KYBERNETIKA — VOLUME 9 (1973), NUMBER 1

On the Complexity of Events Recognizable in Real Time

MILOSLAV NEKVINDA

In the article, events recognizable in real time on multitape automata are studied. It is proved that there exists a nontrivial classification of these events with respect to memory requirements.

1. INTRODUCTION

Nowadays, a number of articles dealing with the complexity of events exists. The complexity of a given event can be measured, e.g., by the maximal number of tacts which needs a Turing machine for recognition of words with respect to their length. The events can also be classified by the number of cells needed for the recognization. These questions were studied in [4], [5] for various types of growing automata. P. C. Fischer poses in [2] this problem: being given a complexity class of events with respect to the time measure, classify the class furthermore with respect to the memory measure. We shall show that for the events which are recognizable in real time (in the sense of M. O. Rabin (see [6])) a nontrivial classification with respect to the memory requirements exists.

2. DEFINITIONS AND BASIC PROPERTIES

In the following, N denotes the set of natural numbers, N_0 the set of nonnegative integers. If X is a nonempty set, then the symbol X^∞ stands for the set of all finite sequences (words) of elements of the set (alphabet) X, including the empty word. If $x \in X^\infty$, then symbol $|x|$ denotes the length of the word x; thus, if $x = x_1x_2 \ldots x_n$, where $x_i \in X$ for $i = 1, 2, \ldots, n$, then $|x| = n$. Being given two words $x = x_1x_2 \ldots$ $\ldots x_m \in X^\infty$, $y = y_1y_2 \ldots y_n \in X^\infty$, then the symbol xy denotes concatenation of the words x and y, i.e., the word $x_1x_2 \ldots x_my_1y_2 \ldots y_n$. Under an event over the alphabet X we understand any subset $A \subset X^\infty$.

Definition 2.1. Let $n \in N$. Let be given finite nonempty sets (alphabets) F (the alphabet of internal states), Σ (the input alphabet), S_i (the alphabet of ith tape), $i = 1, 2, \ldots, n$, Π (the output alphabet), $P = \{-1; 0; 1\}$ (the alphabet of moves). Under an automaton I with input and output and with n (in both directions infinite) working tapes we understand a finite set of $(3n + 4)$-tuples of the form

$$(2.1) \qquad (s_i; \alpha_j; \mathrm{S}_{i_1}, \mathrm{S}_{i_2}, \ldots, \mathrm{S}_{i_n}; \mathrm{S}_{j_1}, \mathrm{S}_{j_2}, \ldots, \mathrm{S}_{j_n}; m_1, m_2, \ldots, m_n; s_k; \beta_r),$$

where $s_i \in F$; $\alpha_j \in \Sigma$; $\mathrm{S}_{i_p} \in S_p$, $\mathrm{S}_{j_p} \in S_p$, $p = 1, 2, \ldots, n$; $m_i \in P$, $i = 1, 2, \ldots, n$; $s_k \in F$; $\beta_r \in \Pi$. We assume that for any combination of the first $n + 2$ symbols in (2.1) just one such $(3n + 4)$-tuple exists.

Remark. The rules (2.1) are interpreted in the usual manner. The activity of the automaton is divided into tacts. In a given tact, the automaton being in the internal state s_i, the symbol α_j on input, and the symbols S_{i_p} on the active (just scanned) cells of tapes, it acts as follows: the symbols on tapes are changed to S_{j_p}, the reading heads move (to the right if $m_p = 1$, left if $m_p = -1$, the head does not move if $m_p = 0$), the automaton goes over to the new internal state s_k and the symbol β_r is printed on the output. Further, we assume that in the first tact the automaton I is in an initial state $\bar{s} \in F$, and all tapes are blank.

