Tomus 8. Fasciculus 2.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM
CYBERNETICARUM HUNGARICUN

Szeged, 1987
Curat: Universitas Szegediensis de Attila József nominata

8. kötet 2. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK
KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1987. július
A Szegedi József Attila Tudományegyetem gondozásában

On isomorphic realization of automata with α_0-products

Z. ÉSIK

1. Notions and notations

In this section we give a brief summary of some basic concepts to be used in the sequel.

An *automaton* is a triplet $\mathbf{A}=(A, X, \delta)$ with finite *state set* A, finite *input set* X and *transition* $\delta: A\times X\to A$. The sets A and X are nonempty. The transition is also treated in the extended sense, i.e., as a mapping $A\times X^*\to A$, where X^* is the free monoid generated by X. Take a word $p\in X^*$. The *transition induced by* p is the state map $\delta_p: A\to A$ with $\delta_p(a)=\delta(a, p)$ $(a\in A)$. The collection of these transitions forms a monoid $S(\mathbf{A})$ under composition of mappings. We call $S(\mathbf{A})$ the *characteristic monoid* of $\mathbf{A}$.

The concepts as *subautomaton, homomorphism, congruence relation* and *isomorphism* are used with their usual meaning. Given an automaton $\mathbf{A}=(A, X, \delta)$ and a state $a\in A$, the *subautomaton generated by* a has state set $\{\delta(a, p)|p\in X^*\}$. An automaton (B, Y, δ') is an *X-subautomaton* of an automaton (A, X, δ) if $B\subseteq A$, $Y\subseteq X$ and δ' is the restriction of δ to $B\times Y$. The *factor automaton* of an automaton $\mathbf{A}$ with respect to a congruence relation θ of $\mathbf{A}$ is denoted $\mathbf{A}/\theta$. We write $\theta_1\prec\theta_2$ to mean that θ_1 is a refinement of θ_2 and $\theta_1\neq\theta_2$. An automaton is called *simple* if it has only the trivial congruence relations ω (identity relation) and ι (total relation). Thus *trivial* (i.e., one-state) automata are simple.

Let $\mathbf{A}_i=(A_i, X_i, \delta_i)$ $(i=1, \ldots, n,\ n\geqq 0)$ be automata. Take a finite nonempty set X and a family of *feedback functions* $\varphi_i: A_1\times\ldots\times A_n\times X\to X_i$ $(i=1, \ldots, n)$. By the *product* $\mathbf{A}_1\times\ldots\times\mathbf{A}_n[X, \varphi]$ we mean the automaton $(A_1\times\ldots\times A_n, X, \delta)$, where

$$\delta((a_1, \ldots, a_n), x) = (\delta_1(a_1, x_1), \ldots, \delta_n(a_n, x_n))$$

with

$$x_i = \varphi_i(a_1, \ldots, a_n, x) \quad (i = 1, \ldots, n)$$

for all $(a_1, \ldots, a_n)\in A_1\times\ldots\times A_n$ and $x\in X$. The integer n is referred to as the length of the product. If, for every i, φ_i is independent of the state variables $a_i, \ldots, a_n$, we speak about an α_0-*product*. In an α_0-product a feedback function φ_i is alternatively treated as a mapping $A_1\times\ldots\times A_{i-1}\times X\to X_i$. Moreover, φ_i extends to a mapping $A_1\times\ldots\times A_{i-1}\times X^*\to X_i^*$ in a natural way.

Let $\mathcal{K}$ be a (possibly empty) class of automata. We will use the following notations:

$\mathbf{P}_{\alpha_0}(\mathcal{K})$:=all α_0-products of automata from $\mathcal{K}$;
$\mathbf{P}_{1\alpha_0}(\mathcal{K})$:=all α_0-products with length at most 1 of automata from $\mathcal{K}$;
$\mathbf{S}(\mathcal{K})$:=all subautomata of automata from $\mathcal{K}$;
$\mathbf{H}(\mathcal{K})$:=all homomorphic images of automata from $\mathcal{K}$;
$\mathbf{I}(\mathcal{K})$:=all isomorphic images of automata from $\mathcal{K}$;
$\mathcal{K}^*$:=the collection of all automata $\mathbf{A}=(A, X, \delta)$ such that there is an automaton $\mathbf{B}=(A, Y, \delta')\in\mathcal{K}$ with the following properties: (i) $\mathbf{B}$ is an X-subautomaton of $\mathbf{A}$; (ii) for every sign $x\in X$ there is a word $p\in Y^*$ inducing the same transition as p, i.e., $\delta'_p=\delta_x$. (Note that we have $S(\mathbf{A})=S(\mathbf{B})$.)

We call a class $\mathcal{K}$ of automata an α_0-*variety* if it is closed under $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{P}_{\alpha_0}$. An α_0-variety is never empty. An α_0^*-*variety* is an α_0-variety $\mathcal{K}$ with $\mathcal{K}^*\subseteq\mathcal{K}$. For later use we note that $\mathbf{HSP}_{\alpha_0}(\mathcal{K})$ ($\mathbf{HSP}_{\alpha_0}(\mathcal{K}^*)$) is the smallest α_0-variety (α_0^*-variety) containing a class $\mathcal{K}$. Similarly, $\mathbf{ISP}_{\alpha_0}(\mathcal{K})$ is the smallest class containing $\mathcal{K}$ and closed under $\mathbf{I}$, $\mathbf{S}$ and $\mathbf{P}_{\alpha_0}$. It is worth noting that $\mathbf{SP}_{1\alpha_0}(\mathcal{K})$ contains all X-subautomata of automata in $\mathcal{K}$.

A class $\mathcal{K}_0$ is said to be *isomorphically* α_0-*complete* for $\mathcal{K}$ if $\mathcal{K}\subseteq\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$. The following statement is a direct consequence of results in [5] (see also [3], [4]):

Proposition 1.1. If $\mathcal{K}_0$ is isomorphically α_0-complete for $\mathcal{K}$ and $\mathbf{A}\in\mathcal{K}$ is a simple automaton then $\mathbf{A}\in\mathbf{ISP}_{1\alpha_0}(\mathcal{K}_0)$.

Thus, any isomorphically α_0-complete class for $\mathcal{K}$ must "essentially" contain all simple automata in $\mathcal{K}$. The converse fails in general, yet it holds for some important classes: the class of all automata and the classes of permutation automata, monotone automata and definite automata are equally good examples (see [2], [3], [6], [7], [9]). Isomorphically α_0-complete classes for the class of all commutative automata essentially consist of automata very close to simple commutative automata (cf. [7]). In a sense there is a unique nontrivial simple nilpotent automaton. On the other hand no finite subclass of nilpotent automata is isomorphically α_0-complete for the class of all nilpotent automata. Thus, the class of nilpotent automata is a counterexample. Isomorphically α_0-complete classes for nilpotent automata are studied in [8].

Some more notation. The cardinality of a set A is denoted $|A|$. The symbol $\mathbf{E}$ denotes the automaton $(\{0, 1\}, \{x_0, x_1\}, \delta)$ with $\delta(0, x_0)=0$, $\delta(0, x_1)=\delta(1, x_0)=\delta(1, x_1)=1$. We call $\mathbf{E}$ the *elevator*.

The relation of the α_0-product to other product concepts is explained in [3]. The Krohn—Rhodes Decomposition Theorem gives a basis for studying α_0-products. For this, see [1], [3], [4].

2. Preliminary results

Let $\mathbf{A}=(A, X, \delta)$ be an automaton. As usual, we say that $\mathbf{A}$ is *strongly connected* if it is generated by any state $a\in A$. Further, $\mathbf{A}$ is called a *cone* if there is a state $a_0\in A$ with the following properties:

(i) $\delta(a_0, x)=a_0$, for all $x\in X$,
(ii) $A-\{a_0\}$ is nonempty and every state $a\in A-\{a_0\}$ generates $\mathbf{A}$.

Obviously, the state a_0 with the above properties is unique, whence it will be referred to as the *apex* of **A**. The set $A-\{a_0\}$ constitutes the *base* of **A**. It should be noted that every simple automaton is either a strongly connected automaton or a cone or an automaton $(\{a_1, a_2\}, X, \delta)$ with $\delta(a_i, x)=a_i$, $i=1, 2$, $x\in X$.

Theorem 2.1. Let $\mathcal{K}$ be a class of automata with $\mathbf{H}(\mathcal{K})\subseteqq\mathcal{K}$, $\mathbf{S}(\mathcal{K})\subseteqq\mathcal{K}$ and $\mathcal{K}^*\subseteqq\mathcal{K}$. If $\mathbf{E}\in\mathcal{K}$ then for an arbitrary class $\mathcal{K}_0$, $\mathcal{K}\subseteqq\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$ if and only if every strongly connected automaton and every cone belonging to $\mathcal{K}$ is in $\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$.

Proof. The necessity of the statement is trivial. For the sufficiency let $\mathbf{A}=(A, X, \delta)$ be an automaton in $\mathcal{K}$. We are going to apply induction on $|A|$ to show that $\mathbf{A}\in\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$. Since $\mathcal{K}^*\subseteqq\mathcal{K}$ and $\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$ is closed under X-subautomata, it can be assumed that for every word $p\in X^*$ there is a sign $\bar{p}\in X$ inducing the same transition as p, i.e., $\delta(a, \bar{p})=\delta(a, p)$ for all $a\in A$.

If $|A|=1$ then **A** is strongly connected and $\mathbf{A}\in\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$. Suppose that $|A|>1$. If **A** is strongly connected or a cone then $\mathbf{A}\in\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$ by assumption. Otherwise two cases arise.

Case 1: **A** contains a nontrivial proper subautomaton $\mathbf{B}=(B, X, \delta)$ generated by a state $b_0\in B$. Let $\varrho\subseteqq A\times A$ be the relation defined by $a\varrho b$ if and only if $a=b$ or $a, b\in B$. A straightforward computation proves that ϱ is a congruence relation of **A**. For every state $b\in B$ fix an $x_b\in X$ with $\delta(b_0, x_b)=b$. Take the α_0-product

$$\mathbf{C}=(C, X, \delta')=\mathbf{A}/\varrho\times\mathbf{B}[X, \varphi],$$

where $\varphi_1(x)=x$,

$$\varphi_2(\{a\}, x)=\begin{cases} x_{b_0} & \text{if } \delta(a, x)\notin B, \\ x_b & \text{if } \delta(a, x)=b\in B \end{cases}$$

and $\varphi_2(B, x)=x$ for every $x\in X$ and $a\in A-B$. Set

$$C'=\{(\{a\}, b_0)\mid a\in A-B\}\cup\{(B, b)\mid b\in B\}.$$

It is immediately seen that $\mathbf{C}'=(C', X, \delta')$ is a subautomaton of **C** isomorphic to **A**. Since both $\mathbf{A}/\varrho$ and **B** are in $\mathcal{K}$ and have fewer states than **A**, we have $\mathbf{A}/\varrho$, $\mathbf{B}\in\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$ from the induction hypothesis. The result follows by the fact that $\mathbf{ISP}_{\alpha_0}(\mathcal{K}_0)$ is closed under **I**, **S** and $\mathbf{P}_{\alpha_0}$.

Case 2: There are distinct states $a_1, a_2\in A$ with $\delta(a_i, x)=a_i$, $i=1, 2$, $x\in X$. Define $\varrho\subseteqq A\times A$ by $a\varrho b$ if and only if $a=b$ or $a, b\in\{a_1, a_2\}$. Again, ϱ is a congruence relation of **A**. Let

$$\mathbf{C}=(C, X, \delta')=\mathbf{A}/\varrho\times\mathbf{E}[X, \varphi]$$

be the α_0-product with $\varphi_1(x)=x$,

$$\varphi_2(\{a\}, x)=\begin{cases} x_1 & \text{if } \delta(a, x)=a_2, \\ x_0 & \text{otherwise} \end{cases}$$

and $\varphi_2(\{a_1, a_2\}, x)=x_0$, where $x\in X$ and $a\in A-\{a_1, a_2\}$. It follows that $\mathbf{C}'=(C', X, \delta')$ with

$$C'=\{(\{a\}, 0)\mid a\in A-\{a_1, a_2\}\}\cup\{(\{a_1, a_2\}, 0), (\{a_1, a_2\}, 1)\}$$

is a subautomaton of $\mathbf{C}$ isomorphic to $\mathbf{A}$. Since $\mathscr{K}$ is closed under homomorphic images and $\mathbf{A}/\varrho$ has fewer states than $\mathbf{A}$ we have $\mathbf{A}/\varrho \in \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ from the induction hypothesis. On the other hand, $\mathbf{E} \in \mathscr{K}$ and $\mathbf{E}$ is a cone. Thus $\mathbf{E} \in \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ and we conclude $\mathbf{A} \in \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$.

