ON COMPLETE SYSTEMS OF AUTOMATA

DEDICATED TO PROFESSOR JOHN L. RHODES ON HIS 60TH BIRTHDAY

Pál Dömösi *
Institute of Mathematics and Informatics
L. Kossuth University
Egyetem tér 1
4032 Debrecen
Hungary
domosi@math.klte.hu

Chrystopher L. Nehaniv *,†
School of Computer Science & Engineering
University of Aizu
Aizu-Wakamatsu City 965-8580
Japan
nehaniv@u-aizu.ac.jp

ABSTRACT

A primitive product is a composition of a finite sequence of finite automata such that feedback is limited to no further than the previous factor. Furthermore, the input to each factor depends only on the global input to the system and the states of at most three factors (including the factor itself). Conversely, the state of a factor may directly influence only at most three factors (including the factor itself). Additional conditions guarantee a strong planarity property known as outerplanarity.

Any primitive product of automata can be realized in an outerplanar layout. This is desirable from the engineering point of view of simple circuit wiring as a circuit whose components and wires comprise the nodes and edges of an outerplanar graph may be realized on a two-dimensional surface, and moreover, new wires can be run from a point outside the circuit to any or all nodes of the circuit without crossing each other or any existing wires.

We constructively show that if $\mathcal{A}$ is a finite automaton satisfying Letichevsky's criterion, then any finite automaton can be homomorphically represented by (*i.e.* is a homomorphic image of a subautomaton of, or equivalently, is a letter-to-letter [length-preserving] divisor of) a primitive product of copies of $\mathcal{A}$.

A class $\mathcal{K}$ of finite automata is homomorphically complete under a given product π, by definition, if every finite automaton can be homomorphically represented as a π-product of automata from $\mathcal{K}$. By Letichevsky's characterization of homomorphically complete classes under the general product (unrestricted finite composition), our results imply that a class of finite automata is homomorphically complete under the general product if and only if it is homomorphically complete under the primitive product.

* This work was supported by grants of the University of Aizu "Algebra & Computation" and "Automata Networks" projects (R-10-1, R-10-4), the "Automata & Formal Languages" project of the Hungarian Academy of Sciences and Japanese Society for Promotion of Science (No. 15), the Hungarian National Foundation for Scientific Research (OTKA T019392), and the Higher Education Research Foundation of the Hungarian Ministry of Education (No. 222). † Current address: Interactive Systems Engineering, University of Hertfordshire, College Lane, Hatfield Herts AL10 9AB, United Kingdom.

1. Introduction

An important area of automata theory has been to investigate how automata can be realized with compositions of certain basic automata. One of the most celebrated results in the field of compositions of automata is the Letichevsky Decomposition Theorem. In this study, we give an extension of this result. While the Krohn - Rhodes Decomposition Theorem [15, 16, 1, 19] is a basis for studying the cascade product of automata, the fundamental information concerning homomorphically complete classes of finite automata under the Gluškov product [13] is concentrated in the well-known classical criterion of A. A. Letichevsky [18] (see Section 3 below). In order to decrease the complexity of the general product, F. Gécseg [8] introduced a family of semi-cascade products named α_i-products, where the index i is a nonnegative integer, which denotes the maximal admissible length of feedbacks.[1] Z. Ésik [6] proved that Letichevsky's criterion can be used to describe those classes which are homomorphically complete under the α_2-product. On the basis of this result, Z. Ésik and Gy. Horváth [7] showed that for every $i \geq 2$, the α_i-product is homomorphically as general as the Gluškov product.

P. Dömösi and B. Imreh [5] introduced another product hierarchy, the ν_i-hierarchy, where i is a positive integer. In a ν_i-product of automata, the working of each factor can be directly influenced by at most i of the factors. P. Dömösi and Z. Ésik [4] have proved that the ν_i-hierarchy is proper from the point of view of homomorphic representation. A comparison of the α_i-products and ν_i-products can be found in [10].

An $\alpha_i - \nu_j$-product is an α_i-product that is also a ν_j-product. Thus, *e.g.*, an $\alpha_0 - \nu_1$-product is a loop-free product with the additional property that each factor depends only on the general input and the state of at most one other factor. F. Gécseg and H. Jürgensen [11] proved that the $\alpha_0 - \nu_1$-product is homomorphically as general as the general product if infinite product is permitted. Since the ν_i-hierarchy is strict with respect to homomorphic representation, we cannot get similar general results for the $\alpha_i - \nu_j$-products provided that infinite products are not permitted.

In this paper, we introduce a new product of special type, called the *primitive product*, and we show that Letichevsky's criterion characterizes exactly those classes of automata which are homomorphically complete under this kind of product. Since the primitive product can be considered as a special form of the $\alpha_2 - \nu_3$-product, we have the rather surprising result that, contrary the fact that the ν_i-hierarchy is strict with respect to the homomorphic representation, the $\alpha_i - \nu_j$-hierarchy (and the ν_j-hierarchy) collapses at $i = 2$ and $j = 3$ whenever we are confined to homomorphically complete classes under these kinds of products (*i.e.* $(\alpha_i - \nu_j)$- resp. ν_j-).

Thus, when a class of finite automata satisfies Letichevsky's criterion, one can

[1] A feedback from a factor to itself is considered to be of length 1. Thus, in a sequence of automata, a feedback of length 2 is understood to be to the preceding factor.

build any finite automaton as a homomorphic image of a subautomaton[2] of a product of automata from that class, and conversely. But more strongly, one can guarantee that the working of each factor depends on at most three other factors such that one of these three factors is the factor itself and that feedback length is not more than two. Thus, the "wiring" of the representing automaton is quite restricted, and, furthermore, its factors and wires may be arranged on the nodes and edges of a highly restricted type of planar graph demanded by the definition of primitive product.

For all notions and notation not defined here, we refer to the books [14], [12] and [9].

2. Notation and Basic Notions

An *alphabet* is a finite nonvoid set X, the elements of X are called *letters.* For any alphabet X, let X^* denote the *free monoid* of all *words* over X (including the *empty word* λ). Moreover, denote by $X^+(= X^* \setminus \{\lambda\})$ the *free semigroup* of all non-empty words over X. The *length* of a word $p = x_1 \dots x_n \in X^+(x_1, \dots, x_n \in X)$ is denoted by $|p|(= n)$. The length of the empty word λ is zero by definition. The *reverse* of p is $\bar{p} = x_n \dots x_1$. For any alphabet X and nonnegative integer n, X^n denotes the set of n-length words of X^*. Moreover, we put $p^0 = \lambda, p^n = p^{n-1}p$ $(p \in X^*, n > 0)$. If $p = qr$ for some $q, r \in X^*$, then q is said to be a *prefix* and r a *suffix* of p. If there is no danger of confusion, we shall sometimes denote an n-tuple $(a_1, \dots, a_n)$ with each $a_i \in X$ by the word $a_1 \dots a_n$. Throughout this paper, for integers $k, n\,(n \geq 2)$, $k \pmod n$ denotes the least positive integer k' such that n divides $k - k'$. (In particular, $0 \pmod n = n$.)

By a (finite) *automaton* we mean a system $\mathcal{A} = (A, X, \delta)$, where A is a finite nonvoid set of *states,* X is the *input alphabet,* and the mapping $\delta : A \times X \to A$ is the *transition function* of $\mathcal{A}$. We extend δ to a mapping $\delta : A \times X^* \to A$ in the following way: for arbitrary $a \in A, \delta(a, \lambda) = a$ and $\delta(a, px) = \delta(\delta(a, p), x)$ $(p \in X^*, x \in X)$. Then every $p \in X^*$ *induces* (under δ) a transformation $\gamma_p : A \to A$ of the state set A with $\gamma_p(a) = \delta(a, p)$ for each $a \in A$.

Let $\mathcal{A} = (A, X, \delta), \mathcal{B} = (B, Y, \delta')$ be automata. $\mathcal{B}$ is a *subautomaton* of $\mathcal{A}$ if $B \subseteq A, Y \subseteq X$, and $\delta'(b, y) = \delta(b, y)$ $(b \in B, y \in Y)$. $\psi = (\psi_1, \psi_2)$ with the surjective mappings $\psi_1 : A \to B, \psi_2 : X \to Y$ is a *homomorphism* of $\mathcal{A}$ onto $\mathcal{B}$ if $\psi_1(\delta(a, x))$ $= \delta'(\psi_1(a), \psi_2(x))$ $(a \in A, x \in X)$. If there exists a homomorphism of $\mathcal{A}$ onto $\mathcal{B}$, then we also say that $\mathcal{A}$ *can be mapped homomorphically* onto $\mathcal{B}$. The automaton $\mathcal{B}$ *can be represented homomorphically* by the automaton $\mathcal{A}$ if $\mathcal{A}$ has a subautomaton which can be mapped homomorphically onto $\mathcal{B}$. If the component mappings are bijective, then we speak of *isomorphism.* We also say that the automaton $\mathcal{B}$ *can be embedded isomorphically* into the automaton $\mathcal{A}$ if $\mathcal{B}$ is isomorphic to a subautomaton of $\mathcal{A}$. In addition, if $X = Y$ and $\psi_2 : X \to Y$ is the identity mapping, then we also refer to ψ_1

[2] In the terminology of the Krohn-Rhodes school, "homomorphic image of a subautomaton" is equivalent to "length preserving divisor", and to "cover with letter-to-letter lifting", where the input symbols (letters) are considered as generators of (possibly non-faithful) transformation semigroups.

as a homomorphism (or an isomorphism).

Let $f : X_1 \times \ldots \times X_n \to X$ be a mapping having n variables for some positive integer n, moreover, let $t \in \{1, \ldots, n\}$. f is said to *really depend* on its t^{th} variable if there exist $x_1 \in X_1, \ldots, x_{t-1} \in X_{t-1}, x_t, x_t{}' \in X_t, x_{t+1} \in X_{t+1}, \ldots, x_n \in X_n$ having $f(x_1, \ldots, x_n) \neq f(x_1, \ldots, x_{t-1}, x_t{}', x_{t+1}, \ldots, x_n)$. If f does not have this property, then we also say that f is *really independent* of its t^{th} variable. Moreover, if there is no danger of confusion then sometimes we leave the attribute "really".

Let $\mathcal{A}_t = (A_t, X_t, \delta_t)$ $(t = 1, \ldots, n, n \geq 1)$ be automata. Take an alphabet X and a system of *feedback functions* $\varphi_t : A_1 \times \ldots \times A_n \times X \to X_t$ $(t = 1, \ldots, n)$. Let $\mathcal{A} = (A, X, \delta) = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n(X, \varphi_1, \ldots, \varphi_n)$ be the automaton with $A = A_1 \times \ldots \times A_n$, $\delta((a_1, \ldots, a_n), x) = (\delta_1(a_1, \varphi_1(a_1, \ldots, a_n, x)), \ldots, \delta_n(a_n, \varphi_n(a_1, \ldots, a_n, x))$ $((a_1, \ldots, a_n) \in A, x \in X)$. The automaton $\mathcal{A}$ is called the *(general) product* (or *Gluškov product*) of $\mathcal{A}_1, \ldots, \mathcal{A}_n$ (with respect to X and $\varphi_1, \ldots, \varphi_n$). If $\mathcal{A}_1 = \ldots = \mathcal{A}_n = \mathcal{B}$, then $\mathcal{A}$ is a *(general) power* of $\mathcal{B}$.

Proposition 2.1. *Let $\mathcal{A} = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n(X, \varphi_1, \ldots, \varphi_n)$ be a product of automata $\mathcal{A}_t = (A_t, X_t, \delta_t)$, $t = 1, \ldots, n$ and consider a permutation P over $\{1, \ldots, n\}$. Define the product $\mathcal{A}' = \mathcal{A}'_1 \times \ldots \times \mathcal{A}'_n(X, \varphi'_1, \ldots, \varphi'_n)$ such that $\mathcal{A}'_t = \mathcal{A}_{P(t)}$, moreover, for any state $(a_{P(1)}, \ldots, a_{P(n)}) \in A_{P(1)} \times \ldots \times A_{P(n)}$ and input letter $x \in X$, $\varphi'_t(a_{P(1)}, \ldots, a_{P(n)}, x) = \varphi_{P(t)}(a_1, \ldots, a_n, x)$, $t = 1, \ldots, n$. Then $\mathcal{A}$ is isomorphic to $\mathcal{A}'$.* □

Let us consider a class $\mathcal{K}$ of automata. We say $\mathcal{K}$ is *homomorphically complete* under the Gluškov product if every finite automaton can be homomorphically represented by a Gluškov product of automata from $\mathcal{K}$. Homomorphic completeness under any of various other products is defined analogously. We define the *underlying graph* $\mathcal{D} = (V, E)$ $(V = \{1, \ldots, n\}, E \subseteq V \times V)$ of $\mathcal{A}$ such that $(i, j) \in E$ if and only if the feedback function φ_j really depends on its i^{th} state variable. Thus, an underlying graph is a directed graph (or, in short, a digraph) which may contain loop edges. Moreover, if $(i, j) \in E$ then it is said that (i, j) is an *outgoing edge* of i, and simultaneously, (i, j) is an *incoming edge* for j. (In this way, a loop edge (i, i) has both of these properties concerning the vertex i.) We shall use the following statement.

Proposition 2.2. *Let $\mathcal{A} = (A_1 \times \ldots \times A_n, X, \delta_{\mathcal{A}}) = \mathcal{A}_1 \times \ldots \times \mathcal{A}_n(X, \varphi_1, \ldots, \varphi_n)$ be a product of automata having an underlying graph $\mathcal{D} = (\{1, \ldots, n\}, E)$, vertices i, j, k with $i < j, k$ such that, $(i, j), (i, k) \in E$. Suppose that for every pair ℓ, m, $\ell \leq i < m$ implies $(m, \ell) \notin E$. Then there exists a product $\mathcal{A}' = (A_1 \times \ldots \times A_i \times A_1 \times \ldots \times A_n, X, \delta_{\mathcal{A}'})$ $= \mathcal{A}_1 \times \ldots \times \mathcal{A}_i \times \mathcal{A}_1 \times \ldots \times \mathcal{A}_n(X, \varphi'_1, \ldots, \varphi'_{i+n})$ having the underlying graph with nodes $\{1, \ldots, i+n\}$ and edges $(\{(i+u, i+v) \mid (u, v) \in E\} \setminus \{(2i, i+j)\}) \cup \{(i, i+j)\} \cup \{(u, v) \mid (u, v) \in E, u, v \leq i\})$ such that, for any $a_1 \in A_1, \ldots, a_n \in A_n, x \in X$,*

$$\delta_{\mathcal{A}'}((a_1, \ldots, a_i, a_1, \ldots, a_n), x) = (a'_1, \ldots, a'_i, a'_1, \ldots, a'_n),$$

whenever

$$\delta_{\mathcal{A}}((a_1,\ldots,a_n),x) = (a'_1,\ldots,a'_n).$$

Proof: By the condition on edges, $\varphi_1,\ldots,\varphi_i$ do not depend on their $(i+1)^{th},\ldots,n^{th}$ state components.