Definition 2.2. Let Σ and Π be two alphabets and Φ be an operator (mapping) which maps Σ^∞ into Π^∞. We say that the operator Φ is a machine-operator (see [3]) if two following conditions hold:

1. $|\Phi(x)| = |x|$ for any $x \in \Sigma^\infty$;
2. for any two words $x \in \Sigma^\infty$, $u \in \Sigma^\infty$ there is a word $v \in \Pi^\infty$ that

$$\Phi(xu) = \Phi(x)\, v\,.$$

It is evident that any automaton I described in definition 2.1 defines (realizes) some machine-operator Φ_I. But, there are machine-operators which cannot be realized by any automaton of the above type.

Definition 2.3. Let be given a machine-operator Φ which maps Σ^∞ into Π^∞. We say that Φ is realizable in real time if an automaton I from definition 2.1 exists so that $\Phi_I = \Phi$. The class of such operators we denote by symbol $\boldsymbol{R}$.

If we deal with recognition of events over alphabet Σ, we usually choose the output alphabet consisting of two elements, e.g., $\Pi = \{0; 1\}$. The word $x \in \Sigma^\infty$ is then accepted, i.e., $x \in A$, if the word $\Phi(x)$ ends with symbol 1, else $x \notin A$. The event $A \in \Sigma^\infty$ is thus characterized by the machine-operator Φ, which maps Σ^∞ into $\{0; 1\}^\infty$.

Definition 2.4. We say that an event $A \subset \Sigma^\infty$ is recognizable in real time if the operator Φ which characterizes the event A is realizable in real time.

Let be given an automaton I with m tapes. Let $x \in \Sigma^{\infty}$, $|x| = n$. When working on the word, automaton I (during the first n tacts) uses on ith tape $q_i(x)$ cells, $i = 1, 2, \ldots, m$. Denote

$$r_i(n) = \max_{|x|=n} q_i(x) .$$

Define function M

$$M(n) = \max_{1 \leqq i \leqq m} r_i(n)$$

and function M_1

$$M_1(n) = \sum_{i=1}^{m} r_i(n) .$$

It is evident that inequalities

$$M(n) \leqq n \,, \quad M_1(n) \leqq mn \tag{2.2}$$

hold for any $n \in N$.

Definition 2.5. Let be given a nondecreasing function L mapping N into the set of nonnegative numbers (integer values are not assumed). Such a function we shall call a complexity function. We say that automaton I works with space limitation L if

$$M(n) \leqq L(n)$$

holds for almost all $n \in N$ (i.e., for all natural numbers from a certain on).

We say that I works with total space limitation L_1 if for almost all $n \in N$

$$M_1(n) \leqq L_1(n) .$$

Immediately from the definition it follows that an automaton I with m tapes which works with space limitation L works with total space limitation $L_1 = mL$. By the theorem of space compression (see [4]), there exists an automaton I_1, which works with space limitation $1/mL$, and thus with total space limitation L. Thus when studying the complexity of operators with respect to memory requirements, we can deal only with the function M.

Definition 2.6. Let be given a complexity function L. We say that an operator $\Phi \in \boldsymbol{R}$ belongs to the complexity class $\boldsymbol{R}(L)$, $\Phi \in \boldsymbol{R}(L)$, if an automaton I exists such that

1. $\Phi_I = \Phi$;
2. automaton I works with space limitation L.

We say that an event A belongs to the complexity class $\boldsymbol{R}(L)$ if the operator Φ which characterizes the event A, belongs to the class $\boldsymbol{R}(L)$.

