^n \, , \quad f_{k+1}(n) = f_1(f_k(n)) \, , \quad n \in N \, ; \quad k \in N \, .$$

Using Theorem 2.5 we have $f_k \in C(\log)$ for each $k \in N$. Let $p, q \in N$, $r = p/q$. Define g by $g(n) = n^q$; we have $g \in C(\log)$, see the proof of Lemma 3.1. Using again Theorem 2.5 we see that $h \in I(\log)$, where $h(n) = f_k(g(n))$, $n \in N$. Let ζ be the i-function which corresponds to h. Then

$$\zeta(n) = [(\lg_k n)^{1/q}] \, , \quad n \in N \, ,$$

where the square brackets denote the entire part of the number. Obviously, ζ is an almost concave function, $\zeta \in I(\log)$. Setting $\varphi_1 = \varphi_2 = \ldots = \varphi_p = \zeta$ in Theorem 2.9, we find out that an almost concave function $\eta \in I(\log)$ exists which is equivalent to ζ^p. Since the functions ζ^p, ψ are obviously equivalent to each other, we conclude that η is equivalent to ψ, completing the proof.

Example 3.1. Let $p \in N$, let r be a rational number, $0 < r < 1$, let $r_1, r_2, \ldots, r_p$ be nonnegative rational numbers. Then there exists a simple i-function φ equivalent to η, where

$$\eta(n) = n^r \lg_1{}^{r_1} n \lg_2{}^{r_2} n \ldots \lg_p{}^{r_p} n \, , \quad n \in N \, .$$

Proof. By Lemma 3.1, there is an almost concave i-function $\varphi_0 \in I(\log)$ equivalent to ψ_0, where $\psi_0(n) = n^r$, $n \in N$.

By Lemma 3.2, for each $j = 1, 2, \ldots, p$ there is an almost concave i-function $\varphi_j \in I(\log)$ equivalent to ψ_j, where $\psi_j(n) = \lg_j{}^{r_j} n$, $n \in N$.

Now, by Theorem 2.9, there is an (almost concave) i-function $\varphi \in I(\log)$ equivalent to ψ, where $\psi(n) = \varphi_0(n)\, \varphi_1(n) \ldots \varphi_p(n)$, $n \in N$. Since φ_j is equivalent to ψ_j for each $j = 1, 2, \ldots, p$, φ is equivalent to η. Now, since $\eta(n) \geqq \log n$ for almost all $n \in N$, we conclude that $\varphi \in I(\eta)$ and furthermore, $\varphi \in I(\varphi)$, for η is equivalent to φ. Thus, the i-function φ is simple.

Example 3.2. Let $p \in N$, let $r_1, r_2. \ldots, r_p$ be arbitrary nonnegative rational numbers $r_1 \geqq 1$. Then there exists a simple i-function φ equivalent to η, where

$$\eta(n) = \lg_1{}^{r_1} n \lg_2{}^{r_2} n \ldots \lg_p{}^{r_p} n\,, \quad n \in N\,.$$

Proof. It is analogical to the proof of the previous example. Again, it may be shown that $\varphi \in I(\log)$.

(Received March 25, 1977.)

REFERENCES

[1] J. Hartmanis, P. M. Lewis II, R. E. Stearns: Classifications of computations by time and memory requirements. IFIP Congress in New York, 1965.

[2] H. Yamada: Counting by a class of growing automata. PhD Thesis, Moore School of Elect. Eng., University of Pennsylvania (1960).

[3] H. Yamada: Real-time computation and recursive functions not real time computable. IRE Trans. on Electronic Computers, *EC-11* (1960).

[4] P. C. Fischer, A. R. Meyer, A. L. Rosenberg: Time-restricted sequences generation. Journal of Computer and System Sciences *4*, (1970) 50—73.

[5] M. Nekvinda: On the complexity of events recognizable in real time. Kybernetika *9*, (1973), 1, 1—10.

[6] M. Nekvinda: On the complexity of countable functions. Proceedings of Symposium and Summer School on Mathematical Foundations of Computer Science, High Tatras, Czechoslovakia, 1973.

RNDr. Miloslav Nekvinda, CSc., strojní fakulta ČVUT (Faculty of Mechanical Engineering — Czech Technical University), Karlovo nám. 13, 131 35 Praha 2. Czechoslovakia.