Fix any arbitrary $a'_{i+1}\in A_{i+1},\ldots,a'_n\in A_n$. We construct the following feedback functions:

$$\varphi'_t(a_1,\ldots,a_{i+n},x) = \begin{cases} \varphi_t(a_1,\ldots,a_i,a'_{i+1},\ldots a'_n,x) & \text{if } t\le i,\\ \varphi_{t-i}(a_{i+1},\ldots,a_{i+n},x) & \text{if } t>i \text{ and } t\ne i+j,\\ \varphi_j(a_{i+1},\ldots,a_{2i-1},a_i, & \\ \qquad a_{2i+1},\ldots,a_{i+n},x) & \text{if } t=i+j \end{cases}$$

(where $t=1,\ldots,i+n,(a_1,\ldots,a_{i+n})\in A_1\times\ldots\times A_i\times A_1\times\ldots\times A_n, x\in X$).

It is easy to check that the product $\mathcal{A}'$ having the above feedback function components satisfies the required conditions. □

Corollary 2.3. *Every cascade of automata can be homomorphically represented by a cascade of (copies of the same) automata such that: For each i, at most one feedback function φ_j really depends on the state of $\mathcal{A}_i$. (Also, the analogous statement holds for the α_1-product).* □

Several families of products can be derived from the general product by imposing restrictions on the feedback dependency. $\mathcal{A}$ is an α_i-*product* $(i=0,1,\ldots)$ if each φ_t $(t=1,\ldots,n)$ is really independent of its j^{th} state component $(j=1,\ldots,n)$ whenever $j\ge t+i$. Especially, if $\mathcal{A}$ is an α_0-product, then we often give the system of feedback functions in the form $\varphi_1: X\to X_1, \varphi_2: A_1\times X\to X_2,\ldots,\varphi_n: A_1\times\ldots\times A_{n-1}\times X\to X_n$. If i is a positive integer for which every φ_t $(t=1,\ldots,n)$ really depends on not more than i state variables, then $\mathcal{A}$ is a ν_i-*product.* In addition, an $\alpha_i-\nu_j$-*product* $(i=0,1,\ldots,j=1,2,\ldots)$ is an α_i-product that is also a ν_j-product.

Our fundamental concept is that of the *primitive product.* Take the above considered general product $\mathcal{A}=\mathcal{A}_1\times\ldots\times\mathcal{A}_n(X,\varphi_1,\ldots,\varphi_n)$ and its underlying graph $\mathcal{D}=(V,E)$. For any $t\in V$, denote by $i(t)$ and $o(t)$ the sets of incoming and outgoing edges of t, respectively, and assume that

(i) for any $t\in V$ there exist $j,k\in\{1,\ldots,t-1,t+1\}$ and $r\in\{t-1,t+1,\ldots,n\}$ such that one of the following conditions is satisfied.

(i1) $i(t)\subseteq\{(j,t),(t,t)\}$ and $o(t)\subseteq\{(t,t-1),(t,t),(t,r)\}$,

or

(i2) $i(t)\subseteq\{(j,t),(k,t),(t,t)\}$ and $o(t)\subseteq\{(t,t),(t,r)\}$;

(ii) if $(a,b),(c,d)\in E$ and $\{a,b\}\cap\{c,d\}=\emptyset$, then $\min\{c,d\}<a<\max\{c,d\}$ if and only if $\min\{c,d\}<b<\max\{c,d\}$.

Then we say that $\mathcal{A}$ is a *primitive product.* For any class $\mathcal{K}$ of automata, let us consider the class $P(\mathcal{K})$ of primitive products having factors from $\mathcal{K}$. It is easy to see that $P(P(\mathcal{K})) \subseteq P(\mathcal{K})$ does not hold in general. But, we have the following.

Proposition 2.4. *Let $\mathcal{M} = \mathcal{M}_1 \times \ldots \times \mathcal{M}_{n+1}(X, \psi_1, \ldots, \psi_{n+1}), n \geq 1$ be a product of primitive products $\mathcal{M}_i = \mathcal{M}_{i,1} \times \ldots \times \mathcal{M}_{i,j_i}(X_i, \psi_{i,1}, \ldots, \psi_{i,j_i}), j_i \geq 2, i = 1, \ldots, n+1$ having the following properties. $\psi_1, \ldots, \psi_n$ may really depend only on their input variables. Moreover, $\psi_{i,1}, \ldots, \psi_{i,j_i-1}, i = 1, \ldots, n$ really do not depend on their last (i.e. j_i^{th}) state variables, and, if some $\psi_{n+1,k}$ $(k = 1, \ldots, j_{n+1})$ really depends on its input variable, then it may additionally depend only on its k^{th} state variable and at most one other state variable, and simultaneously, there exists at most one $\psi_{n+1,k'}$ $(k' = 1, \ldots, j_{n+1})$ with $k \neq k'$ depending on its k^{th} state variable. Furthermore, the input set of $\mathcal{M}_{n+1}$ is $X_{n+1} = M_{1,j_1} \times M_{2,j_2} \times \ldots \times M_{n,j_n}$, where $M_{i,j_i}, i = 1, \ldots, n$ denotes the state set of the last factor of the product $\mathcal{M}_i$, and each $\psi_{n+1,k}$ $(k = 1, \ldots, j_{n+1})$ may depend at most on one component of X_{n+1}; and moreover $\psi_{n+1,k}$ and $\psi_{n+1,k'}$ do not depend on the same component of X_{n+1} for $k \neq k'$ $(k, k' = 1, \ldots, j_{n+1})$.*

If ψ_{n+1} has the form $\psi_{n+1}(m_1, \ldots, m_n, m_{n+1}, x) = (m_{1,j_1}, \ldots, m_{n,j_n}) \in X_{n+1}$, where m_i is the state of $\mathcal{M}_i$ and m_{i,j_i} the state of $\mathcal{M}_i$'s last factor, then $\mathcal{M}$ is isomorphic to a primitive product of the $\mathcal{M}_{i,j}, i = 1, \ldots, n+1, j = 1, \ldots, j_i$.

Proof: Let P be an arbitrary permutation over $\{1, \ldots, n\}$. Considering the short notation $\mathcal{N}_\ell = \mathcal{M}_{\ell,1} \times \ldots \times \mathcal{M}_{\ell,j_\ell}$ $(\ell = 1, \ldots, n+1)$, by Proposition 2.1, we can construct the product

$$\mathcal{M}' = \mathcal{N}_{P(1)} \times \ldots \times \mathcal{N}_{P(n)} \times \mathcal{N}_{n+1}(X, \varphi'_1, \ldots, \varphi'_u)$$

with $u = j_1 + \ldots + j_{n+1}$ such that $\mathcal{M}'$ is isomorphic to $\mathcal{M}$. Denote $\psi_{n+1,s_1}, ..., \psi_{n+1,s_r}$ with $s_1 < \ldots < s_r$ all feedback functions of the product $\mathcal{M}_{n+1}$ depending at least one component of the input set $X_{n+1} = M_{1,j_1} \times M_{2,j_2} \times \ldots \times M_{n,j_n}$. From our assumptions it follows that $r \leq n$. Suppose that for every $\ell \in \{1, \ldots, r\}$, $P(n-\ell+1) = t$, whenever ψ_{n+1,s_ℓ} depends on the t^{th} component of X_{n+1}. Clearly, then $\mathcal{M}'$ forms a primitive product of the $\mathcal{M}_{i,j}, i = 1, \ldots, n+1, j = 1, \ldots, j_i$. □

The following statement is obvious.

Proposition 2.5. *Every primitive product is an $\alpha_2 - \nu_3$-product.* □

For any directed graph $\mathcal{D} = (V, E)$, we consider the associated undirected graph $U(\mathcal{D}) = (V, E')$ such that,

$$E' = \{\{i, j\} \mid (i, j) \in E\}.$$

Following Harary [14], we define a *walk* in an (undirected) graph (V, E') to be a sequence of vertices $v_0, \ldots, v_n$, such that $\{v_i, v_{i+1}\} \in E'$. A *path* is a walk with all

$n+1$ vertices distinct. A walk is *closed* if $v_0 = v_n$. A *cycle* in a graph is a closed walk such that its n points are distinct and $n \geq 3$. A *face* of a graph Γ embedded in the Euclidean plane $\mathbb{E}^2$ is the closure of a connected component of $\mathbb{E}^2 \setminus \Gamma$.

We say that a graph $\Gamma = (V, E)$ has the *ordered cycle property* if its nodes can be labelled with distinct positive integers such that if we identify each vertex with its label, then every cycle, considered as an undirected path, can be arranged in the form $c_1 < \ldots < c_k$, $(c_i \in V,\ i = 1, \ldots, k)$ with $(c_i, c_{i+1(\mathrm{mod}\,k)}) \in E$ or $(c_{i+1(\mathrm{mod}\,k)}, c_i) \in E$ edges.

Lemma 2.6. *Let $\mathcal{D} = (V, E)$ be the underlying graph of a primitive product of automata. Then $\mathcal{D}$ has the ordered cycle property.*

The nodes of $\mathcal{D}$ are already integers, so we consider $\mathcal{D}$ under its natural labelling.

Claim: Take any pair of undirected paths $i_1 \ldots i_m$, $j_1 \ldots j_n$ consisting of nodes in $\mathcal{D}$, with $j_1 < i_1 < j_n$ and suppose either $i_m < j_1$ or $j_n < i_m$. Then the paths contain a common point.

Proof of Claim: Assume that the claim is false, then there is a minimal counterexample, with all nodes *distinct* and $n + m$ least.

Consider i_{m-1}: if $i_{m-1} < j_1$ or $j_n < i_{m-1}$, then the path $i_1 \ldots i_{m-1}$ would yield smaller counterexample unless $m = 2$. If on the other hand, $j_1 < i_{m-1} < j_n$ then $i_{m-1} i_m$ yields a shorter counterexample unless $m = 2$. So $m = 2$ for a minimal counterexample. Now, consider the path $j_1 \ldots j_n$. If $i_2 < j_1$, then $i_2 < j_1 < i_1 < j_n$. In this case, by (ii), $i_2 < j_2 < i_1$ must hold, and thus, $j_2 \ldots j_n$ yields a shorter counterexample unless $n = 2$. If $j_n < i_2$, then $j_1 < i_1 < j_n < i_2$. Then, by (ii), $i_1 < j_{n-1} < i_2$ must hold, but in this case, $j_1 \ldots j_{n-1}$ yields a shorter counterexample.

We have established that $n = m = 2$ in any least counterexample. Thus, $i_2 < j_1 < i_1 < j_2$ or $j_1 < i_1 < j_2 < i_2$: now by condition (ii) of the definition of primitive product, since $j_1 = \min\{j_1, j_2\} < i_1 < \max\{j_1, j_2\} = j_2$, we have $j_1 < i_2 < j_2$, a contradiction. Therefore, no least counterexample can exist. This establishes the claim.

Now let c_1 denote the least node in the cycle. It is connected by edges in the cycle to two other nodes. Now these two nodes and c_1 are pairwise distinct. Let c_2 denote the lesser of the two and let c_k denote the greater. We have $c_1 < c_2 < c_k$. Proceeding around the cycle in the direction from c_1 to c_2 denote the nodes c_3, c_4, *et cetera* until we reach c_k. We assert that c_k is the greatest node in the cycle: if not, let c_i be the node with least i such that $c_i > c_k$. Note that $i \geq 3$. By leastness of i, $c_{i-1} < c_k$ and so it must be that $c_i > c_k > c_{i-1} > c_1$, but then the path $c_k c_1$ and the path $c_i c_{i-1}$ would comprise a counterexample to the claim. Hence, c_k must indeed be the greatest node.

Furthermore, it must be true for each $i = 1, \ldots, k-1$, that $c_{i+1} > c_i$: if not, take an i such that $c_{i+1} < c_i$, then we have $i \notin \{1, k-1\}$, and so $c_1 < c_{i+1} < c_i < c_k$. But then, $c_{i+1} \ldots c_k$ is a path disjoint from the path $c_1 \ldots c_i$, and we would have contradiction to the claim.

We have established that $c_1 < c_2 < \ldots < c_k$ for the nodes $c_1, c_2, \ldots, c_k$ met in sequence traced as we go around the cycle starting in the direction from c_1 to c_2. $\square$

A graph is called *outerplanar* if it has a planar embedding so that all its vertices lie on the same face. In this case, face may be taken to be the unbounded face. Outerplanarity is a strengthening of the notion of planarity, which has an analogous characterization in terms of forbidden subgraphs [17, 3, 14].

Corollary 2.7. *The underlying graph of any primitive product is an outerplanar graph.*

Proof: A graph is outerplanar if and only if it contains no subdivision of K_4, the complete graph on four nodes, and no subdivision of the complete bipartite graph $K_{2,3}$ [2], [14, p. 106]. But such a subdivision cannot have the ordered cycle property established in the lemma, since if it did, then by restriction the property would hold also for K_4 or $K_{2,3}$. But it is easy to check that K_4 and $K_{2,3}$ do not have this property. $\square$

Remark: As we see from the proofs of the Lemma 2.6 and Corollary 2.7, every product of automata whose underlying graph satisfies condition (ii) in the definition of primitive product has the ordered cycle property and an outerplanar underlying graph.

From the engineering point of view of circuit wiring, outerplanarity is an extremely desirable property, since a circuit whose components and wires comprise the nodes and edges of an outerplanar graph may be realized on a flat surface. Moreover, new wires can be run from a point outside the circuit to any or all nodes of the circuit without crossing each other or any of the existing wires.

3. Control Words

An automaton $\mathcal{A}$ *satisfies Letichevsky's criterion* if there are a state $a_0 \in A$, two input letters $x, y \in X$ and two input words $p, q \in X^*$ under which $\delta(a_0, x) \neq \delta(a_0, y)$ and $\delta(a_0, xp) = \delta(a_0, yq) = a_0$. If a class $\mathcal{K}$ of automata contains an automaton satisfying Letichevsky's criterion, then we also say that $\mathcal{K}$ *satisfies Letichevsky's criterion.* Otherwise, we say that $\mathcal{K}$ does not satisfy it.

Letichevsky Decomposition Theorem ([18]). *A class $\mathcal{K}$ of finite automata is homomorphically complete under the general product if and only if $\mathcal{K}$ satisfies Letichevsky's criterion.*

We will create "control words" for any automaton that satisfies Letichevsky's criterion. These will serve as logical signals in nearly all our further constructions.

Let $\mathbf{a} = a_0 \dots a_m$ and $\mathbf{b} = b_0 b_1 \dots b_n$ denote non-empty words over an alphabet A having the following properties.

(i) $a_0 = b_0$, the letters of $\mathbf{a}$ are pairwise distinct, the letters of $\mathbf{b}$ are pairwise distinct, and b_1 does not occur in $\mathbf{a}$.

(ii) if $\mathbf{a} = wxy$ and $\mathbf{b} = w'xy'$ for any factorizations with x a letter and w, w' non-empty, then $y = y'$ ($w, w' \in A^+$, $x \in A$, $y \in A^*$).

(iii) $m \le n$ (and $n > 0$). Equivalently, $|\mathbf{a}| \le |\mathbf{b}|$ (and $|\mathbf{b}| \ge 2$).

Given $\mathbf{a}$ and $\mathbf{b}$ as above, define *control words*, $\mathbf{u} = u_1 \dots u_s$ and $\mathbf{v} = v_1 \dots v_s$:

$$\text{(iv)}\quad u_1 \dots u_s = \begin{cases} {a_0}^{n+1} & \text{if } m = 0, \\ (a_1 \dots a_m a_0)^k & \text{if } m+1 \mid n+1, m \ne 0, n+1 = k(m+1), \\ a_1 \dots a_m a_0 b_1 \dots b_n a_0 & \text{if } m+1 \nmid n+1, \end{cases}$$

$$\text{(v)}\quad v_1 \dots v_s = \begin{cases} b_1 \dots b_n a_0 & \text{if } m+1 \mid n+1 \text{ (including the case } m = 0), \\ b_1 \dots b_n a_0 a_1 \dots a_m a_0 & \text{if } m+1 \nmid n+1. \end{cases}$$

The following lemma is obvious from (i) and (ii).