It is evident that finite automata realize operators with space limitation $L(n) = \text{const}$; e.g., we can define $L(n) = 1$ for all $n \in N$. It means that the class of operators which are realizable by finite automata is equal to the class $\boldsymbol{R}(1)$. From (2.2) it follows that for any operator $\boldsymbol{R}$, which is realizable in real time, $\Phi \in \boldsymbol{R}(E)$ holds, with $E(k) = k$ for $k \in N$. Thus, $\boldsymbol{R} = \boldsymbol{R}(E)$. The theorem of space compression implies that if L is a complexity function and $c > 0$, then $\boldsymbol{R}(L) = \boldsymbol{R}(cL)$. Further, if L_1, L_2 are two complexity functions such that $L_1(n) \leqq L_2(n)$ for almost all n, then $\boldsymbol{R}(L_1) \subset \boldsymbol{R}(L_2)$. From these two remarks it follows: if L_1, L_2 are two complexity functions and a constant $c > 0$ exists that $L_1(n) \leqq cL_2(n)$ for almost all n, then $\boldsymbol{R}(L_1) \subset \boldsymbol{R}(L_2)$. If $\lim_{n\to\infty} L_1(n)/L_2(n)$ exists and $0 < \lim_{n\to\infty} L_1(n)/L_2(n) < +\infty$, then $\boldsymbol{R}(L_1) = \boldsymbol{R}(L_2)$.

In the following, we shall show that there exist infinitely many distinct complexity classes between the simpliest class $\boldsymbol{R}(1)$ and the class $\boldsymbol{R} = \boldsymbol{R}(E)$ of operators realizable in real time.

3. BASIC THEOREMS ABOUT CLASSIFICATION

We begin with a theorem which gives information about the lower bound of the hierarchy of complexity.

Theorem 3.1. *Let for a complexity function L hold*

$$\liminf_{n\to\infty} \frac{L(n)}{\log n} = 0\,,$$

where $\log n = \log_2 n$. *Then*

$$\boldsymbol{R}(L) = \boldsymbol{R}(1)\,,$$

i.e., each operator realizable with space limitation L is realizable by a finite automaton.

This theorem was formulated for some classes of growing automata in [4]. The proof given in [4] can be basically reproduced in our case, and we leave it out.

We introduce some other notions so that we could formulate the second basic theorem about classification.

Definition 3.1. Y-automaton (automaton of Yamada's type, or autonomous automaton) is an automaton from definition 2.1, the input alphabet of which consists of one element.

Remark. These automata and generated functions were studied in detail in [7] and [8].

It is clear that Y-automaton can be interpreted as an automaton without input. Such an automaton generates an infinite sequence of elements of the output alphabet. Naturally, each of the above given definitions of complexity of operators remain valid for autonomous operators, i.e., for operators realizable by Y-automata. Let the output alphabet consist of two elements, suppose that $\Pi = \{0; 1\}$. In this case, Y-automaton generates a sequence

$$\alpha = \alpha_1\alpha_2\alpha_3 \ldots \tag{3.1}$$

of 0's and 1's. If the sequence (3.1) contains an infinite number of 1's, we call it regular. To any regular sequence we assign a function f (see [7] and [8]) mapping N into N:

$$f(n) = \min \{p;\ p \in N, \sum_{i=1}^{p} \alpha_i = n\} . \tag{3.2}$$

It is clear that f is increasing. We denote by F the mapping which assigns to any regular sequence (3.1) the function f according to the rule (3.2), thus $f = F(\alpha)$. On the other hand, to any increasing function f mapping N into N we can assign just one sequence α of the form (3.1) so that $F(\alpha) = f$. Naturally, not any sequence α can be generated by an Y-automaton.

Definition 3.2. An increasing function f mapping N into N is called countable if an Y-automaton I exists which generates the sequence α so that $F(\alpha) = f$. In this case, we say that automaton I generates f.

We can assign to the sequence (3.1) also a function which is, roughly speaking, inverse to f. Denoting this assignment F_1, we can define the function $\varphi = F_1(\alpha)$ by the rule

$$\varphi(n) = \sum_{i=1}^{n} \alpha_i . \tag{3.3}$$

The function defined in (3.3) has following properties:

1. φ is a nondecreasing mapping from N to N_0, $\varphi(1) = 0$, or $\varphi(1) = 1$;
2. $\varphi(n + 1) - \varphi(n) \leqq 1$ for any $n \in N$;
3. for any regular sequence α, $\lim_{n\to\infty} \varphi(n) = +\infty$ holds.

