Algebra and Discrete Mathematics RESEARCH ARTICLE
Number 4. **(2005).** pp. 36 – 47

Automaton extensions of mappings on the set of words defined by finite Mealy automata

Mirosław Osys

Communicated by V. I. Sushchansky

Abstract. The properties of an automaton extensions of mappings on the set of words over a finite alphabet is discussed. We obtain the criterion whether the automaton extension of given mapping if defined by a finite automaton.

Introduction

In the set of all transformations of the set of finite words over given alphabet we distinguish a subset of the automaton mappings, i.e. transformations induced by (finite or infinite) Mealy automata . Although both sets are uncountable, not every function $f : X^* \to X^*$ is defined by certain automaton.

In sixties of the XX century has been indicated (e.g. in [3]) that after addition a new symbol to the alphabet, arbitrary transformation can be extended to an automaton mapping, that uniquely determines the initial transformation. Moreover, an effective method for such constructions has been established. Algol 60 algorithm for finding the minimum necessary number of the new symbols is developed in [1].

Since mentioned extension is not unique, we define two different possibilities of the construction, the plain extension and the cyclic extension, and develop some basic properties of them. We discuss the problem whether exists a finite automaton inducing extension for given transformation. The main result of this paper is

2000 Mathematics Subject Classification: *68Q70, 68Q45.*
Key words and phrases: *automaton mapping, Mealy automaton.*

Theorem 1. *Let $\widehat{f} : X_\alpha^* \to X_\alpha^*$ be the plain or cyclic extension of the transformation $f : X^* \to X^*$. The extension $\widehat{f}$ is a finite–state automaton mapping if and only if the following conditions are satisfied:*

1. *$f(X^*)$ is a finite set,*
2. *the inverse image $f^{-1}(u)$ is a regular language for each $u \in f(X^*)$.*

The contents are organized as follows. In Section 1 we list the basic notations and recall a notions of a Mealy automaton and a Rabin–Scott automaton. In Section 2 we define a notion of an automaton extension of the transformation. In Section 3 we prove Theorem 1.

1. Preliminaries

1.1. Let X be an alphabet, and X^* be the free monoid over X with an empty word ε as a neutral element. We shall write uv for the product of $u, v \in X^*$ and u^k for $\overbrace{u \ldots u}^{k}$. The length of the word u is denoted by $|u|$. The word u is a *prefix* of the word v (denoted by $u \leq v$) if $v = uw$ for the certain word w. The word v is a *segment* of u if there exist words $u_1, u_2 \in X^*$ such that $u = u_1 v u_2$.

1.2. An *initial Mealy-type automaton* over the alphabet X is a tuple

$$\mathcal{A} = (Q, q_0, X, \delta, \lambda)$$

which consists of the following data:

- ▷ a set Q of the *internal states*, $Q \neq \emptyset$
- ▷ a distinguished state $q_0 \in Q$ called *initial*
- ▷ the alphabet of the automaton, $X \neq \emptyset$
- ▷ a *next-state function* $\delta : Q \times X \to Q$
- ▷ an *output function* $\lambda : Q \times X \to X$

A tuple $\mathcal{A} = (Q, q_0, X, \delta, \lambda)$ is *partial Mealy automaton* if either δ or λ is a partial function. The automaton $\mathcal{A}$ is *finite* if the sets Q and X are finite.

We often use a notation

$$q_i \xmapsto{x/y} q_j$$

instead of

$$\delta(q_i, x) = q_j\,, \qquad \lambda(q_i, x) = y\,.$$

1.3. The next–state function and the output function of the automaton $\mathcal{A} = (Q, q_0, X, \delta, \lambda)$ can be extended on the set $Q \times X^*$ by the following recurrent equalities:

$$\delta(q, \varepsilon) = q\,, \qquad \delta(q, ux) = \delta(\delta(q, u), x)\,,$$

$$\lambda(q, \varepsilon) = \varepsilon\,, \qquad \lambda(q, ux) = \lambda(\delta(q, u), x)\,,$$

where $x \in X$ and $u \in X^*$. An initial automaton $\mathcal{A}$ defines mapping $f_{\mathcal{A}} : X^* \to X^*$ as follows:

$$f_{\mathcal{A}}(\varepsilon) = \varepsilon\,, \qquad f_{\mathcal{A}}(x_1 \dots x_k) = \lambda(q_0, x_1)\lambda(q_0, x_1x_2)\dots\lambda(q_0, x_1 \dots x_k)\,.$$

In case of the partial automaton, $f_{\mathcal{A}}$ is a partial function.