Lemma 3.1. *Given control words* $\mathbf{u}$, $\mathbf{v}$, *for all* $1 \le i, j \le s-1$ *we have:*
(i1) $u_i = u_j \ne a_0$ *implies* $u_{i+1} = u_{j+1}$,
(ii1) $v_i = v_j \ne a_0$ *implies* $v_{i+1} = v_{j+1}$,
(iii1) $u_i = v_j \ne a_0$ *implies* $u_{i+1} = v_{j+1}$. □

We next show

Lemma 3.2. *Let* $\mathcal{A} = (A, X, \delta)$ *be an automaton satisfying Letichevsky's criterion. There are states* $u_1, \dots, u_s, v_1, \dots, v_s$ $(\in A)$ *and input letters* $x'_1, \dots, x'_s, x''_1, \dots, x''_s$ $(\in X)$ *such that* $\delta(u_t, x'_t) = u_{t+1}, \delta(v_t, x''_t) = v_{t+1}$ $(t = 1, \dots, s-1), \delta(u_s, x'_s) = u_1, \delta(v_s, x''_s) = v_1$. *Moreover,* $\mathbf{u} = u_1 \dots u_s$ *and* $\mathbf{v} = v_1 \dots v_s$ *are control words.*

Proof: Consider an automaton $\mathcal{A} = (A, X, \delta)$ satisfying Letichevsky's criterion, *i.e.*, there are a state $a_0 \in A$, two input letters $x_1, y_1 \in X$ and two input words $p = x_2 \dots x_{m+1}, q = y_2 \dots y_{n+1} \in X^*$ $(x_2, \dots, x_{m+1}, y_2, \dots, y_{n+1} \in X)$ under which $\delta(a_0, x_1) \ne \delta(a_0, y_1)$ and $\delta(a_0, x_1 p) = \delta(a_0, y_1 q) = a_0$. Suppose that p and q have minimal length, *i.e.*, $\delta(a_0, x_1 p_1) = \delta(a_0, y_1 q_1) = a_0$ $(p_1, q_1 \in X^*)$ implies $|p| \le |p_1|$ and $|q| \le |q_1|$. Introduce the notations $a_u = \delta(a_0, x_1 \dots x_u)$ $(u = 1, \dots, m)$ and $b_v = \delta(a_0, y_1 \dots y_v)$ $(v = 1, \dots, n)$. Moreover, we set $b_0 = a_0$, $\mathbf{a} = a_0 \dots a_m$, and $\mathbf{b} = b_0 \dots b_n$.

Without loss of generality, we may assume $|p| = m \le n = |q|$. If p is the empty word $(m = 0)$, then $\delta(a_0, x_1) = a_0$ and q cannot be empty lest $\delta(a_0, y_1) = \delta(a_0, x_1)$. In any case, $n > 0$. This yields condition (iii).

If for every pair i $(= 1, \ldots, m), k$ $(= 1, \ldots, n)$ we have $a_i \neq b_k$, then we get condition (ii). (And, of course, b_1 does not occur in $\mathbf{a}$). By minimality, each of the state words $\mathbf{a} = a_0 \ldots a_m$ and $\mathbf{b} = b_0 \ldots b_n$ has then no repeated states letters. In other words, $a_i \neq a_j$ if $i \neq j$ $(i, j = 0, \ldots, m)$, and $b_k \neq b_\ell$ if $k \neq \ell$ $(k, \ell = 0, \ldots, n)$.

Otherwise, $a_i = b_k$ for some i $(= 1, \ldots, m)$, and k $(= 1, \ldots, n)$. We will take i to be the least such i, and k to be least such k for this i. (Observe $k = 1$ is not possible, for otherwise $|b_0 b_1 a_{i+1} \ldots a_m| = n + 1$ (by minimality); and then $n + 1 = m - i + 2$, whence $m - i + 1 \geq m$ implying $i \leq 1$, but then we would have $a_1 = b_1$, which is not the case.) So none of the states $a_1, \ldots, a_{i-1}$ is the same as any of the states $b_1, \ldots, b_{k-1}$. By minimality, $|x_{i+1} \ldots x_{m+1}| = |y_{k+1} \ldots y_{n+1}|$ since either of these words result in transition from $a_i = b_k$ back to a_0. Thus, we may replace $y_{k+1} \ldots y_{n+1}$ by $x_{i+1} \ldots x_{m+1}$ (or *vice versa*) to obtain condition (ii). Under this replacement, $\mathbf{a}$ and $\mathbf{b}$ are of unchanged minimal length, and so of course cannot contain repeated letters. We know $b_1 \notin \{a_0, \ldots a_{i-1}\}$ and $b_1 \notin \{b_k, \ldots b_n\} = \{a_i, \ldots, a_m\}$. Thus, b_1 does not occur in $\mathbf{a}$. Thus, conditions (i) and (ii) are established in every case.

Finally, we can define $\mathbf{u} = u_1 \ldots u_s$ and $\mathbf{v} = v_1 \ldots v_s$ as in (iv) and (v). The proof is complete. □

Using Lemma 3.1, we now prove the following technical lemma useful in establishing well-definedness of and performance of 'logical operations' with control words and inputs[3]:

Lemma 3.3. *For any alphabet X, control words $\mathbf{u}, \mathbf{v}$ over an alphabet A and any mapping $f : \{u_1, v_1\}^2 \times X \to \{u_1, v_1\}$ with $f(u_1, u_1, x) = u_1$ and $f(v_1, v_1, x) = v_1$ $(x \in X)$ there exists a mapping $g : A^3 \times X \to A$ satisfying:*

$$g(a, w_1, w_2, x) = \begin{cases} f(w_1, w_2, x) & \textit{if } a \in \{u_s, v_s\} \ (= \{a_0\}), w_1, w_2 \in \{u_1, v_1\}, \\ u_{j+1} & \textit{if } a = u_j, w_1, w_2 \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1, \\ v_{j+1} & \textit{if } a = v_j, w_1, w_2 \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1 \end{cases}$$

$((a, w_1, w_2, x) \in A^3 \times X)$.

Proof: Let $g(a, w_1, w_2, x)$ be any fixed element of A whenever $a \in \{u_i, v_i\} \setminus \{a_0\}$ with $\{w_1, w_2\} \not\subseteq \{u_{i+1}, v_{i+1}\}$ $(i = 1, \ldots, s-1)$ or $a \notin \{u_1, \ldots, u_s, v_1, \ldots, v_s\}$. Furthermore, in the case that $w_1, w_2 \in \{u_{j+1}, v_{j+1}\}$ and $a \neq a_0$, set $g(a, w_1, w_2, x) = u_{j+1}$ if $a = u_j$, and $g(a, w_1, w_2, x) = v_{j+1}$ if $a = v_j$. Taking into consideration Lemma 3.1, g is unambiguously determined on $(A \setminus \{a_0\}) \times A^2 \times X$ and has the values given in the statement of this lemma.

We still must extend g in a well-defined way to $\{a_0\} \times A^2 \times X$. That is, $a_0 \in \{u_i, v_i\} \cap \{u_j, v_j\}$, $\{w_1, w_2\} \subseteq \{u_{i+1(\operatorname{mod} s)}, v_{i+1(\operatorname{mod} s)}\}$, $\{w_1', w_2'\} \subseteq \{u_{j+1(\operatorname{mod} s)}, v_{j+1(\operatorname{mod} s)}\}$ and $(w_1, w_2) = (w_1', w_2')$ imply $g(a_0, w_1, w_2, x) = g(a_0, w_1', w_2', x)$ $(i, j = 1, \ldots s, x \in X)$.

[3]In the sequel $k (\operatorname{mod} m)$ denotes the least positive integer ℓ for which $m | k - \ell$.

We distinguish the following three cases.

Case 1. $m = 0$. We put

$$g(a_0, w_1, w_2, x) = \begin{cases} b_1 & \text{if } \{w_1, w_2\} \subseteq \{u_1, v_1\}(= \{a_0, b_1\}) \text{ and } f(w_1, w_2, x) = b_1, \\ a_0 & \text{otherwise} \end{cases}$$

$(w_1, w_2 \in A, x \in X)$.

Then we obtain $a_0 = u_1 = \ldots = u_s$ and $a_0 \notin \{v_1, \ldots, v_{s-1}\}$. On the other hand, $f(u_1, u_1, x) = u_1(= a_0)$ and $f(v_1, v_1, x) = v_1(= b_1)$ are supposed. Hence, our assertions hold, whenever $a = a_0$ and $\{w_1, w_2\} \neq \{u_1, v_1\}(= \{a_0, b_1\})$. Now we suppose $\{w_1, w_2\} = \{u_1, v_1\}(= \{a_0, b_1\})$. Then $g(a_0, w_1, w_2, x) = f(w_1, w_2, x)$, moreover, for every $j = 2, \ldots, s$ we have $\{w_1, w_2\} \nsubseteq \{u_j, v_j\}$. Therefore, we have our conditions. This ends the proof of Case 1.

Case 2. $m \neq 0, m + 1 \mid n + 1$, i.e., $n + 1 = k(m + 1)$ for some positive integer k. We set

$$g(a_0, w_1, w_2, x) = \begin{cases} b_1 & \text{if } \{w_1, w_2\} \subseteq \{u_1, v_1\} \text{ and } f(w_1, w_2, x) = b_1, \\ a_1 & \text{otherwise} \end{cases}$$

$(w_1, w_2 \in A, x \in X)$. Then $v_i \neq a_0$ if $i \in \{1, \ldots, s - 1\}$. Moreover, $u_i = a_0$ implies $u_{i+1} = a_1$ for any $i \in \{1, \ldots, s - 1\}$. Therefore, similarly to Case 1, we have our assertion if $i, j \in \{1, \ldots, s - 1\}$. If $\{w_1, w_2\} \subseteq \{u_1, v_1\}$ with $(w_1, w_2) \neq (u_1, u_1)$, then $b_1 \in \{w_1, w_2\}$. Hence, in this case, $\{w_1, w_2\} \nsubseteq \{u_z, v_z\}$ if $z > 1$. It is remained to study the case of $(w_1, w_2) = (u_1, u_1)$. Then we supposed $f(w_1, w_2, x) = u_1$ $(= a_1)$ corresponding to $g(a_0, a_1, a_1, x) = a_1$ $(x \in X)$. On the other hand, by $\{w'_1, w'_2\} \subseteq \{u_z, v_z\}, z > 1$ and $((a_1, a_1) =)(w_1, w_2) = (w'_1, w'_2)$, we have $g(a_0, w'_1, w'_2, x) = u_z$ $(x \in X)$ with $u_z = a_1$, whenever $a_0 \in \{u_{z-1}, v_{z-1}\}$ (or more precisely, whenever $a_0 = u_{z-1}$). This completes the proof of Case 2.

Case 3. $m + 1 \not| n + 1$. Define

$$g(a_0, w_1, w_2, x) = \begin{cases} a_1 & \text{if } \{w_1, w_2\} \subseteq \{u_{n+2}, v_{n+2}\}, \\ b_1 & \text{if } \{w_1, w_2\} \subseteq \{u_{m+2}, v_{m+2}\}, \\ f(w_1, w_2, x) & \text{if } \{w_1, w_2\} \subseteq \{u_1, v_1\} \end{cases}$$

$(w_1, w_2 \in A, x \in X)$. Then $u_1 = a_1, v_1 = b_1, u_{n+2} = b_{n-m+1}, v_{n+2} = a_1, u_{m+2} = b_1$, furthermore, $v_{m+2} = a_0$ or $v_{m+2} = b_{m+2}$ depending on whether $m+1 = n$ or $m+1 < n$.

By the property (i) of $a_0a_1 \ldots a_m$ and $b_0b_1 \ldots b_m$ (see their definition), we have, in order, $a_0 \notin \{a_1, b_1, b_{n-m+1}\}, a_1 \neq b_1$, and, if $m + 1 < n$, then $b_1 \neq b_{m+2}$. On the other hand, $m + 1 \not| n + 1$ implies $n \neq 2m + 1$ leading to $b_{m+2} \neq a_1$ (provided $m + 1 < n$) by the property (ii) of $a_0 \ldots a_m$ and $b_0b_1 \ldots b_n$ (see their definitions, again). Furthermore, $b_i = b_j$ $(i, j = 0, \ldots, n)$ implies $i = j$ by (i). Therefore, by $m + 1 < n$, $n \neq 2m + 1$ implies $b_{m+2} \neq b_{n-m+1}$, too. Similarly, since $m \leq n$ and $m + 1 \not| n + 1$, then $m < n$ which, in addition, shows $b_{n-m+1} \neq b_1$. But then $\{a_1, b_1\}, \{a_1, b_{n-m+1}\}, \{a_0, b_1\}$ by $m + 1 = n$ or $\{a_1, b_1\}, \{a_1, b_{n-m+1}\}, \{b_1, b_{m+2}\}$ by $m + 1 < n$ are pairwise different sets. Therefore, if $w_1 \neq w_2$ and $\{w_1, w_2\} \in \{\{u_1, v_1\}, \{u_{m+2}, v_{m+2}\}, \{u_{n+2}, v_{n+2}\}\}$, then our statement is valid, where the appropriate values of $g(a_0, w_1, w_2, x)$ $(x \in X)$ are, in order, $f(w_1, w_2, x), b_1, a_1$. (By the way, $a_1 = b_{n-m+1}$ is possible. In this case, we may leave the set $\{u_{n+2}, v_{n+2}\}$

$= \{a_1, b_{n-m+1}\}$ out of consideration whenever $w_1 \neq w_2$ is assumed.) Finally, if $w_1 = w_2$ then $f(u_1, u_1, x) = u_1$ and $f(v_1, v_1, x) = v_1$ $(x \in X)$ lead to $g(a_0, a_1, a_1, x)$ $= g(a_0, b_{n-m+1}, b_{n-m+1}, x) = a_1$ $(x \in X)$ and $g(a_0, b_1, b_1, x) = g(a_0, a_0, a_0, x) = b_1$ or $g(a_0, b_1, b_1, x) = g(a_0, b_{m+2}, b_{m+2}, x) = b_1$ $(x \in X)$ depending on whether $m+1 = n$ or $m+1 < n$. In other words, $g(a_0, u_1, u_1, x) = g(a_0, u_{n+2}, u_{n+2}, x) = g(a_0, v_{n+2}, v_{n+2}, x)$ $= u_1 = v_{n+2}$ and $g(a_0, v_1, v_1, x) = g(a_0, u_{m+2}, u_{m+2}, x) = g(a_0, v_{m+2}, v_{m+2}, x) = v_1$ $= u_{m+2}$. This completes the proof. □

Considering X as a singleton, we have the following consequence of Lemma 3.3.

Lemma 3.4. *For any mapping $f : \{u_1, v_1\}^2 \to \{u_1, v_1\}$ with $f(u_1, u_1) = u_1$ and $f(v_1, v_1) = v_1$ there exists a mapping $g : A^3 \to A$ satisfying:*

$$g(a, w_1, w_2) = \begin{cases} f(w_1, w_2) & \text{if } a \in \{u_s, v_s\} \ (= \{a_0\}), w_1, w_2 \in \{u_1, v_1\}, \\ u_{j+1} & \text{if } a = u_j, w_1, w_2 \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1, \\ v_{j+1} & \text{if } a = v_j, w_1, w_2 \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1 \end{cases}$$

$((a, w_1, w_2) \in A^3)$. □

Lemma 3.4 leads to the following statement.