Definition 3.3. A function φ mapping N into N_0 is an i-function if an Y-automaton I exists which generates regular sequence α such that $F_1(\alpha) = \varphi$.

It is evident that there exists a one-to-one mapping between i-functions and countable functions given by $f = F(F_1^{-1}(\varphi))$.

Definition 3.4. An i-function φ is simple if an Y-automaton I exists which generates the function (i.e., the corresponding sequence α) with space limitation $L = \varphi$. In this case, we shall also use the adjective simple both for the Y-automaton and the countable function f.

It appears that simple i-functions form an important subset of complexity functions.

Theorem 3.2. *Let L_1 be a simple i-function. Then there exists an event A such that*

1. $A \in \boldsymbol{R}(L_1)$;

2. *if L is another complexity function such that* $\liminf_{n \to \infty} L(n)/L_1(n) = 0$, *then* $A \notin \boldsymbol{R}(L)$.

Proof. I. First, we construct the event A. We denote $\Sigma = \{0; 1; *\}$, $\Sigma_1 = \{0; 1\}$. Let I_1 be an Y-automaton generating the i-function L_1 with space limitation L_1. Such an automaton exists because L_1 is simple according to the assumption. Let f be the countable function generated by the automaton I_1; thus, $f = F(F_1^{-1}(L_1))$. Define the event A as the set of all words over alphabet Σ of the form

(3.4) $$\alpha_1\alpha_2 \ldots \alpha_n * \alpha'_m\alpha'_{m-1} \ldots \alpha'_1 ,$$

where

$$\alpha_i \in \Sigma_1 , \quad \alpha'_j \in \Sigma_1 , \quad i = 1, 2, \ldots, n; j = 1, 2, \ldots, m ,$$

and the following conditions hold

(3.5) $$m = L_1(n) ; \quad \alpha_{f(i)} = \alpha'_i , \quad i = 1, 2, \ldots, m .$$

(If $L_1(n) = 0$, then the word from (3.4) reduces to $\alpha_1\alpha_2 \ldots \alpha_n*$.)

Less formally, from the sequence $\alpha_1\alpha_2 \ldots \alpha_n$ we remember the symbols which are on the input when automaton I_1 prints symbol 1. Having written the symbol $*$, we write the selected sequence in inverse ordering.

II. Now, we shall show that the constructed event can be recognized in real time with space limitation L_1. To this end we add another tape and input to the automaton I_1. The tape will serve for writing down symbols of Σ, which will be also on the added input. Actually, we have constructed a new automaton I which contains the automaton I_1. Naturally, we add new internal states to those of I_1, if necessary. Leaving out such details, we now describe the activity of I. As a matter of fact, the activity of I follows from the definition of the event A. It can be divided into two stages:

Stage 1. The sequence $\alpha_1\alpha_2 \ldots \alpha_i \in \Sigma_1$ is on the input. Now, the automaton copies on the added tape from the left to the right the symbols which are on input when I_1 prints symbol 1. The main thing is that the space needed on that tape equals $L_1(n)$, the needed space (i.e., the needed cells) on other tapes being less or equal to $L_1(n)$ as well (I_1 is a simple automaton). Thus, the automaton I works during this stage with space limitation L_1.

Stage 2. Let symbol $*$ appear for the first time on the input. Now, with respect to (3.4), (3.5) the head on the added tape, where "the important" symbols of the input word are written, goes from the right to the left comparing its symbols with those read. During this stage the space limitation L_1 is also respected (we can even stop the activity of the subautomaton I_1). It is evident that automaton I recognizes the event A in real time (organization of the output symbols is evident).