Definition 1. [6] A function $f : X^* \to X^*$ is called an *(finite–state) automaton mapping* if there exists an (finite) automaton $\mathcal{A}$, such that $f = f_{\mathcal{A}}$.

Proposition 1. *A function $f : X^* \to X^*$ is an automaton mapping if and only if it has the following properties:*

1. *it preserves the lengths of the words, that is $|f(u)| = |u|$ for every $u \in X^*$*
2. *(common prefix property) if $u, v \in X^*$ then each prefix w of these words is translated in the same manner, thus it follows that $f(u)$ and $f(v)$ have common prefix of the length greater or equal $|w|$.*

□

Definition 2. A function $f : X^* \to X^*$ is called an *partial automaton mapping* if there exists an partial automaton $\mathcal{A}$, such that $f = f_{\mathcal{A}}$.

Proposition 2. *A function $g : X^* \to X^*$ is a partial automaton mapping if and only if it has the following properties:*

1. *domain of the function is prefix closed, that is, $u \in Dom\, g$ and $v \leq u$ implies that $v \in Dom\, g$,*
2. *there exists an automaton mapping $f : X^* \to X^*$, such that*

$$g = f|_{Dom\, g}\,.$$

□

For the details of the proofs refer to [6], [2], [5], [4].

1.4. A *Rabin–Scott automaton* is a tuple

$$\mathcal{A} = (Q, q_0, T, X, \delta)\,,$$

which is similar to the Mealy automaton, except deleted function λ and added set $T \subset Q$ which collects the *terminal nodes* (or *accept states*). A set of the words

$$L(\mathcal{A}) = \{u \in X^* : \delta(q_0, u) \in T\}$$

will be referred to as the *language recognizable by the automaton* $\mathcal{A}$.

The language $L \subset X^*$ is said to be *regular* if there exists a finite automaton recognizing L. For more information refer to [7], [2].

2. Automaton extension of mapping

2.1. Let $f : X^* \to X^*$ be a function that satisfies $f(\varepsilon) = \varepsilon$. Let $X_\alpha = X \cup \{\alpha\}$, $\alpha \notin X$ be the extended alphabet. With a symbol t we will denote a homomorphism $X_\alpha^* \to X^*$ given by:

$$t(\alpha) = \varepsilon, \qquad t(x) = x, \qquad x \in X\,.$$

Definition 3. An automaton mapping $\widehat{f} : X_\alpha^* \to X_\alpha^*$ is called an *automaton extension mapping* (or simply *extension*) of $f : X^* \to X^*$ if there exists an embedding $\mu_f : X^* \to X_\alpha^*$ such that the following diagram is commutative:

$$\begin{array}{ccc} u \in X^* & \overset{f}{\longmapsto} & f(u) \in X^* \\ \mu_f \downarrow & & \uparrow t \\ u' \in X_\alpha^* & \overset{\widehat{f}}{\longmapsto} & \widehat{f}(u') \in X_\alpha^*\,. \end{array}$$

2.2. The extension $\widehat{f}$ of an arbitrary function $f : X^* \to X^*$ will be defined in two steps:

1. a partial extension is constructed, that is function $X_\alpha^* \to X_\alpha^*$ defined on the certain fixed subset $M \subset X_\alpha^*$,
2. the domain of the obtained function is extended on the monoid X_α^*.

For the construction related to the first step we will apply a method described in [3], p.19.

Definition 4. For every $u \in X^*$ we define

$$\mu_f(u) = u\alpha^{|f(u)|}$$

and introduce a set

$$M = \{v' \in X_\alpha^* : v' \leq \mu_f(u),\ u \in X^*\}.$$

The mapping $\widehat{f} : M \to X_\alpha^*$ is defined as follows:

a) if u' has the form $u\alpha^{|f(u)|}$, $u \in X^*$ then

$$\widehat{f}(u') = \widehat{f}(u\alpha^{|f(u)|}) = \alpha^{|u|} f(u),$$

b) if $v' \in M$ and $v' \leq u' = u\alpha^{|f(u)|}$ then

$$\widehat{f}(v') = w', \quad w' \leq \widehat{f}(u'), \quad |w'| = |v'|.$$

Above definition of the function $\widehat{f}$ is correct since w' does not depend on choosing the word u'. Also, for $u' = \mu_f(u)$, the properties $t(u') = u$ and $t(\widehat{f}(u')) = f(u)$ hold. Thus, the diagram from the definition 3 is commutative.