Lemma 3.5. *There exists a mapping $g : A^2 \to A$ satisfying:*

$$g(a, b) = \begin{cases} b & \text{if } a \in \{u_s, v_s\} \ (= \{a_0\}), b \in \{u_1, v_1\}, \\ u_{j+1} & \text{if } a = u_j, b \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1, \\ v_{j+1} & \text{if } a = v_j, b \in \{u_{j+1}, v_{j+1}\}, j = 1, \ldots, s-1 \end{cases}$$

$((a, b) \in A^2)$. □

We close this section with the following definitions. Let $\mathcal{A} = (A, X, \delta)$ be an automaton satisfying Letichevsky's criterion. Moreover, let $\mathbf{u} = u_1 \ldots u_s, \mathbf{v}$ $= v_1 \ldots v_s$ $(\in A^*)$ be control words as constructed in Lemma 3.2 such that for appropriate input letters $x'_1, \ldots, x'_s, x''_1, \ldots, x''_s$ $(\in X)$ we have $\delta(u_t, x'_t) = u_{t+1}, \delta(v_t, x''_t)$ $= v_{t+1}$ $(t = 1, \ldots, s-1)$, $\delta(u_s, x'_s) = u_1, \delta(v_s, x''_s) = v_1$. For any $a, a', a'' \in A$ and fixed pair $u_1 \ldots u_s$, $v_1 \ldots v_s$ of control words we shall use the following operations on the alphabet A:

$$x[a, a']$$

$$= \begin{cases} \text{an arbitrary fixed } x \in X \text{ with } \delta(a, x) = a' & \text{if } (a, a') \in \{(u_s, u_1), (u_s, v_1)\}, \\ \text{an arbitrary fixed } x \in X \text{ with } \delta(a, x) = u_{i+1} & \text{if } (a, a') \in \{(u_i, u_{i+1}), (u_i, v_{i+1})\}, \\ & \quad i = 1, \ldots, s-1, \\ \text{an arbitrary fixed } x \in X \text{ with } \delta(a, x) = v_{i+1} & \text{if } (a, a') \in \{(v_i, u_{i+1}), (v_i, v_{i+1})\}, \\ & \quad i = 1, \ldots, s-1, \\ \text{any fixed element of } X, & \text{otherwise,} \end{cases}$$

$$x[a, a' \vee a''] = \begin{cases} x[a, a'] & \text{if } (a, a') = (u_s, v_1), \\ x[a, a''], & \text{otherwise,} \end{cases}$$

$$x[a, a' \wedge a''] = \begin{cases} x[a, a'] & \text{if } (a, a') = (u_s, u_1), \\ x[a, a''], & \text{otherwise.} \end{cases}$$

Using Lemma 3.5, it is clear that $x[a, a']$, $x[a, a' \vee a'']$ and $x[a, a' \wedge a'']$ are unambiguously defined.

4. Auxiliary Results

Take two alphabets X and Y. Let $n = \ell s$ ($\ell > 1$) be a fixed integer and consider a mapping $\tau : X^n \to Y^n$ having the property: $\{\tau(p) \mid p \in X^n\} \subseteq \{\mathbf{w} \mid \mathbf{w} \in \{\mathbf{u}, \mathbf{v}\}^\ell\}$ ($n = \ell s$) for some fixed words $\mathbf{u}, \mathbf{v} \in Y^s$. We shall denote the reverse of $\tau(p)$ by $\bar{\tau}(p)$.

Set $H \subseteq \{p \in X^+ \mid |p| = n\}, H \neq \emptyset$. Define $\mathcal{R}_{\tau,H,d} = (R_{\tau,H,d}, X, \delta_{\tau,H,d})$ be the automaton, where d is a positive integer, $R_{\tau,H,d} = \{(k, p, q) \in \{1, \ldots, n\} \times X^* \times Y^+ \mid k + |q| = n + d,\ |p| \in \{0, k\},\ p$ is a prefix of a word in H ($pp' \in H$ for some $p' \in X^*$), furthermore, $q = q'q''$, where q' is a suffix of $\mathbf{u}$ or $\mathbf{v}$ and $q'' \in \{\mathbf{u}, \mathbf{v}\}^*\}$, and, for arbitrary $(k, p, yq) \in R_{\tau,H,d}$ ($y \in \{u_t, v_t \mid t = 1, \ldots, s\}$) and $x \in X$,

$$\delta_{\tau,H,d}((k, p, yq), x) = \begin{cases} (k+1, px, q) & \text{if } k < n, p \neq \lambda, \text{ and } pxp' \in H \text{ for some } p' \in X^*, \\ (k+1, \lambda, q) & \text{if } k < n, p = \lambda, \\ & \text{or } k < n, p \neq \lambda, \text{ and for all } p' \in X^*, pxp' \notin H, \\ (1, x, q\tau(p)) & \text{if } k = n, p \in H, \text{ and } xp' \in H \text{ for some } p' \in X^*, \\ (1, \lambda, q\tau(p)) & \text{if } k = n, p \in H, \text{ and for any } p' \in X^*, xp' \notin H, \\ (1, x, q\mathbf{u}^\ell) & \text{if } k = n, p = \lambda, \text{ and } xp' \in H \text{ for some } p' \in X^*, \\ (1, \lambda, q\mathbf{u}^\ell) & \text{if } k = n, p = \lambda, \text{ and for any } p' \in X^*, xp' \notin H. \end{cases}$$

In order to simplify the proof of the next result, we introduce some auxiliary notions. Let $\mathcal{A} = (A, X_\mathcal{A}, \delta_\mathcal{A})$ be an automaton satisfying Letichevsky's criterion, and let $u_1 \ldots u_s, v_1 \ldots v_s$ be any pair of its control words. For any $w_1 \ldots w_s$ with $w_t \in \{u_t, v_t\}$ ($t = 1, \ldots, s$) we shall use the short notation $\mathbf{w}$. Consider a word $a_1 \ldots a_n \in A^+$ and an integer k ($= 1, \ldots, n$). We will denote by $c(a_1 \ldots a_n, k)$ the $(k+1)^{th}$ cyclic permutation of $a_1 \ldots a_n$. In more details, let

$$c(a_1 \ldots a_n, k) = \begin{cases} a_{k+2} \ldots a_n a_1 \ldots a_{k+1} & \text{if } k < n - 1, \\ a_1 \ldots a_n & \text{if } k = n - 1, \\ a_2 \ldots a_n a_1 & \text{if } k = n. \end{cases}$$

In addition, for any pair t, k with $t = \pm 1, \pm 2, \ldots, k = 1, \ldots, n$, let $c(a_1 \ldots a_n, nt + k) = c(a_1 \ldots a_n, k)$ and for any integer r, denote by $\overline{c}(a_1 \ldots a_n, r)$ the reverse of $c(a_1 \ldots a_n, r)$.

Let $\mathcal{M} = (W \times Z, X_{\mathcal{M}}, \delta_{\mathcal{M}})$ be an automaton with $Y \subseteq Z$, moreover, let $\mathcal{B} = (B, X, \delta_{\mathcal{B}})$ be a subautomaton of $\mathcal{M}$ having a homomorphism $\psi : B \to R_{\tau,H,d}$ onto $\mathcal{R}_{\tau,H,d}$ such that $\psi((w,z)) = (k,p,yq)$ $((w,z) \in B, (k,p,yq) \in R_{\tau,H,d},\ y \in Y)$ implies $z = y$. Then we say that $\mathcal{M}$ *y-represents* $\mathcal{R}_{\tau,H,d}$ (with respect to ψ).

We have the following

Lemma 4.1. *Let $\mathcal{A} = (A, X_{\mathcal{A}}, \delta_{\mathcal{A}})$ be an automaton satisfying Letichevsky's criterion and let $u_1 \ldots u_s, v_1 \ldots v_s$ be any pair of its control words. Consider an alphabet X, a multiple n of s with $n = \ell s, \ell > 1$, a word $r \in X^n$, and a mapping $\tau : X^n \to A^n$ having the property $\tau(p) \in \{\mathbf{u}, \mathbf{v}\}^\ell$ for each $p \in X^n$. Then there exists a primitive power $\mathcal{M}$ of $\mathcal{A}$ such that $\mathcal{R}_{\tau,\{r\},1}$ is y-represented by $\mathcal{M}$. In addition, apart from the feedback functions for the last factor, the feedback functions of the factors of $\mathcal{M}$ really do not depend on their last state variable.*

Proof: First we define the product $\mathcal{N} = \mathcal{A}^{3n+1}(X, \varphi_1, \ldots, \varphi_{3n+1})$ such that for any $(a_1, \ldots, a_{3n+1}) \in A^{3n+1}, x \in X$ and $t \in \{1, \ldots, 3n+1\}$, we have

$$\varphi_t(a_1, \ldots, a_{3n+1}, x)$$

$$= \begin{cases} x[a_t, a_{t+1}] & \text{if } t = 1, \ldots, s-1, \\ & \text{or } t = s+1, \ldots, s+n-1, \\ x[a_t, a_1] & \text{if } t = s, \\ x[a_t, a_{s+1}] & \text{if } t = s+n, \\ x[a_t, u_1] & \text{if } t = s+n+1, \ldots, s+2n, a_t = u_s (= v_s), (a_{t-1}, \\ & \quad a_{1-t (\mathrm{mod}\, s)}) = (u_1, u_1), \\ & \text{or } t = s+n+1, \ldots, s+2n,\ a_t = u_s (= v_s),\ (a_{t-1}, \\ & \quad a_{1-t (\mathrm{mod}\, s)}) = (v_1, u_1), \text{ and } x \text{ is not the } t-(s+n)^{\mathrm{th}} \\ & \quad \text{letter of } r, \\ x[a_t, v_1] & \text{if } t = s+n+1, \ldots, s+2n,\ a_t = u_s (= v_s), (a_{t-1}, \\ & \quad a_{1-t (\mathrm{mod}\, s)}) = (v_1, u_1), \text{ and } x \text{ is the } t-(s+n)^{\mathrm{th}} \\ & \quad \text{letter of } r, \\ x[a_t, a_{t-1}] & \text{if } t = s+n+1, \ldots, s+2n,\ a_t \in \{u_i, v_i\}, a_{t-1}, a_{1-t (\mathrm{mod}\, s)} \\ & \quad \in \{u_{i+1}, v_{i+1}\}, i = 1, \ldots, s-1, \\ x[a_t, a_s] & \text{if } t = 2n+s+1, \text{ and the } s^{\mathrm{th}} \text{ letter of } \bar{\tau}(r) \text{ is } u_1, \\ x[a_t, a_{s+2n} \vee a_s] & \text{if } t = 2n+s+1, \text{ and the } s^{\mathrm{th}} \text{ letter of } \bar{\tau}(r) \text{ is } v_1, \\ x[a_t, a_{t-1}] & \text{if } t = 2n+js+1, j = 2, \ldots, \ell, \text{ and the } js^{\mathrm{th}} \\ & \quad \text{letter of } \bar{\tau}(r) \text{ is } u_1, \\ & \text{or } t = 2n+js+i, j = 1, \ldots, \ell-1, i = 2, \ldots, s, \\ x[a_t, a_{s+2n} \vee a_{t-1}] & \text{if } t = 2n+js+1,\ j = 2 \ldots, \ell, \text{ and the } js^{\mathrm{th}} \\ & \quad \text{letter of } \bar{\tau}(r) \text{ is } v_1. \end{cases}$$

The first s factors provide a "small clock" in which $\mathbf{u} = u_1 \dots u_s$ cycles. The next n factors $(s+1, \dots, s+n)$ comprise a "big clock" in which $\mathbf{v}\mathbf{u}^{\ell-1}$ cycles. The next n factors $(s+n+1, \dots, s+2n)$ comprise a buffer into which values flow from the big clock, starting with v_1. At the k^{th} position of the buffer, if the input letter x matches the k^{th} letter of r when the signal (headed by v_1) is about to reach this position, then the signal is permitted to continue, otherwise instead of switching to state v_1 we switch to u_1 indicating rejection of the input.[4] Finally, if the word has not been rejected by the n^{th} input step, the acceptance signal reaches the end of the buffer, then the buffer contains n letters which are the reverse of $\mathbf{v}\mathbf{u}^{\ell-1}$ with $a_{2n+s} = v_1$ (and of $\mathbf{u}^{\ell}$ otherwise with $a_{2n+s} = u_1$).

In the next step, the buffer cycle starts again, while in the last $n - s + 1$ factors $(2n + s + 1, \dots, 3n + 1)$, the coded word $\tau(r)$ begins to take form if the signal has arrived. Now $\tau(r) = \mathbf{w}_1 \dots \mathbf{w}_\ell$, where each $\mathbf{w}_j \in \{\mathbf{u}, \mathbf{v}\}$. For each $j = 1, \dots, \ell$ with $\mathbf{w}_{\ell-j+1} = \mathbf{v}$ in this step v_1 simultaneously enters factor $2n + js + 1$, while for the j with $\mathbf{w}_{\ell-j+1} = \mathbf{u}$ and u_1 enters this factor. It is important to observe that $\tau(r)$ can be fully recovered from the states of these ℓ nodes at this time, as follows from $u_1 \neq v_1$ and the form of $\tau(r) \in \{\mathbf{u}, \mathbf{v}\}^\ell$. In the next $s - 1$ steps, the letters in these factors shift to the next highest factor and the respective letters of $\mathbf{u}$ and $\mathbf{v}$ flow in. Thus, this last part will contain $\bar{\tau}(r)$ except for its first $s - 1$ letters, as $a_{2n+s+1}, \dots, a_{3n+1}$ after s steps. Observe that the letters of $\tau(r)$ appear as n successive states a_{3n+1} of $\mathcal{A}_{3n+1}$, which is the last factor.

If the signal did not arrive, the above transition rules imply that $\mathbf{u}^\ell$ will be in the buffer after n input letters, and will then flow through and out of the next part.

We will use the fact that, except for the last $3n+1^{\text{st}}$ factor, the feedback function φ_t of the t^{th} does not depend on its last state factor a_{3n+1}.