Remark. Let $\mathscr{K}$ be a class as in Theorem 2.1, i.e. $\mathscr{K}^* \subseteq \mathscr{K}$, $\mathbf{H}(\mathscr{K}) \subseteq \mathscr{K}$ and $\mathbf{S}(\mathscr{K}) \subseteq \mathscr{K}$. Assuming $\mathbf{E} \notin \mathscr{K}$ it follows that $\mathscr{K}$ consists of permutation automata. (See the last section for the definition of permutation automata.) Every permutation automaton is the disjoint sum of strongly connected permutation automata. Now obviously, if $\mathscr{K}$ contains a nontrivial strongly connected automaton then $\mathscr{K} \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ for a class $\mathscr{K}_0$ if and only if $\mathbf{A} \in \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ for every strongly connected permutation automaton $\mathbf{A} \in \mathscr{K}$. (Or even, the same holds if α_0-product is replaced by the so-called quasi-direct product.) If in addition $\mathscr{K}$ is closed under X-subautomata then, as we shall see later, $\mathscr{K} \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ if and only if every simple strongly connected permutation automaton in $\mathscr{K}$ is already contained by $\mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$. Suppose now that every strongly connected automaton in $\mathscr{K}$ is trivial. Then, if $\mathscr{K}$ contains a nontrivial automaton, we have $\mathscr{K} \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ if and only if $(\{0, 1\}, \{x\}, \delta) \in \mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$ with $\delta(0, x)=0$ and $\delta(1, x)=1$. Further, $\mathscr{K} \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ holds for every $\mathscr{K}_0$ if $\mathscr{K}$ consists of trivial automata.

The following two lemmas establish some simple facts about homomorphic realization of cones and strongly connected automata in the presence of $\mathbf{E}$.

Lemma 2.2. Let $\mathbf{A}=(A, X, \delta)$ be a cone in $\mathbf{HSP}_{\alpha_0}(\mathscr{K} \cup \{\mathbf{E}\})$. There exist an automaton $\mathbf{D} \in \mathbf{P}_{\alpha_0}(\mathscr{K})$ and an α_0-product $\mathbf{D} \times \mathbf{E}[X, \varphi]$ containing a subautomaton that can be mapped homomorphically onto $\mathbf{A}$.

Proof. Let $\mathbf{B}=(B, X, \delta')=\mathbf{B}_1 \times \ldots \times \mathbf{B}_n[X, \psi]$ be an α_0-product with $\mathbf{B}_t \in \mathscr{K} \cup \{\mathbf{E}\}$, $t=1, \ldots, n$. Let $\mathbf{C}=(C, X, \delta')$ be a subautomaton of $\mathbf{B}$ and $h: C \to A$ a homomorphism of $\mathbf{C}$ onto $\mathbf{A}$. We may assume $\mathbf{C}$ to be in a sense minimal: no proper subautomaton of $\mathbf{C}$ is mapped homomorphically onto $\mathbf{A}$.

Denote by a_0 the apex and by A_0 the base of $\mathbf{A}$. Set $C_0=h^{-1}(A_0)$, $C_1=h^{-1}(\{a_0\})$. Clearly then $\mathbf{C}_1=(C, X, \delta')$ is a subautomaton of $\mathbf{C}$, and $\mathbf{C}$ is generated by any state $a \in C_0$.

Let $1 \leqq i_1 < \ldots < i_r \leqq n$ be all the indices $t=1, \ldots, n$ with $\mathbf{B}_t \in \mathscr{K}$. If $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in C_0$, we have $a_t=b_t$ whenever $t \notin \{i_1, \ldots, i_r\}$ for otherwise $\mathbf{C}$ would not be generated by every state in C_0. Let $j_1, \ldots, j_s \in \{1, \ldots, n\} - \{i_1, \ldots, i_r\}$ be those indices t such that for any $(a_1, \ldots, a_n) \in C_0$, $a_t=0$ if and only if $t \in \{j_1, \ldots, j_s\}$. For every $a=(a_1, \ldots, a_r) \in B_{i_1} \times \ldots \times B_{i_r}$ put $\bar{a}=(\bar{a}_1, \ldots, \bar{a}_n) \in B$ with $\bar{a}_{i_1}=a_1, \ldots, \bar{a}_{i_r}=a_r, \bar{a}_{j_1}=\ldots=\bar{a}_{j_s}=0$ and $\bar{a}_t=1$ otherwise.

To end the proof we give an α_0-product $\mathbf{B}'=\mathbf{B}_{i_1} \times \ldots \times \mathbf{B}_{i_r} \times \mathbf{E}[X, \psi']$ and a subautomaton $\mathbf{C}'=(C', X, \delta'')$ of $\mathbf{B}'$ such that $\mathbf{A}$ is a homomorphic image of $\mathbf{C}'$. For every $a \in B_{i_1} \times \ldots \times B_{i_r}$, $i=0, 1$, $x \in X$ and $j=1, \ldots, r$, define

$$\psi'_j(a, i, x) = \psi_{i_j}(\bar{a}, x),$$

$$\psi'_{r+1}(a, i, x) = \begin{cases} x_1 & \text{if } \delta'(\bar{a}, x) \in C_1, \\ x_0 & \text{otherwise.} \end{cases}$$

Let $\mathbf{C}'$ be the subautomaton generated by the set

$$C_0' = \{(a, 0) \mid a \in B_{i_1} \times \ldots \times B_{i_r},\ \bar{a} \in C_0\}.$$

Set $C_1' = C' - C_0'$. It is clear from the construction that states in C_1' have 1 as their last components. Therefore, C_1' is the state set of a subautomaton of $\mathbf{C}'$. Moreover, for every $(a, 0), (b, 0) \in C_0'$ and $x \in X$ we have $\delta''((a, 0), x) = (b, 0)$ if and only if $\delta'(\bar{a}, x) = \bar{b}$, while $\delta''((a, 0), x) \in C_1'$ if and only if $\delta'(\bar{a}, x) \in C_1$. It follows that $\mathbf{A}$ is a homomorphic image of $\mathbf{C}'$, a homomorphism being the map that takes each state in C_1' to a_0 and each state $(a, 0) \in C_0'$ to $h(\bar{a})$.

If $\mathbf{A}$ were strongly connected we would not need the last factor of the α_0-product $\mathbf{B}'$ either. This gives the following:

Lemma 2.3. Every strongly connected automaton in $\mathbf{HSP}_{\alpha_0}(\mathcal{K} \cup \{\mathbf{E}\})$ is contained in $\mathbf{HSP}_{\alpha_0}(\mathcal{K})$.

Let $\mathbf{A} = (A, X, \delta)$ be a cone with apex a_0 and base A_0. Suppose that the relation $\varrho \subseteq A \times A$ defined by $a \varrho b$ if and only if $a = b = a_0$ or $a, b \in A_0$ is a congruence relation of $\mathbf{A}$, which is to say that for every $x \in X$ either $\delta(A_0, x) \subseteq A_0$ or $\delta(A_0, x) = \{a_0\}$. Set $X_0 = \{x \in X \mid \delta(A_0, x) \subseteq A_0\}$. Assuming $X_0 \neq \emptyset$, the automaton $\mathbf{A}_0 = (A_0, X_0, \delta)$ is a strongly connected X-subautomaton of $\mathbf{A}$, which is guaranteed if $|A_0| > 1$. By definition, we call $\mathbf{A}$ a 0-simple cone if and only if $X_0 \neq \emptyset$ and $\mathbf{A}_0$ is simple. Thus, $\mathbf{E}$ is both a simple cone and a 0-simple cone. Given a strongly connected automaton $\mathbf{A}_0 = (A_0, X_0, \delta_0)$, there is a natural way to imbed $\mathbf{A}_0$ into a 0-simple cone $\mathbf{A}_0^c$: define $\mathbf{A}_0^c = (A \cup \{a_0\}, X_0 \cup \{x_0\}, \delta)$ where $a_0 \notin A_0$, $x_0 \notin X_0$, $\delta(a, x_0) = a_0$ for every $a \in A_0 \cup \{a_0\}$ and $\delta(a_0, x) = a_0$, $\delta(a, x) = \delta_0(a, x)$ if $a \in A_0$, $x \in X_0$. Obviously, $\mathbf{A}_0^c$ is 0-simple if and only if $\mathbf{A}_0$ is simple.

If $\mathbf{A}$ is a simple cone (i.e., a simple automaton that is a cone) then $\mathbf{A} \in \mathbf{ISP}_{\alpha_0}(\mathcal{K})$ for a class $\mathcal{K}$ if and only if $\mathbf{A} \in \mathbf{ISP}_{1\alpha_0}(\mathcal{K})$. In the next statement we investigate what can be said about $\mathcal{K}$ if $\mathbf{ISP}_{\alpha_0}(\mathcal{K})$ contains a 0-simple cone.

Lemma 2.4. If a 0-simple cone $\mathbf{A} = \mathbf{A}_0^c$ is in $\mathbf{ISP}_{\alpha_0}(\mathcal{K})$ then either $\mathbf{A} \in \mathbf{ISP}_{1\alpha_0}(\mathcal{K})$ or $\mathbf{E} \in \mathbf{ISP}_{1\alpha_0}(\mathcal{K})$ and there is an automaton $\mathbf{D} \in \mathcal{K}$ such that $\mathbf{A}$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{D}$.

Proof. Let $\mathbf{A}_0 = (A_0, X_0, \delta_0)$ and $\mathbf{A} = (A, X, \delta)$ so that $A = A_0 \cup \{a_0\}$, $X = X_0 \cup \{x_0\}$ where $a_0 \notin A_0$, $x_0 \notin X_0$, $\delta(a, x_0) = a_0$ $(a \in A)$, $\delta(a_0, x) = a_0$ and $\delta(a, x) = \delta_0(a, x)$ $(a \in A_0, x \in X_0)$. Since $\mathbf{A} \in \mathbf{ISP}_{\alpha_0}(\mathcal{K})$ there exist an α_0-product $\mathbf{B} = (B, X, \delta') = \mathbf{B}_1 \times \ldots \times \mathbf{B}_n[X, \varphi]$ $(\mathbf{B}_t \in \mathcal{K},\ t = 1, \ldots, n)$ and a subautomaton $\mathbf{C} = (C, X, \delta')$ of $\mathbf{B}$ such that $\mathbf{A}$ is isomorphic to $\mathbf{C}$ under a mapping $h: A \to C$. We may assume that n is minimal, i.e., whenever an α_0-product of automata from $\mathcal{K}$ contains a subautomaton isomorphic to $\mathbf{A}$, the length of that product is at least n.

Suppose that $\mathbf{A} \notin \mathbf{ISP}_{1\alpha_0}(\mathcal{K})$. We then have $n > 1$. Let $a = (a_1, \ldots, a_n)$ and $b = (b_1, \ldots, b_n)$ be arbitrary states in C. For every $t = 1, \ldots, n$, put $a \theta_t b$ if and only if $a_1 = b_1, \ldots, a_t = b_t$. Further, let $a \varrho b$ if and only if $a = b = h(a_0)$ or $a, b \in h(A_0)$. Each of these relations is a congruence relation of $\mathbf{C}$, and since n is minimal, $\theta_1 > \ldots > \theta_n (= \omega)$ and $\theta_1 \neq \iota$. Since $\mathbf{A}$ is 0-simple this leaves $n = 2$, $\theta_1 = \varrho$ and $\theta_2 = \omega$. It then follows that $\mathbf{E}$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{B}_1$ with a single factor and $\mathbf{A}$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{B}_2$.