III. Now, we shall show that the event A is too complicated for any automaton which works with space limitation less than L_1. Let L be a complexity function fulfilling the assumptions of the theorem. Suppose that $A \in \boldsymbol{R}(L)$, i.e., that there exists an automaton I which recognizes the event A with space limitation L. The overal state of the automaton in any tact can be characterized by: 1. internal state; 2. words that are written on each tape (the words are finite, their length equals to the number of cells which were active, i.e., were scanned); 3. position of reading heads on tapes. This information together we call a configuration of the automaton I.

Let F be the alphabet of internal states, and S_i be the alphabets of tapes. Denote $s = \operatorname{card} F$ (the number of elements of the set F), $r_i = \operatorname{card} S_i$, $i = 1, 2, \ldots, m$, $r = \max r_i$, $i = 1, 2, \ldots, m$ (automaton I is supposed to have m tapes). Let $G(n)$ be the number of all possible configurations of I in the ith tact. As I works with space limitation L, obviously

$$G(n) \leqq s r^{mL(n)} L^m(n) . \tag{3.6}$$

Automaton I must be able to distinguish any two sequences which differ on "important" places. These important places in the sequence $\alpha_1 \alpha_2 \ldots \alpha_n$ are given by indices $f(1), f(2), \ldots, f(m)$, the number of those, of course, equals $m = L_1(n)$ (see (3.5)). This implies existence of $2^m = 2^{L_1(n)}$ sequences which differ on important places. As any two such sequences necessarily transfer the automaton I into different configurations after n tacts, we have, with respect to (3.6)

$$2^{L_1(n)} \leqq s r^{mL(n)} L^m(n) ,$$

for all $n \in N$. Logarithming gives $L_1(n) \leqq \log s + m(1 + \log r) L(n)$ from which we get inequality $\liminf_{n \to \infty} L(n)/L_1(n) = 1/(m(1 + \log r)) > 0$ what is a contradiction with the assumption of the theorem. Thus, $A \in \boldsymbol{R}(L)$ cannot hold, and theorem is proved.

A simple consequence of the above theorem is the following one which we state without proof.

Theorem 3.3. *Let L_1 be a simple i-function and L be such complexity function that*

$$\lim_{n \to \infty} L(n)/L_1(n) = 0 .$$

Then $\boldsymbol{R}(L) \subsetneqq \boldsymbol{R}(L_1)$, i.e., the complexity class $\boldsymbol{R}(L)$ is a proper subset of the class $\boldsymbol{R}(L_1)$.

Theorem 3.4. *Let φ be an i-function (not supposed to be simple), let I be an automaton generating φ. Then the function M defined in (2.2) is either bounded or there exists such $c > 0$ that for almost all n*

$$M(n) \geqq c \log n .$$

Proof. Suppose, the assertion of the theorem does not hold. Thus, let M be unbounded and

$$\liminf_{n\to\infty} M(n)/\log n = 0 . \tag{3.7}$$

Now, we modify the activity of I in this way: automaton I (assume that it has m tapes) will print on output symbol 1 just at those tacts when some reading head on tapes reaches a cell which has not been scanned. Evidently, this situation can be easily recognized by an appropriate modification of alphabets on tapes. The modified automaton we denote I_1. Clearly, the automaton I_1 works with the same space limitation as automaton I. Thus, we see that if $\varphi_1(n)$ is the number of 1's which automaton I_1 has printed on output up to the nth tact, then

$$M(n) \leqq \varphi_1(n) \leqq m\, M(n) , \quad n \in N .$$

From this follows that i-function φ_1 generated by automaton I_1 is a simple one. We see from (3.7) that for φ_1 we have

$$\liminf_{n\to\infty} \varphi_1(n)/\log n = 0 .$$

Now, by theorem 3.1, the class $\boldsymbol{R}(\varphi_1)$ is the class $\boldsymbol{R}(1)$. On the other hand, theorem 3.2 states that $\boldsymbol{R}(\varphi_1)$ is richer than $\boldsymbol{R}(1)$. This contradiction proves our theorem.