As to the mapping onto $\mathcal{R}_{\tau,\{r\},d}$, take a triplet $(k, p, yq) \in \mathcal{R}_{\tau,\{r\},1}$ ($y \in \{u_t, v_t \mid t = 1, \dots, s\}$). We represent this triplet (k, p, yq) by an appropriate state $b = c(\mathbf{u}, k)c(\mathbf{v}\mathbf{u}^{\ell-1}, k)\overline{c}(\mathbf{z}\mathbf{u}^{\ell-1}, k-1)e_1 \dots e_{n-s+1}$ of $\mathcal{N}$. The number k is represented by the value $c(\mathbf{v}\mathbf{u}^{\ell-1}, k)$ and $\overline{c}(\mathbf{z}\mathbf{u}^{\ell-1}, k-1)$, $\mathbf{z} = z_1 \dots z_s$ represents p with $z_i \in \{u_i, v_i\}$, $i = 1, \dots, s$. Namely, if $z_1 = u_1$, then $p = \lambda$ is assumed, and, if $z_1 = v_1$, then p is understood as the k-length prefix of r. In other words, $z_1 = v_1$ means $r = pp'$ for some $p' \in X^*$ (with $|p| = k$). And $z_1 = u_1$ means $p = \lambda$. Setting $y_1 \dots y_n \in \{\mathbf{u}, \mathbf{v}\}^\ell$, assume

[4] Lemma 3.3 guarantees that, for each factor t of the buffer, $(a_{t-1}, a_{1-t (\mathrm{mod}\, s)}) = (v_1, u_1)$ only when $k = t - s - 1 (\mathrm{mod}\, n)$. Especially, for the first factor of the buffer, i.e., for $t = n + s + 1$, $(a_{s+n}, a_s) = (v_1, u_1)$ if and only if $k = n$.

$$e_1 \dots e_{n-s+1} = \begin{cases} y_{s+k} \dots y_{s+1} u_s \dots u_{k+1} y_k & \text{if } 1 \leq k < s,\ \ell = 2, \\ y_{n-s+k} \dots y_{n-s+1} u_s \dots u_{k+1} y_{n-2s+k} \dots y_{n-2s+1} & \\ u_s \dots u_{k+1} \dots y_{s+k} \dots y_{s+1} u_s \dots u_{k+1} y_k & \text{if } 1 \leq k < s, \ell > 2, \\ y_n \dots y_k & \text{if } k = s, \\ u_j \dots u_1 (u_s \dots u_1)^{i-1} y_n \dots y_k & \text{if } k = is + j, k < n, \\ & \quad 1 \leq j < s, \\ (u_s \dots u_1)^{i-1} y_n \dots y_k & \text{if } k = is \leq n. \end{cases}$$

Then k and the mirror image of $e_1 \dots e_{n-s+1}$ represents $y_k \dots y_n$. If $k \geq s$, then this is obvious considering the structure of $e_1 \dots e_{n-s+1}$. (Recall that $u_s = v_s$.) If $1 \leq k < s$ and $\ell > 2$, then the mirror image of $e_1 \dots e_{n-s+1}$ is $y_k u_{k+1} \dots u_s y_{s+1} \dots y_{s+k} \dots u_{k+1} \dots u_s\ y_{n-2s+1} \dots y_{n-2s+k}\ u_{k+1} \dots u_s y_{n-s+1} \dots y_{n-s+k}$ representing $y_k \dots y_n$. (Observe that k and $y_k \in \{u_k, v_k\}$ unambiguously determines $y_k \dots y_s$, moreover, for any $i = 1, \dots, \ell$, $y_{n-is+1} \in \{u_1, v_1\}$ unambiguously determines $y_{n-is+2} \dots y_{n-(i-1)s}$.) We have similar consequences for $1 \leq k < s$ and $\ell = 2$. The motivation for this representation should be clear from the explanation of the buffer cycle discussed above.

Now we give the formal definition of $\mathcal{B}'$, and that of a mapping $\psi : B' \to R_{\tau,\{r\},1}$ under which $\mathcal{R}_{\tau,\{r\},1}$ is an (y-represented) homomorphic image of $\mathcal{B}'$.

Let B' consists of all $b \in A^{3n+1}$ for which there are $(k, p, q) \in R_{\tau,\{r\},1}$ such that $b = c(\mathbf{u}, k)c(\mathbf{v}\mathbf{u}^{\ell-1}, k)\overline{c}(\mathbf{z}\mathbf{u}^{\ell-1}, k-1)e_1 \dots e_{n-s+1}$, $\mathbf{z} = z_1 \dots z_s$, with $z_i \in \{u_i, v_i\}, i = 1, \dots, s$, where

$$z_1 = \begin{cases} v_1 & \text{if } p \text{ is the } k\text{-length prefix of } r, \\ u_1 & \text{otherwise,} \end{cases}$$

$e_1 \dots e_{n-s+1}$ is defined as above and q is represented by k and $e_1 \dots e_{n-s+1}$ as we explained. (Recall that by the structure of $R_{\tau,\{r\},1}$, $y_{n-s+k} \dots y_{n-s+1} u_s \dots u_{k+1}\ y_{n-2s+k} \dots y_{n-2s+1} u_s \dots u_{k+1} \dots y_{s+k} \dots y_{s+1} u_s \dots u_{k+1} y_k$ unambiguously determines $q = y_k \dots y_n$, whenever $k < s$ and $\ell > 2$. Similarly, by the structure of $R_{\tau,\{r\},1}$, $y_{s+k} \dots y_{s+1}\ u_s \dots u_{k+1} y_k$ unambiguously determines $q = y_k \dots y_n$, whenever $k < s$ and $\ell = 2$. Moreover, $q = e_{n-s+1} \dots e_{k-s+1}$ is assumed if $k \geq s$.) Furthermore, let $\psi(b) = (k, p, q)$. It is routine work to show that $\mathcal{N}$ has a subautomaton $\mathcal{B}'$ with state set B' which can be mapped homomorphically by ψ onto $\mathcal{R}_{\tau,\{r\},1}$. Finally, by $\psi(b) = (k, p, q)$, the last letter of b is the same as the first letter of q. Therefore, $\mathcal{N}$ y-represents $\mathcal{R}_{\tau,\{r\},1}$.

Applying Proposition 2.2 to the product $\mathcal{N}$, it is clear that we will get a product $\mathcal{N}'$, which also y-represents $\mathcal{R}_{\tau,\{r\},1}$, moreover, similarly to $\mathcal{N}$, apart from the last factor, the feedback functions of the factors of $\mathcal{N}'$ really do not depend on their last state variable. Thus it is enough to observe that by an inductive application of Proposition 2.2, we can derive from the product $\mathcal{N}$ a primitive product $\mathcal{M}$.

Especially, every vertex of the underlying graph of $\mathcal{N}$ has not more than two incoming and two outgoing edges in the resulting product. Moreover, if there is a vertex with two outgoing edges, then it is an element of a cycle with one edge going into another element of the same cycle, and all of the other cycle elements have one outgoing edge connecting them with other elements of the cycle.

In addition, cycle elements have only one incoming edge, coming from another element of the cycle. [5] Using Proposition 2.1, we may assume that $\mathcal{N}$ is a primitive product, for otherwise we could relabel its components by an appropriate permutation of their indices.

This ends the proof of Lemma 4.1. $\square$

We next prove

Lemma 4.2. *Let $\mathcal{A}$ be an automaton satisfying Letichevsky's criterion, $\mathcal{A}^k(X, \varphi'_1, \ldots, \varphi'_k)$, $\mathcal{A}^\ell(X, \varphi''_1, \ldots, \varphi''_\ell)$ be primitive powers of $\mathcal{A}$ such that, apart from the last factors, the feedback functions of the factors really do not depend on their last state variable. Suppose that they y-represent, in order, $\mathcal{R}_{\tau,H_1,d}$ and $\mathcal{R}_{\tau,H_2,d}$ for some $\tau : X^n \to A^n$, $H_1, H_2 \subseteq \{p \in X^+ \mid |p| = n\}$ (H_1, H_2 are not necessarily disjoint sets), and $d \geq 1$, where n is a multiple of s as before. There exists a primitive power $\mathcal{M} = \mathcal{A}^{k+\ell+1}(X, \varphi_1, \ldots, \varphi_{k+\ell+1})$, which y-represents $\mathcal{R}_{\tau,H_1 \cup H_2,d+1}$. Moreover, apart from the last factor, the feedback functions of the factors of $\mathcal{M}$ really do not depend on their last state variable.*

Proof: Define the power $\mathcal{A}^{k+\ell+1}(X, \varphi_1, \ldots, \varphi_{k+\ell+1})$ in the following way.

For any $(a_1, \ldots, a_{k+\ell+1}) \in A^{k+\ell+1}, x \in X, t = 1, \ldots, n+\ell+1$,

$$\varphi_t(a_1, \ldots, a_{k+\ell+1}, x) = \begin{cases} \varphi'_t(a_1, \ldots, a_k, x) & \text{if } t \leq k, \\ \varphi''_{t-k}(a_{k+1}, \ldots, a_{k+\ell}, x) & \text{if } k < t \leq k+\ell, \\ x[a_{k+\ell+1}, a_k \vee a_{k+\ell}] & \text{otherwise.} \end{cases}$$

Clearly, this power of $\mathcal{A}$ is primitive.

Now we consider, in order, appropriate homomorphisms ψ' and ψ'' such that $\mathcal{A}^k(X, \varphi'_1, \ldots, \varphi'_k)$, y-represents $\mathcal{R}_{\tau,H_1,d}$ with respect to ψ', and moreover, $\mathcal{A}^\ell(X, \varphi''_1, \ldots, \varphi''_\ell)$, y-represents $\mathcal{R}_{\tau,H_2,d}$ with respect to ψ''. It is clear that φ_t does not depend on its last state variable if $t \neq k+\ell+1$. Therefore, it is a routin work to show that the power $\mathcal{M}$ y-represents $\mathcal{R}_{\tau,H_1 \cup H_2,d+1}$ with respect to the homomorphism ψ having the following properties.

$$\psi(a_1, \ldots, a_{k+\ell+1}) = (c, p, a_{k+\ell+1} y_1 \ldots y_{d-c+n}),$$

whenever $\psi'(a_1, \ldots, a_k) = (c, p', y'_1 \ldots y'_{d-c+n})$, $\psi''(a_{k+1}, \ldots, a_{k+\ell}) = (c, p'', y''_1 \ldots y''_{d-c+n})$, $\{p', p''\} = \{p, \lambda\}$ (with $|p| \in \{0, c\}$ including the possibility of $p = \lambda$),

[5] The cycles may be "wired" in such a way that their first element is connected to the last one and all the others are connected to the previous ones. Then the cycles can represent "clocks" so that, for instance, if $d_1 \ldots d_{ms}$ is a state of a cycle (with ms length) representing the k^{th} state of an arbitrary "clock" then $d_2 \ldots d_{ms} d_1$ will represent its $k+1(\text{mod } ms)^{th}$ state.

$$y_{d-c+n}\ (= y'_{d-c+n} = y''_{d-c+n}) = u_s\ (= v_s),$$

$$y_{j-1} = \begin{cases} u_s\ (= v_s) & \text{if } y'_{j-1} = y''_{j-1} = u_s\ (= v_s) \text{ and } y_j \in \{u_1, v_1\}, \\ u_i & \text{if } y'_{j-1} = y''_{j-1} = u_i \text{ and } y_j = u_{i+1},\ i \in \{1, \ldots, s-1\}, \\ v_i & \text{if } v_i \in \{y'_{j-1}, y''_{j-1}\} \text{ and } y_j = v_{i+1},\ i \in \{1, \ldots, s-1\}, \end{cases}$$

$j = 2, \ldots, d - c + n$, provided

$$a_{k+\ell+1} = \begin{cases} u_s\ (= v_s) & \text{if } c - d + 1 (\text{mod } s) = 1, \\ u_{c-d(\text{ mod } s)} & \text{if } y_1 = u_{c-d+1(\text{ mod } s)}, \\ & \ c - d + 1 (\text{mod } s) \in \{2, \ldots, s-1\}, \\ v_{c-d(\text{ mod } s)} & \text{if } y_1 = v_{c-d+1(\text{ mod } s)}, \\ & \ c - d + 1 (\text{mod } s) \in \{2, \ldots, s-1\}, \\ \text{arbitrary element of } \{\mathrm{u_{s-1}}, \mathrm{v_{s-1}}\} & \text{if } c - d + 1 (\text{mod } s) = s. \end{cases}$$

Using the definition of $\mathcal{R}_{\tau,H,d}$, by Lemma 3.5 it is obvious that $y_1 \ldots y_{d-c+n}$ and ψ are well defined.

□

We shall use the following concept as well. Define the subautomaton $\mathcal{R}_{\tau,d}$ of $\mathcal{R}_{\tau,X^n,d}$ to have state set $R_{\tau,d} = R_{\tau,X^n,d} \setminus \{(k, \lambda, q) \mid (k, \lambda, q) \in R_{\tau,X^n,d}\}$. This is a subautomaton since px is a prefix of a word of X^n for every $p \in X^*$ with $|p| < n, x \in X$. For $(k, p, q) \in R_{\tau,d}$, we have $|p| = k$ always, so we will use the short notation (p, q) for $(k, p, q) \in R_{\tau,d}$. We have, for $(p, yq) \in R_{\tau,d}$ ($y \in \{u_t, v_t \mid t = 1, \ldots, s\}$) and $x \in X$,
$\delta_{\tau,d}((p, yq), x) = \begin{cases} (px, q) & \text{if } |p| < n, \\ (x, q\tau(p)) & \text{if } |p| = n. \end{cases}$

Let $\mathcal{M} = (W \times Z, X_{\mathcal{M}}, \delta_{\mathcal{M}})$ be an automaton with $Y \subseteq Z$, moreover, let $\mathcal{B} = (B, X, \delta_{\mathcal{B}})$ be a subautomaton of $\mathcal{M}$ having a homomorphism $\psi : B \to R_{\tau,d}$ onto $\mathcal{R}_{\tau,d}$ such that $\psi((w, z)) = (p, yq)$ ($(w, z) \in B, (p, yq) \in R_{\tau,d},\ y \in Y$) implies $z = y$. Then we also say that $\mathcal{M}$ *y-represents* $\mathcal{R}_{\tau,d}$ (with respect to ψ).

The following statement is obvious.

Proposition 4.3. *Let $\tau_i : X^n \to Y^n, i = 1, \ldots, m$ be a system of mappings, moreover, let d be a positive integer. For any $i = 1, \ldots, m$, let $\mathcal{M}_i = (W_i \times Z_i, X, \delta_i)$ be an automaton which y-represents $\mathcal{R}_{\tau_i,d}$. Consider an automaton $\mathcal{M}_{m+1} = (M_{m+1}, X_{m+1}, \delta_{m+1})$ with $X_{m+1} = Y^m$, a product $\mathcal{U} = \mathcal{R}_{\tau_1,d} \times \ldots \times \mathcal{R}_{\tau_m,d} \times \mathcal{M}_{m+1}(X, \varphi_1, \ldots, \varphi_{m+1})$ with*

$$\varphi_t((p_1, y_1q_1), \ldots, (p_m, y_mq_m), a, x) = \begin{cases} x & \text{if } t \leq m, \\ (y_1, \ldots, y_m) & \text{if } t = m + 1 \end{cases}$$

$((p_i, y_i q_i) \in R_{\tau_i,d}, y_i \in Y, i = 1, \ldots, m, a \in M_{m+1}, x \in X)$.
Define the product $\mathcal{M} = \mathcal{M}_1 \times \ldots \times \mathcal{M}_m \times \mathcal{M}_{m+1}(X, \psi_1, \ldots, \psi_{m+1})$ *with*

$$\psi_t((w_1, z_1), \ldots, (w_m, z_m), a, x) = \begin{cases} x & \text{if } t \leq m, \\ (z_1, \ldots, z_m) & \text{if } t = m+1. \end{cases}$$

Then $\mathcal{M}$ *homomorphically represents* $\mathcal{U}$. □

Lemma 4.4. *Let* $\mathcal{A} = (A, X, \delta_{\mathcal{A}})$ *be an automaton satisfying Letichevsky's criterion and let* $\mathbf{u} = u_1 \ldots u_s, \mathbf{v} = v_1 \ldots v_s$ *be any pair of its control words. Consider an alphabet* X*, a multiple* n *of* s *with* $n = \ell s, \ell > 1$ *and a mapping* $\tau : X^n \to A^n$ *having the property* $\tau(p) \in \{\mathbf{u}, \mathbf{v}\}^\ell$ *for each* $p \in X^n$*. For any integer* $d \geq |X^n|$*, there exists a primitive power* $\mathcal{M}$ *of* $\mathcal{A}$ *such that* $\mathcal{R}_{\tau,d}$ *is y-represented by* $\mathcal{M}$*. Moreover, apart from the last factor, the feedback functions of the factors of* $\mathcal{M}$ *really do not depend on their last state variable.*