Let $\mathbf{A}_0^c=(A_0\cup\{a_0\}, X_0\cup\{x_0\}, \delta)$ be a 0-simple cone with $\mathbf{A}_0=(A_0, X_0, \delta_0)$, and take an arbitrary automaton $\mathbf{B}=(B, Y, \delta')$. It is not difficult to give a necessary and sufficient condition ensuring that $\mathbf{A}_0^c$ is isomorphic to an α_0-product of $\mathbf{E}$ with $\mathbf{B}$. Clearly this can happen if and only if there are a pair of functions $h: A_0\to B$, $\varphi: X_0\to Y$, a state $b_0\in B$ and two not necessarily distinct signs $y_0, y_1\in Y$ such that:

(i) h is injective;
(ii) for every $a_1, a_2\in A_0$ and $x\in X_0$ we have $\delta_0(a_1, x)=a_2$ if and only if $\delta'(h(a_1), \varphi(x))=h(a_2)$;
(iii) $\delta'(h(A_0), y_0)=\{b_0\}$, $\delta'(b_0, y_1)=b_0$.

If also $b_0\notin h(A_0)$ and $y_0=y_1$ then $\mathbf{A}_0^c$ is isomorphic to an α_0-product of $\mathbf{B}$ with a single factor.

3. The main result

An automaton $\mathbf{A}=(A, X, \delta)$ is called *permutation automaton* if δ_x is a permutation of the state set for every $x\in X$. This is equivalent to saying that δ_p is a permutation for every $p\in X^*$ or that $S(\mathbf{A})$ is a group. Let $\mathscr{K}_p$ denote the class of all permutation automata. It is known that $\mathscr{K}_p$ is an α_0^*-variety, see [1]. Moreover, from the Krohn—Rhodes Decomposition Theorem we have $\mathscr{K}_p=\mathbf{HSP}_{\alpha_0}(\{\mathbf{A}(G)|G \text{ is a simple group}\})$ where the group-like automaton $\mathbf{A}(G)$ on a (finite) group G is defined to be the automaton (G, G, δ) with $\delta(g, h)=gh$, $g, h\in G$.

Another class of automata we shall be dealing with is the class $\mathscr{K}_m$ of all monotone automata. By definition, an automaton $\mathbf{A}=(A, X, \delta)$ is *monotone* if $\delta(a, pq)=a$ implies $\delta(a, p)=a$, for all $a\in A$ and $p, q\in X^*$. This is equivalent to requiring the existence of an ordering $\leqq$ on A such that $a\leqq\delta(a, p)$ for all $a\in A$ and $p\in X^*$ (or $a\leqq\delta(a, x)$ for all $a\in A$ and $x\in X$). The class $\mathscr{K}_m$ is known to be an α_0^*-variety. Further, it is the α_0-variety generated by $\mathbf{E}$, i.e. $\mathscr{K}_m=\mathbf{HSP}_{\alpha_0}(\{\mathbf{E}\})$ (see [1], [10], [11]).

Having defined the classes $\mathscr{K}_p$ and $\mathscr{K}_m$, put $\mathscr{K}_{pm}=\mathbf{HSP}_{\alpha_0}(\mathscr{K}_p\cup\mathscr{K}_m)=$ $=\mathbf{HSP}_{\alpha_0}(\mathscr{K}_p\cup\{\mathbf{E}\})=\mathbf{HSP}_{\alpha_0}(\{\mathbf{A}(G)|G \text{ is a simple group}\}\cup\{\mathbf{E}\})$. It follows from Stiffler's switching rules that $\mathbf{A}\in\mathscr{K}_{pm}$ if and only if there is an α_0-product $\mathbf{B}$ of a permutation automaton with a monotone automaton such that $\mathbf{A}\in\mathbf{HS}(\{\mathbf{B}\})$. For this and other characterizations of the class $\mathscr{K}_{pm}$, see [1] and [10]. It is immediate from our definition that $\mathscr{K}_{pm}$ is an α_0-variety. Or even, it is an α_0^*-variety.

Lemma 3.1. Let $\mathbf{A}$ be a strongly connected automaton. Then $\mathbf{A}\in\mathscr{K}_{pm}$ if and only if $\mathbf{A}\in\mathscr{K}_p$.

Proof. Use Lemma 2.3.

Corollary. If $\mathbf{A}=\mathbf{A}_0^c$ is a cone in $\mathscr{K}_{pm}$ then $\mathbf{A}_0$ a strongly connected permutation automaton.

Lemma 3.2. Let $\mathbf{A}=(A, X, \delta)\in\mathscr{K}_{pm}$ be a cone with apex a_0 and base A_0. If $\delta(a, p)=\delta(b, p)\in A_0$ holds for some $a, b\in A_0$ and $p\in X^*$ then $a=b$.

Proof. From Lemma 2.2 it follows that $\mathbf{A}$ is a homomorphic image of a subautomaton $\mathbf{C}=(C, X, \delta')$ of an α_0-product $\mathbf{B}\times\mathbf{E}[X, \varphi]$ where $\mathbf{B}$ is a permutation

automaton, say $\mathbf{B}=(B, X_1, \delta_1)$. Denote by h an onto homomorphism $\mathbf{C}\to\mathbf{A}$. Set $C_0=h^{-1}(A_0)$. We may assume that every state in C_0 is a generator of $\mathbf{C}$. Each state in C_0 must have 0 as its second component since otherwise we would have $C\subseteqq B\times\{1\}$, and this would yield that $\mathbf{C}$ and $\mathbf{A}$ are permutation automata.

Let $(a_1, 0), (b_1, 0)\in C_0$ with $h(a_1, 0)=a$, $h(b_1, 0)=b$. Take a word $q\in X^*$ with $\delta(a, pq)=a$. We have $\delta(a, (pq)^n)=\delta(b, (pq)^n)=a$, and hence $\delta'((a_1, 0), (pq)^n)$, $\delta'((b_1, 0), (pq)^n)\in C_0$, for all $n\geqq 1$. Define $r=\varphi_1(pq)$. For every integer $n\geqq 1$ we have $\delta'((a_1, 0), (pq)^n)=(\delta_1(a_1, r^n), 0)$ and $\delta'((b_1, 0), (pq)^n)=(\delta_1(b_1, r^n), 0)$, Since $\mathbf{B}$ is a permutation automaton, there is an $n\geqq 1$ with $a_1=\delta_1(a_1, r^n)$ and $b_1=\delta_1(b_1, r^n)$. Thus we obtain $a=h(a_1, 0)=h(\delta'((a_1, 0), (pq)^n))=h(\delta'((b_1, 0), (pq)^n))=h(b_1, 0)=b$.

Theorem 3.3. Let $\mathscr{K}\subseteqq\mathscr{K}_{pm}$ be a class containing $\mathbf{E}$, closed under X-subautomata and homomorphic images and such that $\mathscr{K}^*\subseteqq\mathscr{K}$. A class $\mathscr{K}_0$ is isomorphically α_0-complete for $\mathscr{K}$ if and only if the following conditions hold:

(i) every simple cone and every simple strongly connected permutation automaton belonging to $\mathscr{K}$ is in $\mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$,

(ii) for every 0-simple cone $\mathbf{A}_0^c\in\mathscr{K}$ there is a $\mathbf{B}\in\mathscr{K}_0$ such that $\mathbf{A}_0^c$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{B}$.

Proof. The necessity of (i) comes from Proposition 1.1 while (ii) is necessary in virtue of Lemma 2.4.

For the converse recall that $\mathscr{K}$ satisfies the assumptions of Theorem 2.1. Therefore, by Theorem 2.1, it suffices to show that every strongly connected automaton and every cone belonging to $\mathscr{K}$ is contained by $\mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$.

Let $\mathbf{A}=(A, X, \delta)\in\mathscr{K}$ be a cone with base A_0 and apex a_0. Since $\mathscr{K}^*\subseteqq\mathscr{K}$ and $\mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ is closed under X-subautomata, we may assume that for every $p\in X^*$ there is a $\bar{p}\in X$ inducing the same transition as p. If $\mathbf{A}$ is simple then $\mathbf{A}\in\mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$ by (i). If $\mathbf{A}$ is 0-simple then $\mathbf{A}$ is isomorphic to an α_0-product $\mathbf{A}_0^c[X, \varphi]$ with a single factor where $\mathbf{A}_0^c\in\mathscr{K}$ is a 0-simple cone. (Recall that $\mathscr{K}$ is closed under X-subautomata.) Therefore, we may assume that $\mathbf{A}$ is of the form $\mathbf{A}_0^c$. Now, by (ii), $\mathbf{A}$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{B}$ where $\mathbf{B}\in\mathscr{K}_0$. Since $\mathbf{E}$ is a simple cone we have $\mathbf{E}\in\mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$. It follows that $\mathbf{A}\in\mathbf{ISP}_{\alpha_0}(\mathscr{K}_0)$. Suppose that $\mathbf{A}$ is neither simple nor 0-simple. We proceed by induction on $|A|$. If $|A|=2$ our statement holds vacantly. Let $|A|>2$. There exists a congruence relation $\theta\neq\omega$ of $\mathbf{A}$ such that $a\theta b$ implies $a=b$ or $a, b\in A_0$, and such that A_0 contains at least two blocks of the partition induced by θ.

Let $C_0=\{a_0\}, C_1, \ldots, C_n$ $(n\geqq 2, |C_1|>1)$ be the blocks of θ. Since $\mathbf{A}$ is generated by any state in A_0, from Lemma 3.2 we have the following: for every $i, j\in\{1, \ldots, n\}$ there exists a word $p\in X^*$ with $\delta(C_i, p)=C_j$. Consequently, for every $i\in\{1, \ldots, n\}$ there is a pair of words (p_i, q_i) with $\delta(C_1, p_i)=C_i$, $\delta(C_i, q_i)=C_1$ and such that p_iq_i induces the identity map on C_1 while q_ip_i induces the identity map on C_i.

Set $X'=\{x\in X|\delta(C_1, x)\subseteqq C_0\cup C_1\}$, $\mathbf{C}=(C_0\cup C_1, X', \delta')$, where $\delta'(c, x)=\delta(c, x)$ for all $c\in C_0\cup C_1$ and $x\in X'$. Obviously, both $\mathbf{A}/\theta$ and $\mathbf{C}$ are cones in $\mathscr{K}$. Fix a sign $x_0\in X'$ with $\delta'(C_1, x_0)=C_0$. Take the α_0-product

$$\mathbf{B}=(B, X, \delta'')=\mathbf{A}/\theta\times\mathbf{C}[X, \varphi]$$

where $\varphi_1(x)=x$ and

$$\varphi_2(C_i, x) = \begin{cases} x_0 & \text{if} \quad \delta(C_i, x) = C_0 \\ \overline{p_i x q_j} & \text{if} \quad \delta(C_i, x) = C_j \quad \text{and} \quad i, j \neq 0. \end{cases}$$

It is easy to check that $\mathbf{B}'=(B', X, \delta'')$ is a subautomaton of $\mathbf{B}$ where

$$B' = \{(C_0, a_0)\} \cup \{(C_i, a) \mid i = 1, \ldots, n,\ a \in C_1\}.$$

Further, the map $(C_0, a_0) \mapsto a_0$, $(C_i, a) \mapsto \delta(a, p_i)$ $(i=1, \ldots, n,\ a \in C_1)$ is an isomorphism of $\mathbf{B}'$ onto $\mathbf{A}$. Hence the result follows from the induction hypothesis.

Suppose now that $\mathbf{A}=(A, X, \delta) \in \mathscr{K}$ is a strongly connected automaton. From Lemma 3.1 we know that $\mathbf{A}$ is a permutation automaton. Just as before, we may assume that for every $p \in X^*$ there is a sign $\bar{p} \in X$ with $\delta_p = \delta_{\bar{p}}$. If $\mathbf{A}$ is simple then $\mathbf{A} \in \mathbf{ISP}_{1\alpha_0}(\mathscr{K}) \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K})$. Otherwise let θ be a congruence relation of $\mathbf{A}$ different from ω and ι. Denote by $C_1, \ldots, C_n$ $(n \geqq 2,\ |C_1| > 1)$ the blocks of the partition induced by θ. Set $X' = \{x \in X \mid \delta(C_1, x) = C_1\}$. One shows that $\mathbf{A}$ is isomorphic to an α_0-product of $\mathbf{A}/\theta$ with $\mathbf{C}$, where $\mathbf{C}=(C_1, X', \delta')$, $\delta'(c, x)=\delta(c, x)$ $(c \in C_1,\ x \in X')$.