Theorem 3.5. *Let φ be a simple i-function. Then there exists a constant $c > 0$ that for almost all $n \in N$*

$$\varphi(n) \geqq c \log n . \tag{3.8}$$

Proof. Let I be an arbitrary simple Y-automaton generating the i-function φ. If $I \in \boldsymbol{R}(1)$, then it is equivalent to a finite automaton, and because the activity of a finite automaton is periodic, we have for the function φ that $\varphi(n) \geqq Cn$ for almost all n, where $C > 0$ is a certain constant. All the more, condition (3.8) holds. If $I \notin \boldsymbol{R}(1)$, then from theorem 3.4 it follows that for the function M defined in (2.2) $M(n) \geqq c \log n$ for almost all $n \in N$, where $c > 0$ is a constant. As I is simple, $\varphi(n) \geqq M(n)$ for almost all $n \in N$. This implies that for almost all $n \in N$ the inequality (3.8) is valid, completing the proof.

The above theorems give us insight into possible existence of nontrivial hierarchy of complexity classes. Nonetheless, they do not assert that such a hierarchy does exist. That depends on the existence of simple i-functions. Even theorem 3.1 does not guarantee that the class $\boldsymbol{R}(\log n)$ is richer than the class $\boldsymbol{R}(1)$. To prove this, it is necessary to show that the function $\log n$ is basically a simple i-function. In [8], the countability of the function 2^n is proved, implying that $[\log n]$ is an i-function. The mentioned proof does not guarantee the simplicity of the function.

Theorem 3.6. *There exists such simple i-function φ that*

$$0 < \liminf_{n\to\infty} \varphi(n)/\log n \leqq 1\,. \tag{3.9}$$

Proof. We construct an Y-automaton I with one tape on which the binary codes of consecutive integers $1, 2, 3, \dots$ will be written. Automaton I will print symbol 1 just at those tacts when the numbers of the form 2^k, $k \in N_0$ are witten for the first time on the tape. At the other tacts, it gives symbol 0. In order to distinguish uniqually these situations, we shall mark the extreme left symbol 1 with 1_{L}. The extreme right end we denote 0_{R} or 1_{R} intead of 0 or 1. Symbol B denotes the blank symbol. Adding unity, we get on tape a sequence of configurations (words), the position of the reading head being marked by posing the letter q before the read symbol. The first configuration will give the situation on the tape after the first tact and so on. For the sake of better orientation we describe several initial configurations (leaving out the internal states of the automaton I):

$$q1_{\mathrm{R}},\ qB0_{\mathrm{R}},\ q1_{\mathrm{L}}0_{\mathrm{R}},\ 1_{\mathrm{L}}q0_{\mathrm{R}},\ 1_{\mathrm{L}}q1_{\mathrm{R}},\ q1_{\mathrm{L}}0_{\mathrm{R}},\ qB00_{\mathrm{R}},\ q1_{\mathrm{L}}00_{\mathrm{R}},\ 1_{\mathrm{L}}q00_{\mathrm{R}},\ 1_{\mathrm{L}}0q0_{\mathrm{R}},\ 1_{\mathrm{L}}0q1_{\mathrm{R}},$$
$$1_{\mathrm{L}}q00_{\mathrm{R}},\ 1_{\mathrm{L}}q10_{\mathrm{R}},\ 1_{\mathrm{L}}1q0_{\mathrm{R}},\ 1_{\mathrm{L}}1q1_{\mathrm{R}}, 1_{\mathrm{L}}q10_{\mathrm{R}},\ q1_{\mathrm{L}}00_{\mathrm{R}},\ qB000_{\mathrm{R}},\ q1_{\mathrm{L}}000_{\mathrm{R}}, \dots$$