Proof: By Lemma 4.1 and by an inductive application of Lemma 4.2, we can prove that $\mathcal{R}_{\tau,X^n,|X^n|}$ is y-represented by an appropriate primitive power $\mathcal{M}' = \mathcal{A}^k(X, \varphi'_1, \ldots, \varphi'_k)$ of $\mathcal{A}$. Since $\mathcal{R}_{\tau,|X^n|}$ is a subautomaton of $\mathcal{R}_{\tau,X^n,|X^n|}$, this primitive power $\mathcal{M}'$ also y-represents $\mathcal{R}_{\tau,|X^n|}$. If $d = |X^n|$, then $\mathcal{M}'$ has the required conditions. Otherwise, let $\mathcal{M} = \mathcal{A}^{k+\ell}(X, \varphi_1, \ldots, \varphi_{k+\ell})$ with $\ell = d - |X^n|$ such that for any $(a_1, \ldots, a_{k+\ell}) \in A^{k+\ell}, x \in X, t = 1, \ldots, k+\ell$,

$$\varphi_t(a_1, \ldots, a_{k+\ell}, x) = \begin{cases} \varphi'_t(a_1, \ldots, a_k, x) & \text{if } t \leq k, \\ x[a_t, a_{t-1}] & \text{otherwise.} \end{cases}$$

This power of $\mathcal{A}$ is primitive and y-represents $\mathcal{R}_{\tau,d}$. □

Lemma 4.5. *Let* $\mathcal{D} = (D, X, \delta)$ *and* $\mathcal{B} = (B, Y, \delta')$ *be automata with* $D \subseteq B$*. Moreover, let* $\tau : X^n \to Y^n$ $(n > 0)$ *be a mapping and assume that for a suitable integer* $d > 0$ *the following two conditions are satisfied:*
(i) *For all* $a \in B$, $(p, q) \in R_{\tau,d}$, $p \in X^n$*:*
$\delta'(a, q) \in D$ *implies* $\delta(\delta'(a, q), p) = \delta'(a, q\tau(p))$ $(\in D)$.
(ii) $\{\delta(\delta'(a, q), p) \mid a \in D, (p, q) \in R_{\tau,d}, \delta'(a, q) \in D\} = D$.
Then there exists an α_0*-product* $\mathcal{R}_{\tau,d} \times \mathcal{B}(X, \varphi_1, \varphi_2)$ *which homomorphically represents* $\mathcal{D}$ *such that* $\varphi_2((p, yq), x)$ $((p, yq) \in R_{\tau,d}, x \in X, y \in Y)$ *really depends only on* y.

Proof: Form the α_0-product $\mathcal{C} = (C, X, \delta'') = \mathcal{R}_{\tau,d} \times \mathcal{B}(X, \varphi_1, \varphi_2)$, where for arbitrary $(p, yq) \in R_{\tau,d}$ $(y \in Y), b \in B$ and $x \in X$, $\varphi_1(x) = x$ and $\varphi_2((p, yq), x) = y$. Define the subautomaton $\mathcal{C}'$ of $\mathcal{C}$ with states $C' = \{((p, yq), a) \in C \mid \delta'(a, yq) \in D\}$ and input set X. We map the state $c = ((p, yq), a)$ of $\mathcal{C}'$ to the state $\delta(\delta'(a, yq), p)$ of $\mathcal{D}$.

Assume that $\mathcal{C}'$ receives an input letter x in this state c. If $|p| < n$, then $\delta''(c, x) = ((px, q), \delta'(a, y))$, which maps to the state $\delta(\delta'(\delta'(a, y), q), px) = \delta(\delta(\delta'(a, yq), p), x)$

of $\mathcal{D}$, as required.
If on the other hand, $|p| = n$, then $\delta''(c,x) = ((x, q\tau(p)), \delta'(a,y))$. This maps to $\delta(\delta'(\delta'(a,y), q\tau(p)), x)$ in $\mathcal{D}$, that is, to $\delta(\delta'(\delta'(a,yq), \tau(p)), x) = \delta(\delta(\delta'(a,yq),p),x)$, by (i) since $\delta'(a,yq) \in D$. (Observe in the second case, $|q| = d-1$.)

Thus, the mapping $\psi(((p,q),a)) = \delta(\delta'(a,q),p)$ ($((p,q),a) \in C', a \in D \subseteq B$) is a homomorphism of a subautomaton of $\mathcal{C}$ into $\mathcal{D}$. By (ii), ψ is a mapping onto D. Finally, as $\varphi_2((p,yq),x) = y$ ($(p,yq) \in R_{\tau,d}, x \in X$), we obtain that φ_2 really depends only on y. This ends the proof. □

We shall use the following natural extension of this result.

Lemma 4.6. *Let $\mathcal{D} = (D, X, \delta)$ be an automaton. Consider a product $\mathcal{N} = (B_1 \times \ldots \times B_m, Z^\ell, \delta') = \mathcal{B}_1 \times \ldots \times \mathcal{B}_m(Z^\ell, \varphi_1, \ldots, \varphi_m)$ of automata $\mathcal{B}_t$, $1 \le t \le m$ with $D \subseteq B_1 \times \ldots \times B_m$. Let $\tau_i : X^n \to Z^n$ $(n > 0)$, $1 \le i \le \ell$, be mappings, moreover let $\tau : X^n \to (Z^\ell)^n$ with $\tau_i(p) = z_{1,i} \ldots z_{n,i}, i = 1, \ldots, \ell$ whenever $\tau(p) = (z_{1,1}, \ldots, z_{1,\ell}) \ldots (z_{n,1}, \ldots, z_{n,\ell})$ such that the following two conditions hold:*

(i) For every $a \in B_1 \times \ldots \times B_m$, $(p,q) \in R_{\tau,d}$, $p \in X^n$:
$\delta'(a,q) \in D$ implies
$\delta(\delta'(a,q),p) = \delta'(a, q\tau(p))$ $(\in D)$.
(ii) $D = \{\delta(\delta'(a,q),p) \mid a \in B_1 \times \ldots \times B_m, (p,q) \in R_{\tau,d}, \delta'(a,q) \in D\}$.

Then the product $\mathcal{V} = \mathcal{R}_{\tau_1,d} \times \ldots \times \mathcal{R}_{\tau_\ell,d} \times \mathcal{N}(X, \varphi'_1, \ldots, \varphi'_\ell, \varphi''_{\ell+1}) = \mathcal{R}_{\tau_1,d} \times \ldots \times \mathcal{R}_{\tau_\ell,d} \times \mathcal{B}_1 \times \ldots \times \mathcal{B}_m(X, \varphi'_1, \ldots, \varphi'_{\ell+m})$ homomorphically represents $\mathcal{D}$, where for each $(1 \le i \le \ell + m)$, we have

$$\varphi'_i((p_1, y_1q_1), \ldots, (p_\ell, y_\ell q_\ell), b_1, \ldots, b_m, x)$$

$$= \begin{cases} x & \text{if } 1 \le i \le \ell, \\ \varphi_{i-\ell}(b_1, \ldots, b_m, (y_1, \ldots, y_\ell)) & \text{otherwise} \end{cases}$$

$((p_i, y_iq_i) \in R_{\tau_i,d}$, $1 \le i \le \ell, x \in X, (y_1, \ldots, y_\ell) \in Z^\ell$, and $\varphi''_{\ell+1} = (\varphi'_{\ell+1}, \ldots, \varphi'_{\ell+m})$.

Proof: First we apply Lemma 4.5 taking $\mathcal{N}$ in the role of $\mathcal{B}$. Consider the α_0-product $\mathcal{U} = \mathcal{R}_{\tau,d} \times \mathcal{N}(X, \chi_1, \chi_2)$ given by Lemma 4.5, and the product $\mathcal{V} = \mathcal{R}_{\tau_1,d} \times \ldots \times \mathcal{R}_{\tau_\ell,d} \times \mathcal{B}_1 \times \ldots \times \mathcal{B}_m(X, \varphi'_1, \ldots, \varphi'_{\ell+m})$ just defined.

For a state $(p, yq, b_1, \ldots, b_m)$ of $\mathcal{U}$, where $(p, yq) \in R_{\tau,d}$ with $p \in X^*$, $y = y_1 \ldots y_\ell \in Z^\ell$, $q \in (Z^\ell)^*$, $b_i \in B_i$ $(1 \le i \le m)$ given $q = (x_{1,1}, \ldots, x_{1,\ell}) \ldots (x_{h,1}, \ldots, x_{h,\ell})$, we put $q_j = \begin{pmatrix} x_{1,j} \\ \vdots \\ x_{h,j} \end{pmatrix} \in Z^h$, (for some $h \ge 0$, for each $j = 1, \ldots, \ell$). We write this state as

$$(p, \begin{pmatrix} y_1 & \ldots & y_\ell \\ x_{1,1} & \ldots & x_{1,\ell} \\ \vdots & \ddots & \vdots \\ x_{h,1} & \ldots & x_{h,\ell} \end{pmatrix}, b_1, \ldots, b_m).$$

Define an injective mapping $\varrho : R_{\tau,d} \times B_1 \times \ldots \times B_m \to R_{\tau_1,d} \times \ldots \times R_{\tau_\ell,d} \times B_1 \times B_m$ by

$$\varrho(p, \begin{pmatrix} y_1 & \ldots & y_\ell \\ x_{1,1} & \ldots & x_{1,\ell} \\ \vdots & \ddots & \vdots \\ x_{h,1} & \ldots & x_{h,\ell} \end{pmatrix}, b_1, \ldots, b_m) = (\begin{pmatrix} (p, y_1 x_{1,1} \ldots x_{h,1}) \\ \vdots \\ (p, y_\ell x_{1,\ell} \ldots x_{h,\ell}) \end{pmatrix}, b_1, \ldots, b_m).$$

Denote by $\delta_{\mathcal{U}}$ resp. $\delta_{\mathcal{V}}$ the transition functions of $\mathcal{U}$ resp. $\mathcal{V}$.

If $|p| < n$, in $\delta_{\mathcal{U}}(((p, yq), b_1, \ldots, b_m), x)$, the only changes are that the row of y's is lost, p is replaced by px, and b_i is replaced by b'_i, which is b_i acted on by $\varphi_{i-\ell}(b_1, \ldots, b_m, (y_1, \ldots, y_\ell))$ in $\mathcal{B}_i$ for $i = 1, \ldots, m$, whereas in ϱ of this state, the 'column' of y's is lost and p is replaced by px, while the b's change in the same way.
If $|p| = n$, then $\varrho(\delta_{\mathcal{U}}(((p, yq), b_1, \ldots, b_m), x)$ is

$$\varrho(x, \begin{pmatrix} x_{1,1} & \ldots & x_{1,\ell} \\ \vdots & \ddots & \vdots \\ x_{h,1} & \ldots & x_{h,\ell} \\ z_{1,1} & \ldots & z_{1,\ell} \\ \vdots & \ddots & \vdots \\ z_{n,1} & \ldots & z_{n,\ell} \end{pmatrix}, b'_1, \ldots, b'_m) = (\begin{pmatrix} (x, x_{1,1} \ldots x_{h,1} z_{1,1} \ldots z_{n,1}) \\ \vdots \\ (x, x_{1,\ell} \ldots x_{h,\ell} z_{1,\ell} \ldots z_{n,\ell}) \end{pmatrix}, b'_1, \ldots, b'_m),$$

where $\tau(p) = \begin{pmatrix} z_{1,1} & \ldots & z_{1,\ell} \\ \vdots & \ddots & \vdots \\ z_{n,1} & \ldots & z_{n,\ell} \end{pmatrix}$, $\tau_i(p) = \begin{pmatrix} z_{1,i} \\ \vdots \\ z_{n,i} \end{pmatrix}$, $i = 1, \ldots, \ell$.

This shows that for any $(p, yq) \in R_{\tau,d}$, $(b_1, \ldots, b_m) \in B_1 \times \ldots \times B_m$, $x \in X$, we have

$$\varrho(\delta_{\mathcal{U}}(((p, q), b_1, \ldots, b_m), x)) = \delta_{\mathcal{V}}(\varrho(((p, q), b_1, \ldots, b_m)), x).$$

Therefore, the product $\mathcal{U}$ can be embedded isomorphically into the product $\mathcal{V}$. But by Lemma 4.5, $\mathcal{U}$ homomorphically represents $\mathcal{D}$. Thus, $\mathcal{V}$ also has this property. $\square$

5. Main Result

In this section, we will establish that a primitive product of Letichevsky automata can homomorphically represent any finite automaton $\mathcal{E}$. To avoid trivialities, we note that it is enough to restrict to cases in which $\mathcal{E}$ has at least three states.
Consider an automaton $\mathcal{A}$ satisfying Letichevsky's criterion and let $\mathbf{u} = u_1 \ldots u_s, \mathbf{v} = v_1 \ldots v_s$ denote a pair of control words for $\mathcal{A}$ as before. We put $\mathbf{u} \wedge \mathbf{u} = \mathbf{u} \vee \mathbf{u} = \mathbf{u}$, $\mathbf{v} \wedge \mathbf{v} = \mathbf{v} \vee \mathbf{v} = \mathbf{v}$, $\mathbf{v} \wedge \mathbf{u} = \mathbf{u} \wedge \mathbf{v} = \mathbf{u}$ and $\mathbf{v} \vee \mathbf{u} = \mathbf{u} \vee \mathbf{v} = \mathbf{v}$, so that $\wedge$ and $\vee$ are logical AND resp. logical OR on the set $\{\mathbf{u}, \mathbf{v}\}$.

First we show the following technical result.

Lemma 5.1. *Define the automata* $\mathcal{B} = (\{\mathbf{u},\mathbf{v}\}^9, \{\mathbf{u},\mathbf{v}\}^4, \delta_\mathcal{B})$, $\mathcal{C} = (\{\mathbf{u},\mathbf{v}\}^{18n}, \{\mathbf{u},\mathbf{v}\}^{6n}, \delta_\mathcal{C})$, $n \geq 3$, *with*

$$\delta_\mathcal{B}((\mathbf{a}_1,\ldots,\mathbf{a}_9),(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\mathbf{x}_4)) = (\mathbf{a}_2, \mathbf{a}_3\wedge\mathbf{x}_1, \mathbf{a}_4, \mathbf{a}_5\wedge\mathbf{x}_2, \mathbf{a}_6\vee\mathbf{a}_3, \mathbf{a}_7\wedge\mathbf{x}_3, \mathbf{a}_8, \mathbf{a}_9, \mathbf{x}_4),$$

$$\delta_\mathcal{C}((\mathbf{a}_1,\ldots,\mathbf{a}_{18n}),(\mathbf{x}_1,\ldots,\mathbf{x}_{6n})) = (\mathbf{a}'_1,\ldots,\mathbf{a}'_{18n}), \text{ where for } i=1,\ldots,n,$$

$$(\mathbf{a}'_{18(i-1)+1},\ldots,\mathbf{a}'_{18(i-1)+9}) = \delta_\mathcal{B}((\mathbf{a}_{18(i-1)+1},\ldots,\mathbf{a}_{18(i-1)+9}), (\mathbf{x}_{6(i-1)+1},\mathbf{x}_{6(i-1)+2},\mathbf{x}_{6(i-1)+3},\mathbf{a}_{18(i-1)+10})),$$

$$(\mathbf{a}'_{18(i-1)+10},\ldots,\mathbf{a}'_{18(i-1)+18}) = \delta_\mathcal{B}((\mathbf{a}_{18(i-1)+10},\ldots,\mathbf{a}_{18(i-1)+18}), (\mathbf{x}_{6(i-1)+4},\mathbf{x}_{6(i-1)+5},\mathbf{x}_{6(i-1)+6},\mathbf{a}_{18(i-1)+1}\vee\mathbf{a}_{18(i-1)+19\,(\mathrm{mod}\,18n)})).$$

There exist a positive integer m *and input words* $\hat\gamma_0,\hat\gamma_1,\hat\gamma_2,\hat\gamma_3$ *of* $\mathcal{C}$ *having the following properties. Given an appropriate subset* $\{b_1,\ldots,b_n\}$, *of the state set of* $\mathcal{B}$, *for every transformation* γ *of* $\{b_1,\ldots,b_n\}$, *there exists a word* $\hat\gamma \in \{\hat\gamma_0,\hat\gamma_1,\hat\gamma_2,\hat\gamma_3\}^+$ *inducing* γ *(i.e.* $\delta_\mathcal{B}(b_i,\hat\gamma) = \gamma(b_i), i=1,\ldots,n)$ *such that* $|\hat\gamma| = m$.