We note that a substantial part of the above proof as well as the proofs of Theorem 2.1 and Lemma 2.2 follow well-known ideas (see [1], [4], [5]).

Corollary. Let $\mathscr{K} \subseteq \mathscr{K}_p$ be closed under X-subautomata and homomorphic images and suppose that $\mathscr{K}^* \subseteq \mathscr{K}$. If $\mathscr{K}$ contains a nontrivial strongly connected automaton then a class $\mathscr{K}_0$ is isomorphically α_0-complete for $\mathscr{K}$ if and only if $\mathbf{A} \in \mathbf{ISP}_{1\alpha_0}(\mathscr{K})$ holds for every simple strongly connected automaton $\mathbf{A}$ in $\mathscr{K}$.

Let $\mathscr{G}$ be a nonempty class of (finite) simple groups closed under division. (Recall that G_1 divides G_2 for groups G_1 and G_2, written $G_1 | G_2$, if and only if G_1 is a homomorphic image of a subgroup of G_2.) Denote by $\mathscr{K}(\mathscr{G})$ the class $\mathbf{HSP}_{\alpha_0}(\{\mathbf{A}(G) \mid G \in \mathscr{G}\})$; $\mathscr{K}(\mathscr{G})$ is an α_0^*-variety contained in $\mathscr{K}_p$. It follows from the Krohn—Rhodes Decomposition Theorem that every α_0^*-variety of permutation automata is of the form $\mathscr{K}(\mathscr{G})$ except for the α_0^*-variety consisting of all automata (A, X, δ) such that δ_x is the identity map for each $x \in X$. Moreover, if $\mathscr{G}$ contains a nontrivial simple group then for every permutation automaton $\mathbf{A}$ we have $\mathbf{A} \in \mathscr{K}(\mathscr{G})$ if and only if $G | S(\mathbf{A})$ implies $G \in \mathscr{G}$ for simple groups G. Since $\mathscr{K}(\mathscr{G}) \subseteq \mathscr{K}_p$, also $\mathscr{K}_m(\mathscr{G}) = \mathbf{HSP}_{\alpha_0}(\mathscr{K}(\mathscr{G}) \cup \mathscr{K}_m) \subseteq \mathscr{K}_{pm}$. We obviously have

$$\mathscr{K}_m(\mathscr{G}) = \mathbf{HSP}_{\alpha_0}(\mathscr{K}(\mathscr{G}) \cup \{\mathbf{E}\}) = \mathbf{HSP}_{\alpha_0}(\{\mathbf{A}(G) \mid G \in \mathscr{G}\} \cup \{\mathbf{E}\}).$$

Thus, $\mathscr{K}_m(\mathscr{G})$ is an α_0-variety in $\mathscr{K}_{pm}$, or even, it is an α_0^*-variety.

Corollary. $\mathscr{K}_m(\mathscr{G}) \subseteq \mathbf{ISP}_{\alpha_0}(\mathscr{K})$ if and only if the following hold:

(i) for every simple cone $\mathbf{A} \in \mathscr{K}_m(\mathscr{G})$ we have $\mathbf{A} \in \mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$,

(ii) for every 0-simple cone $\mathbf{A}_0^c \in \mathscr{K}_m(\mathscr{G})$ there is a $\mathbf{B} \in \mathscr{K}_0$ such that $\mathbf{A}_0^c$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{B}$.

Proof. Use Theorem 3.3 and the following fact: every simple strongly connected (permutation) automaton in $\mathscr{K}_m(\mathscr{G})$ is isomorphic to an X-subautomaton of a 0-simple cone $\mathbf{A}_0^c$ in $\mathscr{K}_m(\mathscr{G})$.

Corollary [2]. A class $\mathscr{K}_0$ is isomorphically α_0-complete for $\mathscr{K}_m$ if and only if $\mathbf{E} \in \mathbf{ISP}_{1\alpha_0}(\mathscr{K}_0)$.

Proof. Let $\mathcal{G}$ be the class of trivial groups. We have $\mathcal{K}_m = \mathcal{K}_m(\mathcal{G})$. On the other hand, every cone in $\mathcal{K}_m$ is similar to $\mathbf{E}$. More exactly, if $\mathbf{A} \in \mathcal{K}_m$ is a cone then $\mathbf{A}$ is isomorphic to an α_0-product in $\mathbf{P}_{1\alpha_0}(\{\mathbf{E}\})$.

An automaton $\mathbf{A}=(A, X, \delta)$ is called *commutative* if $\delta(a, xy)=\delta(a, yx)$ for all $a \in A$ and $x, y \in X$, i.e., if $S(\mathbf{A})$ is commutative. Denote by $\mathcal{K}$ the class of all commutative automata; $\mathcal{K}$ is closed under X-subautomata and homomorphic images. Moreover, $\mathcal{K}^* \subseteq \mathcal{K}$ and $\mathcal{K} \subseteq \mathcal{K}_{pm}$. For a prime $p>1$ let $\mathbf{C}_p$ be a fixed automaton of the form $\mathbf{A}(Z_p)^c$, where Z_p is the cyclic group of order p. Every simple commutative automaton is in the class $\mathbf{ISP}_{1\alpha_0}(\{\mathbf{C}_p | p>1$ is a prime$\})$, and every 0-simple commutative cone is in $\mathbf{ISP}_{1\alpha_0}(\{\mathbf{C}_p^c | p>1$ is a prime$\})$.

Corollary [7]. A class $\mathcal{K}_0$ is isomorphically α_0-complete for the class of all commutative automata if and only if the following hold:

(i) $\mathbf{E} \in \mathbf{HSP}_{1\alpha_0}(\mathcal{K}_0)$,

(ii) for every prime $p>1$ there is an $\mathbf{A} \in \mathcal{K}_0$ such that $\mathbf{C}_p^c$ is isomorphic to a subautomaton of an α_0-product of $\mathbf{E}$ with $\mathbf{A}$.

Abstract

Every isomorphically α_0-complete class for a class $\mathcal{K}$ of automata must essentially contain all simple automata belonging to $\mathcal{K}$. In this paper we present some classes $\mathcal{K}$ for which also the converse is true, or isomorphically α_0-complete classes can be characterized by means of automata in $\mathcal{K}$ close to simple automata.

BOLYAI INSTITUTE
A. JÓZSEF UNIVERSITY
ARADI VÉRTANÚK TERE 1
SZEGED, HUNGARY
H—6720

References

[1] EILENBERG, S., Automata, languages, and machines, v. B, Academic Press, New York, 1976.
[2] GÉCSEG, F., On products of abstract automata, Acta Sci. Math, 38 (1976), 21—43.
[3] GÉCSEG, F., Products of automata, Springer, Berlin, 1986.
[4] GINZBURG, A., Algebraic theory of automata, Academic Press, New York, 1968.
[5] HARTMANIS, J. and R. E. STEARNS, Algebraic structure theory of sequential machines, Prentice-Hall, 1966.
[6] IMREH, B., On α_0-products of automata, Acta Cybernet., 3 (1978), 301—307.
[7] IMREH, B., On isomorphic representations of commutative automata with respect to α_0-products, Acta Cybernet., 5 (1980), 21—32.
[8] IMREH, B., On finite nilpotent automata, Acta Cybernet., 5 (1981), 281—293.
[9] IMREH, B., On finite definite automata, Acta Cybernet., 7 (1985), 61—65.
[10] STIFFLER, P., Extension of the fundamental theorem of finite semigroups, Advances in Mathematics, 11 (1973), 159—209.
[11] BRZOZOWSKI, J. A. and F. E. FICH, Languages of $\mathcal{R}$-trivial monoids, J. of Computer and System Sciences, 20 (1980), 32—49.

(Received Nov. 21, 1986)

On metric equivalence of v_i-products

F. GÉCSEG and B. IMREH

In [7] it is shown that the v_3-product is metrically equivalent to the product. Here we strengthen this result by proving that already the v_1-product is metrically equivalent to the general product. It is also obtained that, if a class $\mathscr{K}$ of automata is not metrically complete for the product, then $\mathbf{HSP}_g(\mathscr{K})=\mathbf{HSP}_{v_1}(\mathscr{K})$.

In this paper by an automaton we mean a finite automaton. The only exceptions are varieties of automata; they may contain automata with infinite state-sets. For all notions and notations not defined here, see [1], [7], [8] and [9].

We start with

Lemma 1. If a finite class $\mathscr{K}$ of automata is not metrically complete for the product, then every finitely generated automaton $\mathfrak{A}=(X, A, \delta)$ from $\mathbf{HSPP}_{v_1}(\mathscr{K})$ is in $\mathbf{HSP}_{v_1}(\mathscr{K})$.

Proof. First let us note that the concept of the v_1-product can be generalized in a natural way to products with infinitely many factors, and every automaton in $\mathbf{PP}_{v_1}(\mathscr{K})$ is a v_1-product with possibly infinitely many factors. Thus, take a v_1-product

$$\mathfrak{B} = (X, B, \delta') = \prod (\mathfrak{A}_i \mid i \in I)[X, \varphi, \gamma]$$

with $|X|=m$ and $\mathfrak{A}_i=(X_i, A_i, \delta_i)\in\mathscr{K}$ $(i\in I)$. Let $\{a_1, \ldots, a_n\}$ be a generating set of $\mathfrak{A}$. Suppose that a subautomaton of $\mathfrak{B}$ can be mapped homomorphically onto $\mathfrak{A}$, and let $\mathbf{b}_i$ be a counter image of a_i $(i=1, \ldots, n)$ under this homomorphism. Denote by $\mathfrak{B}'=(X, B', \delta'')$ the subautomaton of $\mathfrak{B}$ generated by $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$. Moreover, set $u=\max\{|A_i| \mid i\in I\}$ and $v=|\mathscr{K}|$. Let $k\geqq 0$ be a fixed integer such that, for arbitrary $\mathfrak{C}=(X_\mathfrak{C}, C, \delta_\mathfrak{C})\in\mathscr{K}$, $c\in C$, $p\in X_\mathfrak{C}^*$ with $|p|\geqq k$ and $x_1, x_2\in X_\mathfrak{C}$, $cpx_1=cpx_2$. (Since $\mathscr{K}$ is not metrically complete, there exists such a k.) We shall show the existence of a v_1-product $\overline{\mathfrak{B}}=(X, \overline{B}, \overline{\delta})$ of automata from $\{\mathfrak{A}_i \mid i\in I\}$ with a number of factors not exceeding vu^{m^t}, where $t=\dfrac{m^{k+1}-1}{m-1}$ if $m>1$, and $t=k+1$ for $m=1$, such that a subautomaton $\overline{\mathfrak{B}}'=(X, \overline{B}', \overline{\delta}')$ of $\overline{\mathfrak{B}}$ is isomorphic to $\mathfrak{B}'$.

Define the binary relation ϱ on I in the following way: $i\equiv j(\varrho)$ $(i, j\in I)$ if and only if $\mathfrak{A}_i=\mathfrak{A}_j$ and $\delta_i(\mathrm{pr}_i(\mathbf{b}_r), \varphi_i(\mathbf{b}_r, p))=\delta_j(\mathrm{pr}_j(\mathbf{b}_r), \varphi_j(\mathbf{b}_r, p))$ hold for arbitrary r $(1\leqq r\leqq n)$ and $p\in X^*$ with $|p|\leqq k$. By the choice of k, $\delta_i(\mathrm{pr}_i(\mathbf{b}_r), \varphi_i(\mathbf{b}_r, q))=\delta_j(\mathrm{pr}_j(\mathbf{b}_r), \varphi_j(\mathbf{b}_r, q))$ is valid for any r $(1\leqq r\leqq n)$ and $q\in X^*$. Moreover, since

t is the number of words over X with length less than or equal to k, we have at most vu^{nt} ϱ-classes. From every ϱ-class take exactly one element, and let $\{i_1, \ldots, i_l\}$ be their set. Form the ν_1-product

$$\overline{\mathfrak{B}} = (X, \bar{B}, \delta) = \prod (\mathfrak{A}_{i_j} | j = 1, \ldots, l)[X, \varphi', \gamma']$$

in the following way:

(i) For every j $(1 \leqq j \leqq l$, $\gamma'(i_j) = \emptyset$ if $\gamma(i_j) = \emptyset$, and $\gamma'(i_j) = \{i_{j_1}\}$ $(j_1 \in \{1, \ldots, l\})$ if $\gamma(i_j) = \{j_2\}$ and $i_{j_1} \equiv j_2 (\varrho)$.