Proposition 3. *The extension $\widehat{f} : X_\alpha^* \to X_\alpha^*$ is a partial automaton mapping over the alphabet X_α.*

Proof. For every word $u' \in X_\alpha^*$ of the form $u\alpha^{|f(u)|}$ we have

$$|u'| = |u\alpha^{|f(u)|}| = |\alpha^{|u|} f(u)| = |\widehat{f}(u')|$$

and for every prefix $v' \leq u'$

$$\widehat{f}(v') \leq \widehat{f}(u'), \qquad |\widehat{f}(v')| = |v'|.$$

Therefore $\widehat{f}$ preserves the lengths of the words and has the common prefix property. It is also clear that the set M is prefix closed. □

2.3. Example. Consider a function $f : X^* \to X^*$, $X = \{0, 1\}$ defined by

$$f(u) = \begin{cases} 0 & |u| \text{ even}, |u| > 0, \\ 11 & |u| \text{ odd}, \\ \varepsilon & u = \varepsilon. \end{cases}$$

We agree that f is not an automaton mapping since it does not preserve either lengths nor has the common prefix property. Let us show how the extension mapping works. Instead $01 \xmapsto{f} 0$ and $101 \xmapsto{f} 11$ we have

$$01\alpha \xmapsto{\widehat{f}} \alpha\alpha 0 \quad \text{and} \quad 101\alpha\alpha \xmapsto{\widehat{f}} \alpha\alpha\alpha 11\,.$$

It can be seen that on the input side the letter α is utilized to terminate a sequence of letters from the set X, whereas on the output it plays role of an "empty" symbol while the automaton waits for completing the input word.

2.4. We introduce two different methods for extending $\widehat{f}$ on the set X_α^*, which will be referred to as 'plain extension' and 'cyclic extension'.

Definition 5. The *plain* extension of the transformation $f : X^* \to X^*$ is a mapping $\widehat{f}_1 : X_\alpha^* \to X_\alpha^*$ defined by:

- $\widehat{f}_1|_M = \widehat{f}$, where $\widehat{f}$ is the automaton extension of f and M is the set established in definition 4,
- $\widehat{f}_1(\alpha^k v') = \alpha^k \widehat{f}_1(v')$,
- $\widehat{f}_1(u\alpha^{|f(u)|}v') = \alpha^{|u|} f(u)\alpha^{|v'|}$,
- $\widehat{f}_1(u\alpha^m v') = \alpha^{|u|} f(u)\alpha^n, \qquad m + |v'| = |f(u)| + n$
  for $\quad m + |v'| \geq |f(u)|$,
- $\widehat{f}_1(u\alpha^m v') = \alpha^{|u|} v, \qquad v \leq f(u), \quad m + |v'| = |v|$
  for $\quad m + |v'| < |f(u)|$

where $u, v \in X^*$ and $v' \in X_\alpha^*$.

The automaton mapping defined this means ignores appended word v' by treating it as a sequence of "empty" symbols.

For the next definition recall that arbitrary word $v' \in X_\alpha^*$ can be uniquely written as

$$v' = \alpha^{k_0} u_1 \alpha^{k_1} u_2 \alpha^{k_2} \ldots u_n \alpha^{k_n}, \qquad u_i \in X^*$$

where $k_0, k_n \geq 0$ and $k_1, \ldots, k_{n-1} \geq 1$.

Definition 6. The *cyclic* extension of the transformation $f : X^* \to X^*$ is a mapping $\widehat{f}_2 : X_\alpha^* \to X_\alpha^*$ defined by:

- $\widehat{f}_2|_M = \widehat{f}$, where $\widehat{f}$ is the automaton extension of f and M is the set established in definition 4,

- $\widehat{f}_2(\alpha^k) = \alpha^k$,
- $\widehat{f}_2(u\alpha^{|f(u)|}\alpha^k) = \alpha^{|u|}f(u)\alpha^k$,
- $\widehat{f}_2(u\alpha^m) = \alpha^{|u|}v, \quad v \le f(u), \quad |v| = m \quad \text{for} \quad m < |f(u)|$,
- $\widehat{f}_2(v') = \alpha^{k_0}\widehat{f}_2(u_1\alpha^{k_1})\widehat{f}_2(u_2\alpha^{k_2})\dots\widehat{f}_2(u_n\alpha^{k_n})$

where $v' = \alpha^{k_0}u_1\alpha^{k_1}u_2\alpha^{k_2}\dots u_n\alpha^{k_n}$.