Proof: Consider states of $\mathcal{C}$ having the form $(\mathbf{u}^4\mathbf{a}_1\mathbf{u}^{13}\ldots\mathbf{u}^4\mathbf{a}_n\mathbf{u}^{13})$, $\mathbf{a}_1,\ldots,\mathbf{a}_n \in \{\mathbf{u},\mathbf{v}\}$ and use the short notation $(\mathbf{d},\mathbf{e}) = \mathbf{u}^4\mathbf{d}\mathbf{u}^8\mathbf{e}\mathbf{u}^4, \mathbf{d},\mathbf{e} \in \{\mathbf{u},\mathbf{v}\}$. We represent b_i, $i \in \{1,\ldots,n\}$ by the state $\mathbf{u}^{18(i-1)}\mathbf{u}^4\mathbf{v}\mathbf{u}^{13}\mathbf{u}^{18(n-i)}$ of $\mathcal{C}$, which, using the short notation, is $(\mathbf{u},\mathbf{u})^{i-1}(\mathbf{v},\mathbf{u})(\mathbf{u},\mathbf{u})^{n-i}$. First we show that we have words $q_{0,0}, q_{i,j} \in \{\mathbf{u},\mathbf{v}\}^+, i=1,2,3,4,\ j=1,\ldots,n$ all having the same length such that

$$\delta_\mathcal{C}((\mathbf{d}_1,\mathbf{e}_1,\ldots,\mathbf{d}_n,\mathbf{e}_n),q_{i,j}) = (\mathbf{d}'_1,\mathbf{e}'_1,\ldots,\mathbf{d}'_n,\mathbf{e}'_n),$$

where

$(\mathbf{d}'_\ell,\mathbf{e}'_\ell) = (\mathbf{d}_\ell,\mathbf{e}_\ell)$ if $\ell \neq j$, and otherwise

$$(\mathbf{d}'_j,\mathbf{e}'_j) = \begin{cases} (\mathbf{d}_j,\mathbf{e}_j) & \text{if } (i,j)=(0,0),\\ (\mathbf{d}_j,\mathbf{d}_{j+1(\mathrm{mod}\,n)}), & \text{if } i=1, j=1,\ldots,n,\\ (\mathbf{e}_j,\mathbf{d}_j), & \text{if } i=2, j=1,\ldots,n,\\ (\mathbf{u},\mathbf{e}_j) & \text{if } i=3, j=1,\ldots,n,\\ (\mathbf{d}_j\vee\mathbf{e}_j,\mathbf{e}_j) & \text{if } i=4, j=1,\ldots,n. \end{cases}$$

Using the symmetry of the structure of $\mathcal{C}$, to show the existence of the q's, it is enough to prove the existence of $q_{0,0}, q_{i,1} \in \{\mathbf{u},\mathbf{v}\}^+, i=1,2,3,4$. Define the following input *letters* (not words!) of $\mathcal{C}$.
$x_0 = (\mathbf{uvu})^{2n}, x_1 = \mathbf{uvuuuuvvu}(\mathbf{uvu})^{2n-3}, x_2 = \mathbf{uvuuuv}(\mathbf{uvu})^{2n-2}$,
$x_3 = \mathbf{vvuvvu}(\mathbf{uvu})^{2n-2}, x_4 = \mathbf{uuvuuv}(\mathbf{uvu})^{2n-2}, x_5 = \mathbf{uuu}(\mathbf{uvu})^{2n-1}$,
$x_6 = \mathbf{uvuvvu}(\mathbf{uvu})^{2n-2}, x_7 = \mathbf{uvv}(\mathbf{uvu})^{2n-1}$.

Let us consider the following computations.

(0) $\delta_\mathcal{C}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_0}^9)$

$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_0}^8)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_0}^7)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_0}^6)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_0}^5)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_0}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_0}^3)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_0}^2)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, x_0)$
$= (\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4).$

(1) $\delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_1}^5{x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^{17}\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_1}^4{x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^{17}\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_1}^3{x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^{14}\mathbf{d}_2\mathbf{u}^2\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_1}^2{x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^{14}\mathbf{d}_2\mathbf{u}^2\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, x_1{x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^{14}\mathbf{d}_2\mathbf{u}^2\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^5\mathbf{d}_2\mathbf{u}^2\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_2}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^{11}\mathbf{d}_2\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^2\mathbf{d}_2\mathbf{u}^5\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_2}^3)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^{11}\mathbf{d}_2\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^2\mathbf{d}_2\mathbf{u}^5\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_2}^2)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^{11}\mathbf{d}_2\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, x_2)$
$= (\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4).$

(2) $\delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_3}^5{x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_3}^4{x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_3}^3{x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}\mathbf{d}_1\mathbf{u}^2\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_3}^2{x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{d}_1\mathbf{u}^2\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, x_3{x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^5\mathbf{d}_1\mathbf{u}^2\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5\mathbf{d}_1, {x_4}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{d}_1\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^2\mathbf{d}_1\mathbf{u}, {x_4}^3)$
$= \delta_{\mathcal{C}}(\mathbf{u}^6\mathbf{e}_1\mathbf{u}^8\mathbf{d}_1\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^2\mathbf{d}_1\mathbf{u}^2, {x_4}^2)$
$= \delta_{\mathcal{C}}(\mathbf{u}^5\mathbf{e}_1\mathbf{u}^8\mathbf{d}_1\mathbf{u}^5\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, x_4)$
$= (\mathbf{u}^4\mathbf{e}_1\mathbf{u}^8\mathbf{d}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\dots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4).$

(3) $\delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_5}^9)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{12}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_5}^8)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{11}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_5}^7)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{13}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_5}^6)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{12}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_5}^5)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{11}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_5}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{13}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_5}^3)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{12}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_5}^2)$
$= \delta_{\mathcal{C}}(\mathbf{u}^{11}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, x_5)$
$= (\mathbf{u}^{13}\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4)$.

(4) $\delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_6}^5{x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_6}^4{x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_6}^3{x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_6}^2{x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, x_6{x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, {x_7}^4)$
$= \delta_{\mathcal{C}}(\mathbf{u}^4\mathbf{d}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4, {x_7}^3)$
$= \delta_{\mathcal{C}}(\mathbf{u}^3\mathbf{d}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^5, {x_7}^2)$
$= \delta_{\mathcal{C}}(\mathbf{u}^2\mathbf{d}_1\mathbf{u}^2\mathbf{e}_1\mathbf{u}^5\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^6, x_7)$
$= (\mathbf{u}^4(\mathbf{d}_1\vee\mathbf{e}_1)\mathbf{u}^8\mathbf{e}_1\mathbf{u}^8\mathbf{d}_2\mathbf{u}^8\mathbf{e}_2\mathbf{u}^8\mathbf{d}_3\mathbf{u}^8\mathbf{e}_3\ldots\mathbf{u}^8\mathbf{d}_n\mathbf{u}^8\mathbf{e}_n\mathbf{u}^4)$.

The above elementary computations show that the words $q_{0,0} = {x_0}^9$, $q_{1,1} = {x_1}^5{x_2}^4, q_{2,1} = {x_3}^5{x_4}^4, q_{3,1} = {x_5}^9, q_{4,1} = {x_6}^5{x_7}^4$ satisfy our requirements.

Continuing the proof, let us consider the following four computations.

(0') identity. $\delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{u},\ldots,\mathbf{d}_n,\mathbf{u}), q_{0,0}) = (\mathbf{d}_1,\mathbf{u},\ldots,\mathbf{d}_n,\mathbf{u})$.

(1') n-cycle. $\delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{u},\mathbf{d}_2,\ldots,\mathbf{u},\mathbf{d}_{n-1},\mathbf{u},\mathbf{d}_n,\mathbf{u}), q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})$
$= \delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_2,\ldots,\mathbf{d}_{n-1},\mathbf{d}_{n-1},\mathbf{d}_n,\mathbf{d}_n,\mathbf{d}_1), q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})$
$= \delta_{\mathcal{C}}((\mathbf{u},\mathbf{d}_2,\mathbf{u},\ldots,\mathbf{d}_{n-1},\mathbf{u},\mathbf{d}_n,\mathbf{u},\mathbf{d}_1), q_{2,1}\ldots q_{2,n}) = (\mathbf{d}_2,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{u},\mathbf{d}_n,\mathbf{u},\mathbf{d}_1,$
$\mathbf{u})$.

(2') transposition. $\delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{u},\ldots,\mathbf{d}_n,\mathbf{u}), q_{1,1}q_{2,1}q_{1,2}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}$
$(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_2,\mathbf{u},\ldots,\mathbf{d}_n,\mathbf{u}), q_{2,1}q_{1,2}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}$
$(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_2,\mathbf{d}_1,\mathbf{d}_2,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{u},\mathbf{d}_{n-1},\mathbf{u},\mathbf{d}_n,\mathbf{u}), q_{1,2}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}$
$(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_2,\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_3,\mathbf{d}_3,\ldots,\mathbf{d}_{n-1},\mathbf{d}_{n-1},\mathbf{d}_n,\mathbf{d}_n,\mathbf{d}_2), q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}$
$(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= \delta_{\mathcal{C}}((\mathbf{u},\mathbf{d}_1,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_{n-1},\mathbf{u},\mathbf{d}_n,\mathbf{u},\mathbf{d}_2), q_{2,1}\ldots q_{2,n}$
$(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= \delta((\mathbf{d}_1,\mathbf{u},\mathbf{d}_3,\mathbf{u},\ldots,\mathbf{u},\mathbf{d}_n,\mathbf{u},\mathbf{d}_2,\mathbf{u}), (q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$,
and now applying the n-cycle operation $n-1$ times, we obtain
$\delta((\mathbf{d}_1,\mathbf{u},\mathbf{d}_3,\mathbf{u},\ldots,\mathbf{u},\mathbf{d}_n,\mathbf{u},\mathbf{d}_2,\mathbf{u}), (q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n})^{n-1})$
$= (\mathbf{d}_2,\mathbf{u},\mathbf{d}_1,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_{n-1},\mathbf{u},\mathbf{d}_n,\mathbf{u})$.

(3') collapsing. $\delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{u},\mathbf{d}_2,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{1,1}q_{3,2}q_{4,1}q_{2,1}q_{3,1}q_{2,1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{d}_2,\mathbf{d}_2,\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{3,2}q_{4,1}q_{2,1}q_{3,1}q_{2,1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_1,\mathbf{d}_2,\mathbf{u},\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{4,1}q_{2,1}q_{3,1}q_{2,1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_1\vee\mathbf{d}_2,\mathbf{d}_2,\mathbf{u},\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{2,1}q_{3,1}q_{2,1})$
$= \delta_{\mathcal{C}}((\mathbf{d}_2,\mathbf{d}_1\vee\mathbf{d}_2,\mathbf{u},\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{3,1}q_{2,1})$
$= \delta_{\mathcal{C}}((\mathbf{u},\mathbf{d}_1\vee\mathbf{d}_2,\mathbf{u},\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u}), q_{2,1}) = (\mathbf{d}_1\vee\mathbf{d}_2,\mathbf{u},\mathbf{u},\mathbf{u},\mathbf{d}_3,\ldots,\mathbf{d}_n,\mathbf{u})$.

Put $\hat\gamma_0 = (q_{0,0})^{3n^2+1}$, $\hat\gamma_1 = q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}(q_{0,0})^{3n^2-3n+1}$, $\hat\gamma_2 = q_{1,1}q_{2,1}$ $q_{1,2}\ldots q_{1,n}\ q_{3,1}\ldots q_{3,n}q_{2,1}\ldots q_{2,n}(q_{1,1}\ldots q_{1,n}q_{3,1}\ldots q_{3,n}\ q_{2,1}\ldots q_{2,n})^{n-1}$, $\hat\gamma_3 = q_{1,1}q_{3,2}q_{4,1}$ $q_{2,1}q_{3,1}q_{2,1}(q_{0,0})^{3n^2-5}$, and use the short notation $b_i = \mathbf{u}^{18(i-1)}\mathbf{u}^4\mathbf{v}\mathbf{u}^{13}\mathbf{u}^{18(n-i)}, i = 1, \ldots, n$. By the computations (0'),(1'),(2'),(3'), we get that the $\hat\gamma_j$ ($j = 0,1,2,3$), which all have the same length, induce the following transformations γ_j of $\{b_1,\ldots,b_n\}$:

$$\gamma_0 = \begin{pmatrix} b_1 & b_2 & b_3 & \ldots & b_n \\ b_1 & b_2 & b_3 & \ldots & b_n \end{pmatrix},\quad \gamma_1 = \begin{pmatrix} b_1 & b_2 & \ldots & b_{n-1} & b_n \\ b_2 & b_3 & \ldots & b_n & b_1 \end{pmatrix},$$
$$\gamma_2 = \begin{pmatrix} b_1 & b_2 & b_3 & \ldots & b_n \\ b_2 & b_1 & b_3 & \ldots & b_n \end{pmatrix},\quad \gamma_3 = \begin{pmatrix} b_1 & b_2 & b_3 & \ldots & b_n \\ b_1 & b_1 & b_3 & \ldots & b_n \end{pmatrix}.$$

Using the well-known fact that $\gamma_1,\gamma_2,\gamma_3$ generate all transformations on the n element set $\{b_1,\ldots,b_n\}$, and that γ_0 is the identity, we obtain our technical result. □

Now we are ready to prove our main result.