(ii) For every j $(1 \leqq j \leqq l)$ and $x \in X$, $\varphi'_{i_j}(x) = \varphi_{i_j}(x)$ if $\gamma'(i_j) = \emptyset$.

(iii) For every j $(1 \leqq j \leqq l)$, if $\gamma'(i_j) = \{i_{j_1}\}$, then $\varphi'_{i_j}(a, x) = \varphi_{i_j}(a, x)$ $(a \in A_{i_{j_1}}$, $x \in X)$.

Moreover, let $\bar{\mathfrak{b}}_i$ $(i = 1, \ldots, n)$ be those states of $\overline{\mathfrak{B}}$ which, for every $j (= 1, \ldots, l)$, satisfy the equality $\mathrm{pr}_{i_j}(\bar{\mathfrak{b}}_i) = \mathrm{pr}_{i_j}(\mathfrak{b}_i)$. Denote by $\overline{\mathfrak{B}}' = (X, \bar{B}', \delta')$ the subautomaton of $\overline{\mathfrak{B}}$ generated by $\{\bar{\mathfrak{b}}_1, \ldots, \bar{\mathfrak{b}}_n\}$. Moreover, consider the mapping $\psi: B' \to \bar{B}'$ given by $\psi(\mathfrak{b}_i p) = \bar{\mathfrak{b}}_i p$ $(p \in X^*, i = 1, \ldots, n)$. Clearly, ψ is an isomorphism of $\mathfrak{B}'$ onto $\overline{\mathfrak{B}}'$. □

Lemma 2. If a finite class $\mathscr{K}$ of automata is not metrically complete for the product, then the equality $\mathbf{HSPP}_g(\mathscr{K}) = \mathbf{HSPP}_{\nu_1}(\mathscr{K})$ holds.

Proof. Obviously, $\mathbf{HSPP}_{\nu_1}(\mathscr{K}) \subseteqq \mathbf{HSPP}_g(\mathscr{K})$. Thus, it is enough to show $\mathbf{HSPP}_g(\mathscr{K}) \subseteqq \mathbf{HSPP}_{\nu_1}(\mathscr{K})$. This latter inclusion holds if and only if $\mathbf{HSPP}_g(\mathscr{K}) \cap \cap \mathscr{K}_X \subseteqq \mathbf{HSPP}_{\nu_1}(\mathscr{K}) \cap \mathscr{K}_X$ for all input alphabet X, where $\mathscr{K}_X$ is the similarity class of all automata with input alphabet X. Since automata identities have at most two variables, $\mathbf{HSPP}_g(\mathscr{K}) \cap \mathscr{K}_X = \mathbf{HSP}(\{\mathfrak{A}_2\})$, where $\mathfrak{A}_2$ is a free automaton of the variety $\mathbf{HSPP}_g(\mathscr{K}) \cap \mathscr{K}_X$ generated by two elements. Let $\mathfrak{A}_1$ be a free automaton in $\mathbf{HSPP}_g(\mathscr{K}) \cap \mathscr{K}_X$ generated by a single element. One can show that every finitely generated automaton in $\mathbf{HSPP}_g(\mathscr{K}) \cap \mathscr{K}_X$ is in $\mathbf{HSP}_g(\mathscr{K})$, $\mathfrak{A}_2$ can be represented homomorphically by a quasi-direct square of $\mathfrak{A}_1$, or by a quasi-direct product of $\mathfrak{A}_1$ by a two-state discrete automaton with a single input signal depending on the forms of the p-identities holding in $\mathfrak{A}_2$ (see the Theorem in [3] and the proof of Theorem 2.1 from [5]). Since every finitely generated automaton from $\mathbf{HSPP}_{\alpha_0}(\mathscr{K})$ is in $\mathbf{HSP}_{\alpha_0}(\mathscr{K})$, by the Theorem of [3] and Proposition 12 from [4], if a two-state discrete automaton is in $\mathbf{HSPP}_g(\mathscr{K})$ then it is in $\mathbf{HSQ}(\mathscr{K})$, where $\mathbf{Q}$ is the quasi-direct product operator. Therefore, to prove $\mathbf{HSPP}_g(\mathscr{K}) \subseteqq \mathbf{HSPP}_{\nu_1}(\mathscr{K})$ it is sufficient to show that $\mathfrak{A}_1 \in \mathbf{HSPP}_{\nu_1}(\mathscr{K})$ for an arbitrary input alphabet X. By the proof of Theorem 2 of [7], we may suppose that there is a largest positive integer t such that for an automaton $\mathfrak{C} = (X, C, \delta_C)$ in $\mathscr{K}$, a state $c \in C$ and a word $r \in X^*$ with $|r| = t - 1$ the state cr is ambiguous.

Assume that the identity $zp = zq$ $(p, q \in X^*)$ does not hold in $\mathfrak{A}_1$, where z is a variable, $p = x_1 \ldots x_k x_{k+1} \ldots x_m$, $q = x_1 \ldots x_k y_{k+1} \ldots y_n$, and $x_{k+1} \neq y_{k+1}$ if $m, n > k$. If $m, n \leqq t$, or $m < t$ and $n \geqq t$, then by the proof of Theorem 2 in [7], $zp = zq$ is not satisfied by $\mathbf{P}_{\nu_1}(\mathscr{K})$. Thus, we may assume that $m, n \geqq t$.

Since $\mathfrak{A}_1 \in \mathbf{HSPP}_g(\mathscr{K})$, there are an automaton $\mathfrak{A} = (\bar{X}, A, \delta)$ in $\mathscr{K}$, a state $a_0 \in A$ and two words $p' = x'_1 \ldots x'_l x'_{l+1} \ldots x'_m$, $q' = x'_1 \ldots x'_l y'_{l+1} \ldots y'_n$ in $\bar{X}^*$ such that $a_0 p' \neq a_0 q'$, $l \geqq k$ and $x'_{l+1} \neq y'_{l+1}$ if $m, n > l$. We shall suppose that there are no words $\bar{p} = \bar{x}_1 \ldots \bar{x}_r \bar{x}_{r+1} \ldots \bar{x}_m$ and $\bar{q} = \bar{x}_1 \ldots \bar{x}_r \bar{y}_{r+1} \ldots \bar{y}_n$ in $\bar{X}^*$ with $a_0 \bar{p} \neq a_0 \bar{q}$ and $r > l$.

Let $a_i=a_0x'_1\dots x'_i$ $(i=1, \dots, m)$ and

$$b_i=\begin{cases}a_0x'_1\dots x'_i & \text{if } 1\leqq i\leqq l,\\ a_0x'_1\dots x'_iy'_{i+1}\dots y'_i & \text{if } l<i\leqq n.\end{cases}$$

In the sequel we can confine ourselves to the case $m, n>l$. Assume to the contrary, say $m=l$. Consider the v_1-product $\mathfrak{B}=(X, B, \delta')=\mathfrak{A}[X, \varphi, \gamma]$ with $\gamma(1)=\{1\}$,

$$\varphi(b_i, x_{i+1})=x'_{i+1} \quad (i=0, \dots, \min\{m-1, u\}),$$

$$\varphi(b_i, y_{i+1})=y'_{i+1} \quad (i=l, \dots, \min\{n-1, u\})$$

where u is the largest index for which the states $b_0, \dots, b_n$ are pairwise distinct, and φ is given arbitrarily in all other cases. Observe that if $b_i=b_j$ for $0\leqq i<j\leqq n$ then $\delta(b_r, x')=\delta(b_r, y')$ for arbitrary $r\geqq i$ and $x', y'\in X$; otherwise $\mathscr{K}$ would be metrically complete for the product. (This observation will be used silently throughout the paper.) By the construction of $\mathfrak{B}$, $a_0p_{\mathfrak{B}}=a_0p'_{\mathfrak{A}}$ and $a_0q_{\mathfrak{B}}=a_0q'_{\mathfrak{A}}$. Therefore, $a_0p_{\mathfrak{B}}\neq a_0q_{\mathfrak{B}}$.

We say that a_m and b_n *induce disjoint cycles,* if the subautomata generated by a_m and b_n are disjoint. Otherwise they induce the same cycle.

Let us distinguish the following cases.

Case 1. The states a_m and b_n induce disjoint cycles. By our assumptions on p' and q', $\{a_{l+1}, \dots, a_m\}\cap\{b_{l+1}, \dots, b_m\}=\emptyset$. Let u_1 $(0\leqq u_1<m)$ be the largest index such that the elements $a_0, a_1, \dots, a_{u_1}$ are pairwise distinct. The number u_2 $(0\leqq u_2<n)$ has the same meaning for $b_0, b_1, \dots, b_{u_2}$.

Take the v_1-product

$$\mathfrak{B}=(X, B, \delta')=\mathfrak{A}[X, \varphi, \gamma]$$

where

$$\gamma(1)=\{1\},$$

$$\varphi(a_{l-k+i}, x_{i+1})=x'_{l-k+i+1} \quad (0\leqq i\leqq u_1+k-l),$$

$$\varphi(b_{l+i}, y_{k+i+1})=y'_{l+i+1} \quad (0\leqq i\leqq u_2-l),$$

and in all other cases φ is given arbitrarily. Take $\mathbf{b}=(a_{l-k})$. Then $\mathbf{b}p=(a_{l-k}x'_{l-k+1}\dots x'_m\bar{x}^{l-k})$ and $\mathbf{b}q=(a_{l-k}x'_{l-k+1}\dots x'_l y'_{l+1}\dots y'_n\bar{x}^{l-k})$, where $\bar{x}\in\bar{X}$ is arbitrary. (Remember that $m, n\geqq t$.) Therefore, $\mathbf{b}p\neq\mathbf{b}q$.

Case 2. The states a_m and b_n induce the same cycle, i.e., in the intersection of the subautomata generated by a_m and b_n there is a cycle C of length w. We distinguish some subcases.

Case 2.1. $m\not\equiv n \pmod w$. Then $w>1$. Take an arbitrary v_1-product $\mathfrak{B}=(X, A, \delta')$ of $\mathfrak{A}$ with a single factor. In $\mathfrak{B}$, for any $c\in C$, we have $cp\neq cq$.

Case 2.2. $m\equiv n \pmod w$. Some further subcases are needed.

Case 2.2.1. $a_m, b_n\in C$ or $m=n$.

If $\{a_{l+1}, \dots, a_m\}\cap\{b_{l+1}, \dots, b_n\}=\emptyset$, then let u_1 $(0\leqq u_1<m)$ be the largest

index such that the elements $a_0, a_1, \ldots, a_{u_1}$ are pairwise distinct. The number u_2 $(0 \leqq u_2 < n)$ has the same meaning for $b_0, b_1, \ldots, b_{u_2}$.

Take the ν_1-product

$$\mathfrak{B} = (X, B, \delta') = (\underbrace{\mathfrak{A} \times \ldots \times \mathfrak{A}}_{l-k+1 \text{ times}})[X, \varphi, \gamma]$$

where

$$\gamma(1) = \{1\}; \quad \gamma(i) = \{i-1\} \quad (i = 2, \ldots, l-k+1),$$

$$\varphi_1(a_{l-k+i}, x_{i+1}) = x'_{l-k+i+1} \quad (0 \leqq i \leqq u_1 + k - l),$$

$$\varphi_1(b_{l+i}, y_{k+i+1}) = y'_{l+i+1} \quad (0 \leqq i \leqq u_2 - l),$$

and for every $j (= 2, \ldots, l-k+1)$,

$$\varphi_j(a_{l-k-(j-2)+i}, x_{i+1}) = x'_{l-k-(j-2)+i} \quad (0 \leqq i \leqq u_1 - l + k + j - 2),$$

$$\varphi_j(b_{l-(j-2)+i}, y_{k+i+1}) = \begin{cases} x'_{l-(j-2)+i} & \text{if} \quad 0 \leqq i \leqq j-2 \\ y'_{l-(j-2)+i} & \text{if} \quad j-2 < i \leqq u_2 - l + j - 2, \end{cases}$$

and in all other cases φ is given arbitrarily in accordance with the definition of the ν_1-product.