The sequence on the output has the form

$$10100001000000000010\dots \tag{3.10}$$

It follows that if there is a word of length $k + 1$ on the tape, then automaton I has printed at least k units. This implies that the i-function φ belonging to the sequence (3.10) is generated by I with space limitation L, where $L(n) = \varphi(n) + 1$, $n \in N$. If we reduce one cell, we can generate the function φ with space limitation $L = \varphi$. Thus, φ is simple. By theorem 3.5, there is a constant $c > 0$ such that for almost all n, $\varphi(n) \geqq c \log n$. On the other hand, for any $k \in N_0$, the number of tacts necessary for the production of the code of the number 2^k is greater or equal 2^k (supposing that we managed to add unit always in one tact, the number of tacts needed would be 2^k), which gives for φ the inequality $\varphi(n) \leqq \log n + 1$, (again, adding units in one tact gives $\varphi(n) = [\log n] + 1$, because the corresponding sequence (3.10) would have the form 1101000010…). The combination of inequalities gives

$$c \log n \leqq \varphi(n) \leqq \log n + 1$$

for almost all n, which implies immediately (3.9), completing the proof.

Remark. It can be shown that for the i-function φ belonging to (3.10)

$$\lim_{n\to\infty} \varphi(n)/\log n = 1 \tag{3.11}$$

holds. A detailed analysis does give that the countable function f belonging to

(3.10) equals

$$f(n) = 3 \cdot 2^{n-1} - n - 1\,, \quad n \in N\,.$$

This implies (3.11).

Theorems 3.1, 3.2 and 3.6 imply that the class $\boldsymbol{R}(L)$, with $L(n) = \log n$, is in the classification hierarchy just above the class $\boldsymbol{R}(1)$ realizable by finite automata. As we have mentioned above, the existence of complexity classes is a consequence of the existence of simple functions. It appears that not any countable function is simple (and therefore not any i-function is a simple one). In [8], f.e., it is proved that 2^{n^2} is countable. Using theorem 3.5 (for the corresponding i-function), we get that the function cannot be simple.

The existence of a sufficiently rich class of simple functions will be discussed in a future paper. There, it will be shown that for any rational number $r \in (0; 1)$ there is a simple i-function φ that

$$\lim_{n\to\infty} n^r/\varphi(n) = 1$$

holds. This implies validity of the following theorem.

Theorem 3.7. *There exists an infinite hierarchy of complexity classes of operators realizable in real time. If r_1, r_2 are any two rational numbers that $0 < r_1 < r_2 \leqq 1$, then for the corresponding classes $\boldsymbol{R}(n^{r_1})$, $\boldsymbol{R}(n^{r_2})$ proper inclusion*

$$\boldsymbol{R}(n^{r_1}) \subsetneqq \boldsymbol{R}(n^{r_2})$$

holds.

(Received January 27, 1972.)

REFERENCES

[1] Bečvář, J.: Real-Time and Complexity Problems in Automata Theory. Kybernetika *1* (1965), 6, 475–497.

[2] Fischer, P. C.: Multi-tape and infinite-state automata. International Colloquium on Algebraic Linguistics and Automata Theory, 1964, Jerusalem, Israel.

[3] Глушков, В. М.: Синтез цифровых автоматов. Физматгиз, Москва 1962.

[4] Hartmanis, J.; Lewis, P. M.; Stearns, R. E.: Classifications of Computations by time and memory requirements. IFIP Congress, New York 1965.

[5] Hartmanis, J.; Stearns, R. E.: On the computational complexity of algorithms. Trans. Amer. Math. Soc. *117* (1965), 285–306.

[6] Rabin, M. O.: Real time computation. Israel J. of Math. *1* (1963), 203–211.

[7] Yamada, H.: Real-time computation and recursive functions not real-time computable. IRE Trans. on Electronic Computers *EC-11* (1965), 753–760.

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

RNDr. Miloslav Nekvinda, CSc., Vysoká škola strojní a textilní v Liberci (The College of Mechanical and Textile Engineering in Liberec), Hálkova 6, 461 17 Liberec 1. Czechoslovakia.