Theorem 5.2. *Let $\mathcal{A} = (A, X_{\mathcal{A}}, \delta_{\mathcal{A}})$ be an automaton satisfying Letichevsky's criterion. For any automaton $\mathcal{E}$ there exists a primitive power $\mathcal{P}$ of $\mathcal{A}$ such that $\mathcal{E}$ can be represented homomorphically by $\mathcal{P}$.*

Proof: Let $\mathbf{u} = u_1 \ldots u_s, \mathbf{v} = v_1 \ldots v_s$ denote a pair of control words of $\mathcal{A}$ as before. Consider an integer $n \geq 3$ and define the power $\mathcal{N} = \mathcal{A}^{18ns}(X_{\mathcal{N}}, \varphi_1, \ldots, \varphi_{18ns})$, $X_{\mathcal{N}} = A^{6n}$, of $\mathcal{A}$ in the following manner. For any state $(a_1, \ldots, a_{18ns}) \in A^{18ns}$, input letter $\zeta = (z_1, \ldots, z_{6n}) \in X_{\mathcal{N}}$, and t $(= 1, \ldots, 18ns)$, we have

$$\varphi_t(a_1, \ldots, a_{18ns}, \zeta)$$

$$= \begin{cases} x[a_t, a_{t+1}] & \text{if } s \text{ does not divide } t \text{ or} \\ & t = (18(i-1)+j)s, i = 1, \ldots, n, j = 1, 3, 7, 8, 9, \\ & \qquad 10, 12, 16, 17, \\ x[a_t, a_{t+1} \wedge z_{6(i-1)+j/2}] & \text{if } t = (18(i-1)+j)s, i = 1, \ldots, n, j = 2, 4, 6, \\ x[a_t, a_{t+1} \wedge z_{6(i-1)+(j-3)/2}] & \text{if } t = (18(i-1)+j)s, i = 1, \ldots, n, j = 11, 13, 15, \\ x[a_t, a_{t-3s+1} \vee a_{t+1}] & \text{if } t = (18(i-1)+j)s, i = 1, \ldots, n, j = 5, 14, \\ x[a_t, a_{t-18s+1} \vee a_{t+1 (\mathrm{mod}\ 18ns)}] & \text{if } t = 18is, i = 1, \ldots, n. \end{cases}$$

It is easy to check that $\mathcal{N}$ is a primitive power of $\mathcal{A}$, moreover, whenever φ_t $(1 \leq t \leq 18ns)$ really depends on its input variable, then it may additionally depend only on its t^{th} state variable and at most one other state variable. Therefore, $\mathcal{N}$ has the properties required by Proposition 2.4 for the last component of $\mathcal{M}$.

Denote by $\delta_{\mathcal{N}}$ the transition function of $\mathcal{N}$ and consider the automaton $\mathcal{C}$ given in Lemma 5.1. Observe that whenever $\mathcal{N}$ is in the state having the form $(w_{1,1}, \ldots, w_{1,s}, \ldots, w_{18n,1}, \ldots, w_{18n,s}), (w_{i,1} \ldots w_{i,s}) \in \{\mathbf{u}, \mathbf{v}\}, i = 1, \ldots, 18n$, by the effect of words having the form $(z_{1,1}, \ldots, z_{1,6n}) \ldots (z_{s,1}, \ldots, z_{s,6n}), z_{1,i} \ldots z_{s,i} \in \{\mathbf{u}, \mathbf{v}\}, i = 1, \ldots, 6n$, $\delta_{\mathcal{N}}((w_{1,1}, \ldots, w_{1,s}, \ldots, w_{18n,1}, \ldots, w_{18n,s}), (z_{1,1}, \ldots, z_{1,6n}) \ldots (z_{s,1}, \ldots, z_{s,6n})) = (w'_{1,1}, \ldots, w'_{1,s}, \ldots, w'_{18n,1}, \ldots, w'_{18n,s})$ if and only if, $\delta_{\mathcal{C}}((\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(18n)}), (\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(6n)})) = (\mathbf{w}'^{(1)}, \ldots, \mathbf{w}'^{(18n)}), \mathbf{w}^{(i)} = w_{i,1} \ldots w_{i,s}, \mathbf{w}'^{(i)} = w'_{i,1} \ldots w'_{i,s}, \mathbf{z}^{(j)} = z_{1,j} \ldots z_{s,j}, i = 1, \ldots, 18n, j = 1, \ldots, 6n$. Therefore, using the short notation $b_k = \mathbf{u}^{18(k-1)+4}\mathbf{v}\mathbf{u}^{18(n-k)+13}$ for the state b_k of $\mathcal{N}$, by Lemma 5.1 we have that there exists a positive integer m having the following property. For every transformation γ on $\{b_1, \ldots, b_n\}$, there exists a word $\hat{\gamma} = \zeta_1 \ldots \zeta_{ms}$, $(\zeta_1, \ldots, \zeta_{ms} \in X_{\mathcal{N}})$ such that, $\delta_{\mathcal{N}}(b_k, \hat{\gamma}) = \gamma(b_k)$, for all $1 \leq k \leq n$.

Every n-state automaton $\mathcal{E}$ is isomorphic to a subautomaton of an n-state automaton $\mathcal{D}$ with the following properties:
(i) for each transformation γ of the n states of D, there is an input letter x_γ inducing γ, and,
(ii) there are at least as many distinct letters of $\mathcal{D}$ which induce γ as there are which induce γ in $\mathcal{E}$.
Thus, to complete the proof, it suffices to establish the result for the following n

state automaton having these properties whose states are a subset of those of $\mathcal{N}$. Let $\mathcal{D} = (D, X_{\mathcal{D}}, \delta_{\mathcal{D}})$, $D = \{b_1, \ldots, b_n\}$, where $b_k, k = 1, \ldots, n$ are the states of $\mathcal{N}$ discussed before. For each transformation γ of $\{b_1, \ldots, b_n\}$ let there be an input letter x_γ of $\mathcal{D}$ having $\delta_{\mathcal{D}}(b_k, x_\gamma) = \gamma(b_k)$, $k = 1, \ldots, n$. And furthermore, let there be at least as many letters of $\mathcal{D}$ which induce each given transformation γ as there are which in $\mathcal{E}$.

We shall show that $\mathcal{D}$ can be represented homomorphically by a primitive power of $\mathcal{A}$. Clearly, $\{\delta_{\mathcal{D}}(b_k, x) \mid b_k \in D, x \in X_{\mathcal{D}}\} = D$.

To each length ms input word $p = x_1 \ldots x_{ms}$ of $\mathcal{D}$, we associate the transformation γ_p induced by this word on the set $D = \{b_1, \ldots, b_n\}$. Define, following Lemma 5.1, $\tau(p) = \widehat{\gamma_p}$. The mapping $\tau : (X_{\mathcal{D}})^{ms} \to (X_{\mathcal{N}})^{ms}$ satisfies $\delta_{\mathcal{D}}(b_k, p) = \delta_{\mathcal{N}}(b_k, \tau(p))$ $(b_k \in D, p \in (X_D)^{ms})$.

For every $d > 0$, and a a state of $\mathcal{N}$, $(p, q) \in R_{\tau,d}$, $p \in (X_{\mathcal{D}})^{ms}$, we clearly have that, whenever $\delta_{\mathcal{N}}(a, q) \in D$, that $\delta_{\mathcal{D}}(\delta_{\mathcal{N}}(a, q), p) = \delta_{\mathcal{N}}(a, q\tau(p)) \in D$. Furthermore, by taking ι to be a letter of $X_{\mathcal{D}}$ inducing the identity under $\delta_{\mathcal{D}}$ (that is, $\delta_{\mathcal{D}}(b_i, \iota) = b_i$ for all $b_i \in D$), and letting q be $(\tau(\iota^{ms}))^j$ with $d \le msj < ms + d$ and $p = \iota^{d-ms(j-1)}$ implying $(p, q) \in R_{\tau,d}$, we derive $\delta_{\mathcal{D}}(\delta_{\mathcal{N}}(b_i, q), p) = \delta_{\mathcal{D}}(b_i, p) = b_i$. Therefore, $D = \{\delta_{\mathcal{D}}(\delta_{\mathcal{N}}(a, q), p) \mid a \text{ a state of } \mathcal{N}, (p, q) \in R_{\tau,d}, \delta_{\mathcal{N}}(a, q) \in D\}$. This shows that conditions (i) and (ii) of Lemma 4.6 hold.

For every i $(= 1, \ldots, 6n)$, define $\tau_i : (X_{\mathcal{D}})^{ms} \to A^{ms}$ as follows: for each $1 \le j \le ms$, the j^{th} letter of $\tau_i(p)$ $(p \in (X_{\mathcal{D}})^{ms})$ is equal to the i^{th} component of the j^{th} letter $\zeta'_j = (z_{j,1}, \ldots, z_{j,6n})$ of $\tau(p)$. Therefore, as in Lemma 4.6 (taking ℓ and n of the lemma to be $6n$ and ms, respectively), we can construct the product $\mathcal{V} = \mathcal{R}_{\tau_1,d} \times \ldots \times \mathcal{R}_{\tau_{6n},d} \times \mathcal{N}(X_{\mathcal{D}}, \varphi'_1, \ldots, \varphi'_{6n}, \varphi''_{6n+1})$ which homomorphically represents $\mathcal{D}$.

By Lemma 4.4, given an integer $d \ge |X_{\mathcal{D}}|^{ms}$, for each $i = 1, \ldots, 6n$, we obtain a primitive power $\mathcal{M}_i$ of $\mathcal{A}$ such that apart from its last one, its feedback functions do not depend on the last state variable, and furthermore, $\mathcal{M}_i$ y-represents $\mathcal{R}_{\tau_i,d}$.

Now set $\psi_i(m_1, \ldots, m_{6n}, m_{6n+1}, x) = x$ for each $i = 1, \ldots, 6n$ and $\psi_{6n+1}(m_1, \ldots, m_{6n}, m_{6n+1}, x) = (z_1, \ldots, z_{6n})$, where $x \in X_{\mathcal{D}}$, z_i is the state of the last factor of $\mathcal{M}_i$ (which represents $\mathcal{R}_{\tau_i,d}$) for $1 \le i \le 6n$, and m_{6n+1} is the state of $\mathcal{N}$. By Proposition 4.3 (considering $\mathcal{N}, X_{\mathcal{D}}, 6n$ to be $\mathcal{M}, X, m$ of the proposition), we obtain $\mathcal{M} = \mathcal{M}_1 \times \ldots \times \mathcal{M}_{6n} \times \mathcal{N}(X_{\mathcal{D}}, \psi_1, \ldots, \psi_{6n+1})$, which homomorphically represents $\mathcal{V}$, hence $\mathcal{D}$, hence $\mathcal{E}$. On the other hand, observe that we have the conditions of Proposition 2.4 for the product $\mathcal{M}$ (taking $\mathcal{N}, X_{\mathcal{D}}, 6n$ to be $\mathcal{M}_{n+1}, X, n$ of the proposition). By Proposition 2.4, $\mathcal{M}$ is isomorphic to a primitive power $\mathcal{P}$ of $\mathcal{A}$. Therefore, then $\mathcal{E}$ is homomorphically represented by the primitive power $\mathcal{P}$. This completes the proof. □

Corollary 5.3. *Let $\mathcal{K}$ be a class of finite automata. If $\mathcal{K}$ satisfies Letichevsky's criterion, then $\mathcal{K}$ is homomorphically complete under the primitive product.* □

By Letichevsky's result [18], a class of finite automata is homomorphically complete under the Gluškov product if and only if it satisfies Letichevsky's criterion. Therefore, one obtains the following statement.

Theorem 5.4. *Suppose that $\mathcal{K}$ is a class of finite automata. Then the following statements are equivalent:*

- *$\mathcal{K}$ satisfies Letichevsky's criterion.*
- *$\mathcal{K}$ is homomorphically complete under the Gluškov product.*
- *$\mathcal{K}$ is homomorphically complete under the α_i-product for all $i \geq 2$.*
- *$\mathcal{K}$ is homomorphically complete under the α_i-product for some $i \geq 2$.*
- *$\mathcal{K}$ is homomorphically complete under the ν_j-product for all $j \geq 3$.*
- *$\mathcal{K}$ is homomorphically complete under the ν_j-product for some $j \geq 3$.*
- *$\mathcal{K}$ is homomorphically complete under the $\alpha_i - \nu_j$-product for all $i \geq 2, j \geq 3$.*
- *$\mathcal{K}$ is homomorphically complete under the $\alpha_i - \nu_j$-product for some $i \geq 2$, $j \geq 3$.*
- *$\mathcal{K}$ is homomorphically complete under the primitive product.* □

Remark. There is a class of finite automata satisfying Letichevsky's criterion which is homomorphically complete for neither the α_1-product nor the ν_2-product. This shows that the above result is sharp.

Acknowledgements

We thank Prof. Balázs Imreh, whose really great help and hard work were invaluable for us. C. Nehaniv gratefully acknowledges the extraordinary hospitality of L. Kossuth University during Summer 1996.

References

1. B. Austin, K. Henckell, C. Nehaniv, and J. Rhodes, Subsemigroups and Complexity via the Presentation Lemma, *Journal of Pure & Applied Algebra*, **101**, No. 3, (1995) 245–289.

2. G. Chartrand and F. Harary, Planar permutation graphs, *Ann. Inst. Henri Poincaré, Sec. B,* **3** (1967), 433–438.

3. G. A. Dirac and S. Schuster, A theorem of Kuratowski, *Nederl. Akad. Wetensch., Proc. Ser. A* **57** (1954), 343–348.

4. P. Dömösi and Z. Ésik, On the hierarchy of ν_i-products of automata, *Acta Cybernetica*, **8** (1988), 315–323.

5. P. Dömösi and B. Imreh, On ν_i-products of automata, *Acta Cybernetica*, **6** (1983), 149–162.

6. Z. Ésik, Homomorphically complete classes of automata with respect to the α_2-product, *Acta Sci. Math.*, **48** (1985), 135–141.

7. Z. Ésik and Gy. Horváth, The α_2-product is homomorphically general. In: *Papers on Automata Theory* V, No. DM83-3, 49–62, Karl Marx University of Economics, Department of Mathematics, Budapest (1983).

8. F. Gécseg, Composition of automata. In: *Automata, Languages and Programming: 2nd Colloquium, Saarbrücken*, 1974, Lecture Notes in Computer Science, Springer-Verlag, **14**, 351–363.

9. F. Gécseg, *Products of Automata,* EATCS Monographs on Theoretical Computer Science, Vol. **7**, Springer-Verlag, 1986.

10. F. Gécseg and B. Imreh, A comparison of α_i-products and ν_i-products, *Foundation of Control Engineering,* **12** (1987), 3–9.

11. F. Gécseg and H. Jürgensen, On $\alpha_0 - \nu_1$-products of automata, *Theoretical Computer Science*, **80** (1991), 35–51.

12. F. Gécseg, and I. Peák, *Algebraic Theory of Automata,* Disquisitiones Mathematicae Hungaricae **2**, Akadémiai Kiadó, Budapest, 1972.

13. V. M. Gluškov, Abstract theory of automata, *Uspekhi matem. nauk* **16:5** (**101**) (1961), 3–62. Correction: ibid., **17:2** (**104**) (1962), 270.

14. F. Harary, *Graph Theory*, Addison-Wesley Series in Mathematics, Addison-Wesley Publishing Company, Inc., 1969.

15. K. B. Krohn and J. L. Rhodes, Algebraic theory of machines, I. Prime decomposition theorem for finite semi-groups and machines, *Trans. Amer. Math. Soc.* **116** (1965), 450–464.

16. K. B. Krohn, J. L. Rhodes and B. R. Tilson, The prime decomposition theorem of the algebraic theory of machines. In: M. Arbib, ed., *Algebraic Theory of Machines, Languages and Semigroups*, Academic Press, New York (1968).

17. K. Kuratowski, Sur le problème des courbes gauches en topologie, *Fund. Math.*, **15** (1930), 271–283.

18. A. A. Letichevsky, Conditions of completeness for finite automata, *Žurn. Včisl. Matem. i Matem. Fiz.*, **1** (1961), 702–710.

19. C. L. Nehaniv, From Relation to Emulation: The Covering Lemma for Transformation Semigroups, *Journal of Pure & Applied Algebra*, **107**, No. 1, (1996), 75–87.