Take $\mathbf{b} = (a_{l-k}, a_{l-k-1}, \ldots, a_0)$. Then $\mathbf{b}p = (a_{l-k}p_0, a_{l-k-1}p_1, \ldots, a_0 p_{l-k})$ and $\mathbf{b}q = (a_{l-k}q_0, a_{l-k-1}q_1, \ldots, a_0 q_{l-k})$ where, for every $j (= 0, \ldots, l-k)$, $p_j = x'_{l-k-j+1} \ldots x'_m \bar{x}^{l-k-j}$ and $q_j = x'_{l-k-j+1} \ldots x'_l y'_{l+1} \ldots y'_n \bar{x}^{l-k-j}$, and $\bar{x} \in \bar{X}$ is arbitrary. Thus $p_{l-k} = p'$ and $q_{l-k} = q'$, implying $\mathbf{b}p \neq \mathbf{b}q$.

If $\{a_{l+1}, \ldots, a_m\} \cap \{b_{l+1}, \ldots, b_n\} \neq \emptyset$, then let r $(l+1 \leqq r \leqq m)$ be the least index for which there is a b_j with $a_r = b_j$. Moreover, let s $(l+1 \leqq s \leqq n)$ be the least index such that $b_s = a_r$. Then $r \neq s$, since in the opposite case $\bar{p} = x'_1 \ldots x'_r x'_{r+1} \ldots x'_m$ and $\bar{q} = x'_1 \ldots x'_r y'_{r+1} \ldots y'_n$ would contradict the choice of p' and q'. Assume that $r < s$. Let u $(0 \leqq u < m)$ be the largest index for which the states $a_0, \ldots, a_u$ are pairwise distinct. Take the ν_1-product

$$\mathfrak{B} = (X, B, \delta') = (\underbrace{\mathfrak{A} \times \ldots \times \mathfrak{A}}_{l-k+1 \text{ times}})[X, \varphi, \gamma]$$

where

$$\gamma(1) = \{1\}; \quad \gamma(i) = \{i-1\} \quad (2 \leqq i \leqq l-k+1),$$

$$\varphi_1(a_{l-k+i}, x_{i+1}) = x'_{l-k+i+1} \quad (0 \leqq i \leqq u + k - l),$$

$$\varphi_1(b_{l+i}, y_{k+i+1}) = y'_{l+i+1} \quad (0 \leqq i \leqq r - l),$$

and for every $j (= 2, \ldots, l-k+1)$,

$$\varphi_j(a_{l-k-(j-2)+i}, x_{i+1}) = x'_{l-k-(j-2)+i} \quad (0 \leqq i \leqq u - l + k + j - 2),$$

$$\varphi_j(b_{l-(j-2)+i}, y_{k+i+1}) = \begin{cases} x'_{l-(j-2)+i} & \text{if} \quad 0 \leqq i \leqq j-2, \\ y'_{l-(j-2)+i} & \text{if} \quad j-2 < i \leqq r - l + j - 2, \end{cases}$$

and in all other cases φ is given arbitrarily. Take the state

$$\mathbf{b} = (a_{l-k}, a_{l-k-1}, \ldots, a_0) \in B.$$

Then $\mathbf{b}p=(b'_1, \ldots, b'_{l-k}, a_0p')$ and $\mathbf{b}q=(b''_1, \ldots, b''_{l-k}, a_0x'_1\ldots x'_ly'_{l+1}\ldots y'_r\bar{q})$ where $\bar{q}\in\bar{X}^*$ satisfies the equality $|\bar{q}|=n-r$. One can easily check that $a_0p'\neq a_0x'_1\ldots x'_ly'_{l+1}\ldots y'_r\bar{q}$. Indeed, in the opposite case let $\bar{q}'$ be the initial segment of $\bar{q}$ with length $m-r$ if $m\leqq n$, and otherwise let $\bar{q}'=\bar{q}\bar{\bar{q}}$, where $\bar{\bar{q}}\in\bar{X}^*$ is arbitrary with $|\bar{q}\bar{\bar{q}}|=m-r$. From our assumptions it follows that $a_0x'_1\ldots x'_ly'_{l+1}\ldots y'_r\bar{q}'\neq a_0q'$. Therefore, by $r>l$, the pair $x'_1\ldots x'_ly'_{l+1}\ldots y'_r\bar{q}'$, q' contradicts the choice of p' and q'.

Case 2.2.2. $m\neq n$ and at least one of a_m and b_n is not in C.

Case 2.2.2.1. None of a_m and b_n is in C and $m<n$. Then the states $b_0, \ldots, b_n$ are pairwise distinct. Take the v_1-product

$$\mathfrak{B} = (X, B, \delta') = \mathfrak{A}[X, \varphi, \gamma]$$

where

$$\gamma(1) = \{1\},$$

$$\varphi(b_i, x_{i+1}) = \begin{cases} x'_{i+1} & \text{if } 0\leqq i<l, \\ y'_{i+1} & \text{if } l\leqq i<m, \end{cases}$$

$$\varphi(b_i, y_{i+1}) = \begin{cases} x'_{i+1} & \text{if } k\leqq i<l, \\ y'_{i+1} & \text{if } l\leqq i<n, \end{cases}$$

and φ is given arbitrarily in all other cases. Taking $\mathbf{b}=(b_0)$ we obtain $\mathbf{b}p=(b_m)$ and $\mathbf{b}q=(b_n)$.

Case 2.2.2.2. $a_m\notin C$, $b_n\in C$ and $n>m$; or $a_m\notin C$, $b_n\in C$ and $n<m$. The states $a_0, a_1, \ldots, a_m$ are pairwise distinct. Take the v_1-product

$$\mathfrak{B} = (X, B, \delta') = \mathfrak{A}[X, \varphi, \gamma]$$

where

$$\gamma(1) = \{1\},$$

$$\varphi(a_i, x_{i+1}) = x'_{i+1} \quad (0\leqq i<m),$$

$$\varphi(a_l, y_{i+1}) = x'_{i+1} \quad (k\leqq i<\min\{m, n\}),$$

and φ is given arbitrarily in the remaining cases. Let $\mathbf{b}=(a_0)$. If $n>m$, then $\mathbf{b}p=(a_m)$ and $\mathbf{b}q=(a_m\bar{x}^{n-m})$, where $\bar{x}\in\bar{X}$ is arbitrary. Obviously, $a_m\neq a_m\bar{x}^{n-m}$, since in the opposite case $a_m\in C$. If $n<m$, then $\mathbf{b}p=(a_m)$ and $\mathbf{b}q=(a_n)$. □

Remark. Let $\mathcal{K}$ be an arbitrary class of automata. In [3] it is shown that if an identity does not hold in an infinite product of automata from $\mathcal{K}$, then there is a finite product of automata from $\mathcal{K}$ which does not satisfy the given identity either. (See also [2], where this result is generalized to automata with infinite input alphabets.) Moreover, by Theorem 1 of [7], the v_1-product is equivalent to the product as regards metric completeness. Therefore, if $\mathcal{K}$ is metrically complete for the product, then none of the nontrivial p-identities holds in $\mathbf{HSPP}_{v_1}(\mathcal{K})$. Thus, using Lemma 2, we obtain that $\mathbf{HSPP}_g(\mathcal{K})=\mathbf{HSPP}_{v_1}(\mathcal{K})$ for arbitrary class of automata. However, Lemma 2 will be sufficient to prove our main result.

By Lemmas 1 and 2, we obtain

Corollary 3. If a class $\mathcal{K}$ of automata is not metrically complete for the product, then $\mathbf{HSP}_g(\mathcal{K})=\mathbf{HSP}_{v_1}(\mathcal{K})$.

Proof. The inclusion $\mathbf{HSP}_{\nu_1}(\mathscr{K}) \subseteq \mathbf{HSP}_g(\mathscr{K})$ is obvious. If $\mathfrak{A} \in \mathbf{HSP}_g(\mathscr{K})$, then there exists a finite subset $\mathscr{K}$ such that $\mathfrak{A} \in \mathbf{HSP}_g(\overline{\mathscr{K}})$. Therefore, by Lemmas 1 and 2, $\mathfrak{A} \in \mathbf{HSP}_{\nu_1}(\overline{\mathscr{K}})$. □

Let us note that by the proof of the Theorem in [6], $\mathbf{HSP}_{\alpha_0}(\mathscr{K}) = \mathbf{HSP}_g(\mathscr{K})$ if $\mathscr{K}$ is not metrically complete for the product. Thus, for such classes $\mathscr{K}$, the equality $\mathbf{HSP}_{\alpha_0}(\mathscr{K}) = \mathbf{HSP}_{\nu_1}(\mathscr{K})$ holds, too.

Now we are ready to state and prove the main result of the paper.

Theorem 4. The ν_1-product is metrically equivalent to the general product.

Proof. Let $\mathscr{K}$ be an arbitrary class of automata. If $\mathscr{K}$ is metrically complete for the product, then by Theorem 1 in [7], $\mathscr{K}$ is metrically complete with respect to the ν_1-product. If $\mathscr{K}$ is not metrically complete, then $\mathbf{HSP}_g(\mathscr{K}) = \mathbf{HSP}_{\nu_1}(\mathscr{K})$, as it is stated in Corollary 3. □

DEPT. OF COMPUTER SCIENCE
A. JÓZSEF UNIVERSITY
ARADI VÉRTANÚK TERE 1
SZEGED, HUNGARY
H—6720

References

[1] DÖMÖSI, P., B. IMREH, On ν_t-products of automata, Acta Cybernet., v. VII. 2, 1983, pp. 149—162.
[2] ÉSIK, Z., On identities preserved by general products of algebras, Acta Cybernet., v. VI. 3, 1983, pp. 285—289.
[3] ÉSIK, Z., F. GÉCSEG, General products and equational classes of automata, Acta Cybernet., VI. 3, 1983, pp. 281—284.
[4] ÉSIK, Z., F. GÉCSEG, On a representation of tree automata, Theorical Computer Science, to appear.
[5] ÉSIK, Z., F. GÉCSEG, A decidability result for homomorphic representation of automata by α_0-products, Acta Mathematica Hungarica, submitted for publication.
[6] ÉSIK, Z., GY. HORVÁTH, The α_2-product is homomorphically general, Papers on Automata Theory V, no. DM 83—3, pp. 49—62, Karl Marx University of Economics, Department of Mathematics, Budapest (1983).
[7] GÉCSEG, F., Metric representations by ν_t-products, Acta Cybernet., v. VII. 2, 1985, pp. 203—209.
[8] GÉCSEG, F., Products of automata, Springer-Verlag, Berlin—Heidelberg—New York—Tokyo, 1986.
[9] GÉCSEG, F., B. IMREH, A comparison of α_t-products and ν_t-products, Foundations of Control Engineering, submitted for publication.

(Received Sept. 12, 1986)

On α_i-product of tree automata

F. GÉCSEG and B. IMREH

In the theory of finite automata it is a central problem to represent a given automaton by composition of — possibly simpler — automata. The composition of tree automata has received little attention. Namely, the cascade product of tree automata was studied in [4] and the work [5] contains the investigation of the general product of tree automata (see also [1]). In this paper generalizing the notion of α_i-product (cf. [2]), we introduce the α_i-product of tree automata, and using the idea in [3] give necessary and sufficient conditions for a system of tree automata to be isomorphically complete with respect to the α_i-product. From the characterizations of complete systems we obtain the α_i-products constitute a proper hierarchy.

1. Definitions

By a *set of operational symbols* we mean the nonempty union $\Sigma=\Sigma_0\cup\Sigma_1\cup\ldots$ of pairwise disjoint sets of symbols, and for any nonnegative integer m, Σ_m is called the *set of m-ary operational symbols*. It is said that the *rank* or *arity* of a symbol $\sigma\in\Sigma$ is m if $\sigma\in\Sigma_m$. Now let a set Σ of operational symbols be given. A set R of nonnegative integers is called the *rank-type of* Σ if for any m, $\Sigma_m\neq\emptyset$ if and only if $m\in R$. Next we shall work always under a fixed rank-type R.