The mapping obtained that way translates independently every segment of the form $u_i\alpha^{k_i}$.

2.5. Example. Let $f : \{0,1\} \to \{0,1\}$ be function defined by

$$f(u) = \begin{cases} 11 & \text{if } u = 0, \\ 0 & \text{if } u = 11, \\ \varepsilon & \text{if } u \notin \{0, 11\}. \end{cases}$$

Then the plain extension $\widehat{f}_1$ and the cyclic extension $\widehat{f}_2$ translate words in the following manner:

$u \in X^*_\alpha$	$0\alpha\alpha$	0α	11α	11	$0\alpha0\alpha$	$1\alpha11\alpha$
$\widehat{f}_1(u)$	$\alpha11$	$\alpha1$	$\alpha\alpha0$	$\alpha\alpha$	$\alpha1\alpha\alpha$	$\alpha\alpha\alpha\alpha\alpha$
$\widehat{f}_2(u)$	$\alpha11$	$\alpha1$	$\alpha\alpha0$	$\alpha\alpha$	$\alpha1\alpha1$	$\alpha\alpha\alpha\alpha0$

Proposition 4.

1. *For every function $f : X^* \to X^*$, $f(\varepsilon) = \varepsilon$ the following conditions hold:*

   a) plain and cyclic extensions are well defined automaton mappings over the alphabet X^*_α,

   b) $\widehat{f}_1 \ne \widehat{f}_2$ unless f is trivial (i.e. $f(u) = \varepsilon$ for all $u \in X^*$).

2. *For every $f, g : X^* \to X^*$, such that $f(\varepsilon) = g(\varepsilon) = \varepsilon$ and $f \ne g$, we have:*

$$\widehat{f}_1 \ne \widehat{g}_1, \qquad \widehat{f}_2 \ne \widehat{g}_2\ .$$

Proof. (1a) Clearly, the definitions allow us to calculate $\widehat{f}_1(u')$ and $\widehat{f}_2(u')$ for all $u' \in X^*_\alpha$ thus extensions are well defined. Furthermore, directly from the definitions, it can be seen that extensions $\widehat{f}_1$ and $\widehat{f}_2$ are automaton mappings since for every $u' \in X^*_\alpha$

$$|\widehat{f}_1(u')| = |\widehat{f}_2(u')| = |u'|$$

and the common prefix property holds as well, directly from definitions. (1b) If f is trivial, then $\widehat{f}_1(u') = \widehat{f}_2(u') = \alpha^{|u'|}$ for all $u' \in X_\alpha^*$. Otherwise, there exists $u \in X^*$ such that $f(u) = v \neq \varepsilon$. The extensions are distinct, since

$$\widehat{f}_1(u\alpha^{|v|}u\alpha^{|v|}) = \alpha^{|u|}v\alpha^{|u|}\alpha^{|v|} \qquad \text{and} \qquad \widehat{f}_2(u\alpha^{|v|}u\alpha^{|v|}) = \alpha^{|u|}v\alpha^{|u|}v\,.$$

(2) It is sufficient to prove that f is uniquely determined by $\widehat{f}$. Let $\widehat{f} : X_\alpha \to X_\alpha$ be the plain or cyclic extension. We will show that for every $u \in X^*$ a word $f(u)$ can be calculated. Indeed, take a sequence $v_i = \widehat{f}(u\alpha^i)$, $i = 1, 2, \ldots$ Then exists the number j, such that the last letters of the words v_i, $i \leq j$ are distinct from α and all words v_i, $i > j$ end with the letter α. It is obvious that $f(u) = t(v_j)$, therefore the mapping f is determined by $\widehat{f}$ and the operations $f \mapsto \widehat{f}_1$ and $f \mapsto \widehat{f}_2$ are one-to-one. □

3. Finite state extensions

3.1. The purpose of this part is to find conditions for the mapping f under which an automaton corresponding to the extension mapping $\widehat{f}$ is finite. We assume the set X is finite.