Let Σ be a set of operational symbols with rank-type R. Then by a *Σ-algebra* $\mathscr{A}$ we mean a pair consisting of a nonempty set A (of elements of $\mathscr{A}$) and a mapping that assigns to every operational symbol $\sigma\in\Sigma$ an m-ary operation $\sigma^{\mathscr{A}}\colon A^m\to A$, where the arity of σ is m. The operation $\sigma^{\mathscr{A}}$ is called the *realization of σ in $\mathscr{A}$*. The mapping $\sigma\to\sigma^{\mathscr{A}}$ will not be mentioned explicitly, but we write $\mathscr{A}=(A,\Sigma)$. The Σ-algebra $\mathscr{A}$ is *finite* if A is finite, and it is of *finite type* if Σ is finite. By a *tree automaton* we mean a finite algebra of finite type. We say that the rank-type of a tree automaton $\mathscr{A}=(A,\Sigma)$ is R if the rank-type of Σ is R. Let us denote by $\mathfrak{A}_R$ the class of all tree automata with rank-type R.

Now let i be a fixed nonnegative integer, and let

$$\mathscr{A}=(A,\Sigma)\in\mathfrak{A}_R,\quad \mathscr{A}_j=(A_j,\Sigma^j)\in\mathfrak{A}_R\quad (j=1,\ldots,k).$$

Moreover, take a family ψ of mappings

$$\psi_{mj}\colon (A_1\times\ldots\times A_k)^m\times\Sigma_m\to\Sigma^j_m,\quad m\in R,\ 1\leqq j\leqq k.$$

It is said that the tree automaton $\mathscr{A}$ is the α_i-product of $\mathscr{A}_j$ $(j=1, \ldots, k)$ with respect to ψ if the following conditions are satisfied:

(1) $A = \prod_{t=1}^{k} A_t$,

(2) for any $m \in R$, $j \in \{1, \ldots, k\}$,

$$((a_{11}, \ldots, a_{1k}), \ldots, (a_{m1}, \ldots, a_{mk})) \in (A_1 \times \ldots \times A_k)^m$$

the mapping ψ_{mj} is independent of elements a_{rs} $(1 \leqq r \leqq m, j+i \leqq s)$,

(3) for any $m \in R$, $\sigma \in \Sigma_m$, $((a_{11}, \ldots, a_{1k}), \ldots, (a_{m1}, \ldots, a_{mk})) \in (A_1 \times \ldots \times A_k)^m$,

$$\sigma^{\mathscr{A}}((a_{11}, \ldots, a_{1k}), \ldots, (a_{m1}, \ldots, a_{mk})) = (\sigma_1^{\mathscr{A}_1}(a_{11}, \ldots, a_{m1}), \ldots, \sigma_k^{\mathscr{A}_k}(a_{1k}, \ldots, a_{mk})),$$

where

$$\sigma_j = \psi_{mj}((a_{11}, \ldots, a_{1k}), \ldots, (a_{m1}, \ldots, a_{mk}), \sigma) \quad (j = 1, \ldots, k).$$

For the above product we shall use the notation $\prod_{j=1}^{k} \mathscr{A}_j(\Sigma, \psi)$ and sometimes we shall write only those variables of ψ_{mj} on which ψ_{mj} depends.

Finally, we shall denote by $[\sqrt[t]{n}]$ the largest integer less than or equal to $\sqrt[t]{n}$.

2. Completeness

Let i be a fixed nonnegative integer and $\mathfrak{B} \subseteq \mathfrak{A}_R$. $\mathfrak{B}$ is called *isomorphically complete* for $\mathfrak{A}_R$ with respect to the α_i-product if any tree automaton from $\mathfrak{A}_R$ can be embedded isomorphically into an α_i-product of tree automata from $\mathfrak{B}$. Furthermore, $\mathfrak{B}$ is called *minimal isomorphically complete* system if $\mathfrak{B}$ is isomorphically complete and for arbitrary $\mathscr{A} \in \mathfrak{B}$, $\mathfrak{B} \setminus \{\mathscr{A}\}$ is not isomorphically complete.

For any natural number $n>0$ let us denote by $\mathscr{B}_n = (\{0, \ldots, n-1\}, \theta^n)$ the tree automaton where for every m-ary operation ϱ: $\{0, \ldots, n-1\}^m \to \{0, \ldots, n-1\}$ there exists exactly one $\sigma \in \theta_m^n$ with $\sigma^{\mathscr{B}_n} = \varrho$ provided that $m \in R$.

The following statement is obvious.

Lemma. *If* $\mathscr{A}_j \in \mathfrak{A}_R$ $(j=1, 2, 3)$ and $\mathscr{A}_j$ can be embedded isomorphically into and α_i-product of $\mathscr{A}_{j+1}$ with a single factor $(j=1, 2)$ then $\mathscr{A}_1$ can be embedded isomorphically into an α_i-product of $\mathscr{A}_3$ with a single factor.

First we consider the special case $R=\{0\}$. Then the following statement is obvious.

Theorem 1. $\mathfrak{B} \subseteq \mathfrak{A}_R$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product if and only if there exists an $\mathscr{A} \in \mathfrak{B}$ such that $\mathscr{B}_2$ can be embedded isomorphically into an α_i-product of $\mathscr{A}$ with a single factor.

Now let us suppose $R \neq \{0\}$. Then the results of completeness is based on the following Theorem.

Theorem 2. If the tree automaton $\mathscr{B}_n$ $(n>1)$ can be embedded isomorphically

into an α_i-product $\prod_{j=1}^{k} \mathscr{A}_j(\theta^n, \psi)$ of the tree automata $\mathscr{A}_j \in \mathfrak{A}_R$ $(j=1, \ldots, k)$ then $\mathscr{B}_{[i^*\sqrt{n}]}$ can be embedded isomorphically into an α_i-product of $\mathscr{A}_j$ with a single factor for some $j \in \{1, \ldots, k\}$, where $i^*=i$ if $i>0$ and $i^*=1$ else.

Proof. If $k=1$ then the statement is obvious. Now let $k>1$. Assume that $\mathscr{B}_n$ can be embedded isomorphically into the α_i-procut $\mathscr{A}=\prod_{j=1}^{k} \mathscr{A}_j(\theta^n, \psi)$ and let μ denote a suitable isomorphism. Let $\mu(t)=(a_{t1}, \ldots, a_{tk})$ $(t=0, \ldots, n-1)$. We may suppose that there exist natural numbers $u \neq v$ $(0 \leqq u, v \leqq n-1)$ such that $a_{u1} \neq a_{v1}$ since otherwise $\mathscr{B}_n$ can be embedded isomorphically into an α_i-product of $\mathscr{A}_j$ $(j=2, \ldots, k)$. Now assume that there exist natural numbers $p \neq q$ $(0 \leqq p, q \leqq n-1)$ with $a_{ps}=a_{qs}$ $(s=1, \ldots, i^*)$. For any t $(0 \leqq t \leqq n-1)$ let us denote by $\sigma_{pt}^{\mathscr{B}_n}$ the m-ary operation of $\mathscr{B}_n$ for which $\sigma_{pt}^{\mathscr{B}_n}(0, \ldots, 0, p)=t$ and $\sigma_{pt}^{\mathscr{B}_n}(0, \ldots, 0, q)=q$, for some $m \in R$. Such operations exist since $R \neq \{0\}$. Then for any $t \in \{0, \ldots, n-1\}$

$$(a_{t1}, \ldots, a_{tk}) = \mu(t) = \mu(\sigma_{pt}^{\mathscr{B}_n}(0, \ldots, 0, p)) = \sigma_{pt}^{\mathscr{A}}(\mu(0), \ldots, \mu(0), \mu(p)) =$$

$$= (\sigma_1^{\mathscr{A}_1}(a_{01}, \ldots, a_{01}, a_{p1}), \sigma_2^{\mathscr{A}_2}(a_{02}, \ldots, a_{02}, a_{p2}), \ldots, \sigma_k^{\mathscr{A}_k}(a_{0k}, \ldots, a_{0k}, a_{pk}))$$

holds, and so $a_{t1}=\sigma_1^{\mathscr{A}_1}(a_{01}, \ldots, a_{01}, a_{p1})$ where

$$\sigma_1 = \psi_{m1}((a_{01}, \ldots, a_{0k}), \ldots, (a_{01}, \ldots, a_{0k}), (a_{p1}, \ldots, a_{pk}), \sigma_{pt}) =$$

$$= \psi_{m1}(a_{01}, \ldots, a_{0i^*}, a_{p1}, \ldots, a_{pi^*}, \sigma_{pt}) \quad \text{if} \quad i>0$$

and $\sigma_1=\psi_{m1}(\sigma_{pt})$ if $i=0$. In the same way we obtain the equality

$$a_{q1} = \bar{\sigma}_1^{\mathscr{A}_1}(a_{01}, \ldots, a_{01}, a_{q1})$$

where

$$\bar{\sigma}_1 = \psi_{m1}(a_{01}, \ldots, a_{0i^*}, a_{q1}, \ldots, a_{qi^*}, \sigma_{pt}) \quad \text{if} \quad i>0$$

and

$$\bar{\sigma}_1 = \psi_{m1}(\sigma_{pt}) \quad \text{if} \quad i=0.$$

Since $a_{ps}=a_{qs}$ $(s=1, \ldots, i^*)$ we obtain that $\sigma_1=\bar{\sigma}_1$ which implies the equality $a_{t1}=a_{q1}$ for any $t \in \{0, \ldots, n-1\}$. This contradicts our assumption $a_{u1} \neq a_{v1}$, therefore the elements $(a_{t1}, \ldots, a_{ti^*})$ $(0 \leqq t \leqq n-1)$ are pairwise different. Now we shall show that in this case $\mathscr{B}_n$ can be embedded isomorphically into an α_i-product $\bar{\mathscr{A}}=\prod_{j=1}^{i^*} \mathscr{A}_j(\theta^n, \varphi)$. Indeed, let us define the family φ of mappings as follows: for any $m \in R$, $j \in \{1, \ldots, i^*\}$, $((a_1^1, \ldots, a_1^{i^*}), \ldots, (a_m^1, \ldots, a_m^{i^*})) \in \prod_{j=1}^{i^*} A_j$, $\sigma \in \theta^n$ elements

(1) if $i>0$ then

$$\varphi_{mj}((a_1^1, \ldots, a_1^{i^*}), \ldots, (a_m^1, \ldots, a_m^{i^*}), \sigma) = \begin{cases} \psi_{mj}((a_{u_1 1}, \ldots, a_{u_1 k}), \ldots, (a_{u_m 1}, \ldots, a_{u_m k}), \sigma) \\ \text{if there exist } u_1, \ldots, u_m \in \{0, \ldots, n-1\} \\ \text{such that } a_s^t = a_{u_s t} (t = 1, \ldots, i^*, s=1, \ldots, m), \\ \text{arbitrary operational symbol from} \\ \Sigma_m^j \text{ otherwise,} \end{cases}$$

(2) if $i=0$ then $\varphi_{mj}(\sigma)=\psi_{mj}(\sigma)$.

It is clear that φ_{mj} is well defined. On the other hand, it is easy to see that the mapping $v(t)=(a_{t1}, \ldots, a_{ti^*})$ $(t=0, \ldots, n-1)$ is an isomorphism of $\mathscr{B}_n$ into $\mathscr{A}$. Using this isomorphism v we prove that $\mathscr{B}_{[\sqrt[i^*]{n}]}$ can be embedded isomorphically into an α_i-product of $\mathscr{A}_j$ with a single factor for some $j\in\{1, \ldots, i^*\}$. If $i=0$ or $i=1$ then this statement obviously holds. Now assume that $i>1$. Since the elements $(a_{t1}, \ldots, a_{ti^*})$ $(t=0, \ldots, n-1)$ are pairwise different, there exists an $s\in\{1, \ldots, i^*\}$ such that the number of pairwise different elements among $a_{0s}, a_{1s}, \ldots, a_{n-1s}$ is greater than or equal to $v=[\sqrt[i^*]{n}]$. Without loos of generality we may assume that $a_{0s}, \ldots, a_{v-1s}$ are pairwise different elements of $\mathscr{A}_s$. For any $m\in R$, $\sigma\in\theta_m^v$ let us denote by $\bar{\sigma}$ an operational symbol from θ_m^n for which $\sigma^{\mathscr{B}_n}|_{\{0, \ldots, v-1\}^m}=\sigma^{\mathscr{B}_v}$. Now let us define the α_i-product $\mathscr{A}_s(\theta^v, \bar{\varphi})$ as follows: for any $m\in R$, $\sigma\in\theta_m^v$, $(a_{u_1s}, \ldots, a_{u_ms})\in A_s^m$

$$\bar{\varphi}_m(a_{u_1s}, \ldots, a_{u_ms}, \sigma)=\begin{cases}\varphi_{ms}((a_{u_11}, \ldots, a_{u_1i^*}), \ldots, (a_{u_m1}, \ldots, a_{u_mi^*}), \bar{\sigma}) \text{ if}\\ 0\leqq u_t\leqq v-1\ (t=1, \ldots, m),\\ \text{arbitrary operational symbol from } \Sigma_m^s \text{ otherwise.}\end{cases}$$

It can be easily see that the correspondence v': $t\to a_{ts}$ $(t=0, \ldots, v-1)$ is an isomorphism of $\mathscr{B}_v$ into $\mathscr{A}_s(\theta^v, \bar{\varphi})$, which completes the proof of Theorem 2.