Lemma 1. *Let $f : X^* \to X^*$ be an arbitrary mapping and $\widehat{f} : X_\alpha^* \to X_\alpha^*$ its plain or cyclic extension. If $\widehat{f}$ is finite–state automaton mapping then the set $f(X^*)$ is finite.*

Proof. Let $\mathcal{A} = (Q, q_0, X_\alpha, \delta, \lambda)$ be a finite automaton defining the mapping $\widehat{f}$. Consider words of the form

$$u' = u\alpha^{|f(u)|}, \qquad u \in X^*.$$

If the set $f(X^*)$ is infinite then, since Q is finite, there exist the words u_1 and u_2 such that

$$\delta(q_0, u_1) = \delta(q_0, u_2).$$

Denote this common state by q. It is clear that

$$\lambda(q, \alpha^{|f(u_1)|}) = f(u_1) \qquad \text{and} \qquad \lambda(q, \alpha^{|f(u_2)|}) = f(u_2)\,.$$

If $|f(u_1)| = |f(u_2)|$ then we at once obtain $f(u_1) = f(u_2)$.

In the opposite case we may assume that $|f(u_1)| < |f(u_2)|$ and let $k = |f(u_2)| - |f(u_1)|$. By definition of $\widehat{f}$ we have

$$\widehat{f} : u\alpha^{|f(u)|}\alpha \mapsto \alpha^{|u|}f(u)\alpha$$

and in particular:

$$\lambda(q, \alpha^{|f(u_1)|}\alpha^k) = f(u_1)\alpha^k\,,$$
$$\lambda(q, \alpha^{|f(u_2)|}) = f(u_2)\,.$$

From $\alpha^{|f(u_1)|}\alpha^k = \alpha^{|f(u_2)|}$ we obtain an equality

$$f(u_1)\alpha^k = f(u_2) \in X^*$$

which is true only with $k = 0$ and $f(u_1) = f(u_2)$.

Although $f(u) \mapsto \delta(q, u)$ need not to be a function (since a nonminimal automaton can include several paths to output $f(u)$), from above discussion it follows that for $f(u_1) \neq f(u_2)$ we have $\delta(q_0, u_1) \neq \delta(q_0, u_2)$. Thus the set $f(X^*)$ is finite. □

3.2. Example. Let $\mathcal{P}$ denote the set of prime numbers. Consider an alphabet $X = \{1, 2\}$ and a function

$$f(u) = \begin{cases} \varepsilon & u = \varepsilon, \\ 1 & |u| \in \mathcal{P}, \\ 2 & |u| \notin \mathcal{P}. \end{cases}$$

The extension mapping $\widehat{f}$ satisfies

$$\widehat{f} : u\alpha \mapsto \begin{cases} \alpha^{|u|}1 & |u| \in \mathcal{P}, \\ \alpha^{|u|}2 & |u| \notin \mathcal{P}. \end{cases}$$

The automaton defined by $\widehat{f}$ cannot be finite. Indeed, were automaton inducing $\widehat{f}$ be finite then by taking

$$Q = \{\delta(q_0, u) : |u| \in \mathcal{P}\}$$

as the set of an accept states we obtain a contradiction since $L = \{u \in X^* : |u| \in \mathcal{P}\}$ is not regular.

3.3. Proof of the main theorem.
Both conditions state that a function f has the form

$$f(u) = \begin{cases} u_1 & u \in L_1, \\ \dots & \\ u_k & u \in L_k, \end{cases}$$

where L_i, $i = 1, \dots, k$ are regular languages satisfying

$$X^* = L_1 \cup L_2 \cup \dots \cup L_k\,, \qquad L_i \cap L_j = \emptyset\,.$$

(⇒) Let $\mathcal{A} = (Q, q_0, X_\alpha, \delta, \lambda)$ be finite automaton inducing $\widehat{f}$. From the previous lemma $f(X^*)$ is a finite set. Furthermore, the language $L = f^{-1}(u)$ is recognizable by the automaton $\mathcal{A}' = (Q, q_0, T, X, \delta')$ where the set of an accept states is taken as

$$T = \{\delta(q_0, v) : f(v) = u\}$$

and the next-state function δ' is obtained from δ by deleting arrows labeled with α on the input side of its label. Since $\mathcal{A}'$ is finite, thus L is regular.

(⇐) Let $k = |f(X^*)|$ and L_i, $i = 1, \ldots, k$ be pairwise distinct regular languages of the form $f^{-1}(u)$, $u \in X^*$. Then, for each i there exists a finite Rabin–Scott automaton $\mathcal{A}_i$ which accepts the language L_i

$$\mathcal{A}_i = (Q_i, q_0^i, T_i, X, \delta_i)$$

where Q_i are finite sets and $1 \leq i \leq k$.

The automaton $\mathcal{A} = (Q, q_0, X, \delta, \lambda)$ defining $\widehat{f}$ can be carried from above by taking

1. $Q' = Q_1 \times \cdots \times Q_k$, where $Q' \subset Q$ and the remaining elements of Q are introduced below,
2. $q_0 = (q_0^1, \ldots, q_0^k)$,
3. $(q^1, \ldots, q^k) \overset{x/\alpha}{\mapsto} (r^1, \ldots, r^k)$ iff $\delta_i(q^i, x) = r^i$ for $1 \leq i \leq k$.

Since $\{L_i : i = 1, \ldots, k\}$ forms a partition on X^*, then obtained partial automaton has the property, that for every state $q = (q^1, \ldots, q^k)$ exactly one component is an accept state in the automaton corresponding to its position.

So far, for the partial automaton, the function δ does not accept symbol α as second argument and the automaton is not capable to produce symbols other than α. From earlier mentioned property, we claim that for every state $q = (q^1, \ldots, q^k)$ there exists corresponding language $L_q = \{u : \delta(q_0, u) = q\}$ and

$$|f(L_q)| = 1$$

that is all words from L_q are mapped into the same word, say $v = x_1 \ldots x_s$. After adding paths of the form

$$q \overset{\alpha/x_1}{\longmapsto} q_1 \overset{\alpha/x_2}{\longmapsto} \ldots \overset{\alpha/x_s}{\longmapsto} q_s\ , \qquad (*)$$

where $q_1, \ldots, q_s$ are new states, we have the finite partial automaton that follows the rule

$$u\alpha^{|f(u)|} \mapsto \alpha^{|u|} f(u)\,.$$

While δ is still the partial mapping, it needs to be extended in order to obtain well-defined plain or cyclic extension. This can be done while preserving finite amount of states. In both cases, consider paths of the form (*). Let q_s be a terminal state of the path. In case of the plain extension we define

$$q_s \stackrel{x'/\alpha}{\longmapsto} q_s, \qquad x' \in X_\alpha$$

and

$$q_i \stackrel{x/x_i}{\longmapsto} q_s, \qquad x \in X, \qquad i = 1, \ldots, s-1\,.$$

However, for the cyclic extension we apply

$$q_s \stackrel{\alpha/\alpha}{\longmapsto} q_s$$

and

$$q_i \stackrel{x/\alpha}{\longmapsto} \delta(q_0, x), \qquad x \in X, \qquad i = 1, \ldots, s\,.$$

Finally, for both plain and cyclic extensions we will apply

$$q_0 \stackrel{\alpha/\alpha}{\longmapsto} q_0\,.$$

Above definitions complete a construction of the well-defined automaton inducing the extension $\widehat{f}$.

□

Acknowledgments

Thanks to Vitalii Sushchanskii for his kindness and for invaluable help.

References

[1] Čulik K. II, *Construction of the Automaton Mapping*, (in Russian) Aplikace Matematiky, Vol. 10, No. 6, 1965

[2] Eilenberg S., *Automata, Languages and Machines,* Volume A, Academic Press, New York, 1974

[3] Glushkov V. M., *Abstract Theory of Automata*, (in Russian) Russian Math. Surveys, Vol. XVI, 5(101), 1961, pp. 3–62

[4] Grigorchuk R. I., Nekrashevich V. V., Sushchanskii V. I., *Automata, Dynamical Systems, and Groups*, Proceedings of Steklov Institute of Mathematics, Vol. 231, 2000, pp. 128–203

[5] Mikolajczak B. (ed.), *Algebraic and Structural Automata Theory*, Annals of Discrete Mathematics, Vol. 44, North-Holland, 1991

[6] Raney G.N., *Sequential Functions*, J. Assoc. Comput. Math., Vol. 5, 1958, N2, pp. 177–180

[7] Sheng Yu, *Regular Languages*, In: *Handbook of Formal Languages*, Volume 1, Springer-Verlag, Berlin, 1997, pp. 41–110

Contact information

Mirosław Osys

Silesian University of Technology,
Faculty of Mathematics and Physics,
ul. Kaszubska 23,
44-100 Gliwice, Poland
E-Mail: odys@zeus.polsl.gliwice.pl
URL: zeus.polsl.gliwice.pl/~odys

Received by the editors: 29.10.2004
and final form in 15.12.2005.