Theorem 3. $\mathfrak{B}\subseteq\mathfrak{A}_R$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_0-product if and only if for any natural number $n>1$ there exists an $\mathscr{A}\in\mathfrak{B}$ such that $\mathscr{B}_n$ can be embedded isomorphically into an α_0-product of $\mathscr{A}$ with a single factor.

Proof. The necessity follows from Theorem 2. To prove the sufficiency let us observe that any tree automaton $\mathscr{A}\in\mathfrak{A}_R$ with $|A|=n$ can be embedded isomorphically into an α_0-product of $\mathscr{B}_n$ with a single factor. From this fact, by our Lemma, we obtain the completeness of $\mathfrak{B}$.

Now let $i>0$ be a fixed nonnegative integer. Then in a similar way as above we obtain the following result.

Theorem 4. $\mathfrak{B}\subseteq\mathfrak{A}_R$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product if and only if for any natural number $n>1$ there exists an $\mathscr{A}\in\mathfrak{B}$ such that $\mathscr{B}_n$ can be embedded isomorphically into an α_i-product of $\mathscr{A}$ with a single factor.

Since an α_i-product with a single factor is an α_1-product with a single factor, by Theorem 4, we get the next corollary.

Corollary 1. $\mathfrak{B}\subseteq\mathfrak{A}_R$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_1-product if and only if $\mathfrak{B}$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product.

Now let i be a nonnegative integer. Then we have the following result for the minimal isomorphically complete systems in the case $R\neq\{0\}$.

Theorem 5. There exists no system $\mathfrak{B}\subseteq\mathfrak{A}_R$ which is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product and minimal.

Proof. Let $\mathfrak{B} \subseteq \mathfrak{A}_R$ be isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product. Moreover, let $\mathscr{A} \in \mathfrak{B}$ with $|A| = n$. It is obvious that $\mathscr{A}$ can be embedded isomorphically into an α_i-product of $\mathscr{B}_s$ with a single factor if $s \geqq n$. Take a natural number $s > n$. By Theorem 3 and Theorem 4, there exists an $\bar{\mathscr{A}} \in \mathfrak{B}$ such that $\mathscr{B}_s$ can be embedded isomorphically into an α_i-product of $\bar{\mathscr{A}}$ with a single factor. Therefore, by our Lemma, $\mathscr{A}$ can be embedded isomorphically into an α_i-product of $\bar{\mathscr{A}}$ with a single factor. From this it follows that $\mathfrak{B} \setminus \{\mathscr{A}\}$ is isomorphically complete for $\mathfrak{A}_R$ with respect to the α_i-product, showing that $\mathfrak{B}$ is not minimal.

3. The hierarchy of α_i-products

Let $R \neq \{0\}$ be a fixed rank-type. Take a nonempty set $M \subseteq \mathfrak{A}_R$, and let i be an arbitrary nonnegative integer. Let $\alpha_i(M)$ denote the class of all tree automata from $\mathfrak{A}_R$ which can be embedded isomorphically into an α_i-product of tree automata from M. It is said that the α_i-product is *isomorphically more general* than the α_j-product if for any set $M \subseteq \mathfrak{A}_R$ the relation $\alpha_j(M) \subseteq \alpha_i(M)$ holds and there exists at least one set $\bar{M} \subseteq \mathfrak{A}_R$ such that $\alpha_j(\bar{M})$ is a proper subclass of $\alpha_i(\bar{M})$. This notion was introduced in [2].

As far as the hierarchy of the α_i-products is concerned, we have the following Theorem.

Theorem 6. For any i, j $(i, j \in \{0, 1, \ldots\})$ the α_i-product is isomorphically more general than the α_j product if $j < i$.

Proof. We shall prove that the α_1-product is isomorphically more general than the α_0-product and the α_{i+1}-product is isomorphically more general than the α_i-product if $i \geqq 1$.

First let $M = \{\mathscr{A}_2\}$, where $\mathscr{A}_2 = (\{1, 2\}, \bigcup_{m \in R} \{\sigma_{m1}, \sigma_{m2}\})$ and the operations of $\mathscr{A}_2$ are defined as follows: for any $0 \neq m$, $m \in R$, $(a_1, \ldots, a_m) \in \{1, 2\}^m$

$$\sigma_{m1}^{\mathscr{A}_2}(a_1, \ldots, a_m) = \begin{cases} 1 & \text{if} \quad a_m = 2, \\ 2 & \text{if} \quad a_m = 1, \end{cases}$$

$$\sigma_{m2}^{\mathscr{A}_2}(a_1, \ldots, a_m) = a_m,$$

and $\sigma_{01}^{\mathscr{A}_2} = 1$, $\sigma_{02}^{\mathscr{A}_2} = 2$ if $0 \in R$.

Now let us denote by $\mathscr{A}_3 = (\{1, 2, 3\}, \Sigma')$ the tree automaton where for any $0 \neq m \in R$ $\sigma \in \Sigma'_m$, $(a_1, \ldots, a_m) \in \{1, 2, 3\}^m$

$$\sigma^{\mathscr{A}_3}(a_1, \ldots, a_m) = \begin{cases} a_m + 1 & \text{if} \quad a_m < 3, \\ 3 & \text{if} \quad a_m = 3, \end{cases}$$

and $\bar{\sigma}^{\mathscr{A}_3} = 1$ if $0 \in R$ and $\bar{\sigma} \in \Sigma'_0$.

It is easy to see that $\mathscr{A}_3 \notin \alpha_0(M)$ and $\mathscr{A}_3 \in \alpha_1(M)$ which yields the required inclusion $\alpha_0(M) \subset \alpha_1(M)$.

Now let $i \geqq 1$ and $M=\{\mathscr{B}_2\}$. Then, by the proof of Theorem 2, we obtain that $\mathscr{B}_{2^{i+1}} \notin \alpha_i(M)$. On the other hand, we shall show that $\mathscr{B}_{2^{i+1}} \in \alpha_{i+1}(M)$ which yields the required inclusion $\alpha_i(M) \subset \alpha_{i+1}(M)$. To prove the above statement it is enough to show that $\mathscr{B}_{2^i} \in \alpha_i(M)$ if $i > 1$. Indeed, let us take the α_i-product $\mathscr{A}=\prod_{j=1}^{i} \mathscr{B}_2(\theta^{2^i}, \psi)$ where the family ψ of mappings is defined as follows: for any $0 \neq m$, $\sigma \in \theta_m^{2^i}$,

$$((a_{11}, \ldots, a_{1i}), \ldots, (a_{m1}, \ldots, a_{mi})) \in (\{0, 1\}^i)^m$$

if

$$\sigma^{\mathscr{B}_{2^i}}\Big(\sum_{t=1}^{i} a_{1t}2^{i-t}, \ldots, \sum_{t=1}^{i} a_{mt}2^{i-t}\Big) = w = \sum_{t=1}^{i} a_{wt}2^{i-t} \quad \text{and} \quad \bar{\sigma}^{\mathscr{B}_2}(a_{1j}, \ldots, a_{mj}) = a_{wj}$$

then

$$\psi_{mj}((a_{11}, \ldots, a_{1i}), \ldots, (a_{m1}, \ldots, a_{mi}), \sigma) = \bar{\sigma}.$$

In the case $\sigma \in \theta_0^{2^i}$ if $\sigma^{\mathscr{B}_{2^i}} = \sum_{t=1}^{i} a_{vt} \cdot 2^{i-t}$ and $\bar{\sigma}^{\mathscr{B}_2} = a_{vj}$ then $\psi_{mj}(\sigma) = \bar{\sigma}$.

It is easy to see that $\mathscr{B}_{2^i}$ can be embedded isomorphically into $\mathscr{A}$ under the isomorphism μ defined as follows: if $w=\sum_{t=1}^{i} a_t 2^{i-t}$ then $\mu(w)=(a_1, \ldots, a_i)$ $(w=0, \ldots, 2^i-1)$.

4. A decidability result

In this section we show that it is decidable if an algebra can be represented isomorphically by an α_i-product of algebras from a given finite set.

Theorem 7. For any nonnegative integer i, $\mathscr{A} \in \mathfrak{A}_R$ and finite set $M \subseteqq \mathfrak{A}_R$ it can be decided whether or not $\mathscr{A} \in \alpha_i(M)$.

Proof. Let us suppose that $\mathscr{A}$ with $A=\{a_1, \ldots, a_k\}$ can be embedded isomorphically into an α_i-product $\mathscr{B}=\prod_{j=1}^{s} \mathscr{A}_j(\Sigma, \varphi)$ of tree automata from M. Let $V=\max\{|A_i|: \mathscr{A}_i \in M\}$ and let $(a_{u1}, \ldots, a_{us})$ denote the image of a_u under a suitable isomorphism μ $(u=1, \ldots, k)$. We define an equivalence relation π on the set of indices of the α_i-product $\mathscr{B}$ as follows: for any l, n $(1 \leqq l, n \leqq s)$, $l \pi n$ holds if and only if $\mathscr{A}_l=\mathscr{A}_n$ and $a_{tl}=a_{tn}$ for all $t=1, \ldots, k$.

It is easy to see that the partition corresponding to π has at most $|M| \cdot V^k$ blocks. Since $\mu(A)$ is a subalgebra of $\mathscr{B}$, if $a_{tl}=a_{tn}$ $(t=1, \ldots, k)$ then the l-th and n-th components of $\mu(\sigma(a^1, \ldots, a^m))$ are equal, where $m \in R$, $\sigma \in \Sigma_m$, $a^j \in A$ $(j=1, \ldots, m)$. From this it follows that $\mathscr{A}$ can be embedded isomorphically into an α_i-product of tree automata from M with at most $|M| \cdot V^k$ factors.

DEPT. OF COMPUTER SCIENCE
A. JÓZSEF UNIVERSITY
ARADI VÉRTANÚK TERE 1
SZEGED, HUNGARY
H—6720

References

[1] ÉSIK, Z., On identities preserved by general products of algebras, Acta Cybernet., v. 6, 1983, pp. 285—289.
[2] GÉCSEG, F., Composition of automata, Proceedings of the 2nd Colloquium on Automata, Languages and Programming, Saarbrücken, 1974, Springer Lecture Notes in Computer Science, v. 14, pp. 351—363.
[3] IMREH, B., On α_t-products of automata, Acta Cybernet., v. 3, 1978, pp. 301—307.
[4] RICCI, G., Cascades of tree automata and computation in universal algebras, Mathematical System Theory, 7, 1973, pp. 201—218.
[5] STEINBY, M., On the structure and realizations of tree automata, Second Coll. sur les Arbres en Algebre et en Programmation, Lille, 1977, pp. 235—248.

(Received Aug. 22, 1986)

On a problem of Ádám concerning precodes assigned to finite Moore automata

MASASHI KATSURA

To investigate the structure of finite Moore automata, the concepts of code, precode and complexity are introduced by Ádám [1] and investigated in [1—8]. Main motivation is the following.

Basic Problem [1]. For arbitrary finite X, let a constructive description of all reduced finite Moore automata, whose input set equals to X, be given.

Relating to this problem Ádám raised four open problems, one of which is the following.

Problem 3 [1]. Consider all pairs (D, D') of precodes with finite complexity such that $D \prec D'$ holds. Either determine the maximal value of $\Omega(D') - \Omega(D)$ (as a function of the cardinality of input set) or prove that the set of these differences is unbounded.

In autonomous case, this problem is solved in [8]. The answer is that the difference is unbounded. However, we show in [8] that the quotient $\Omega(D')/\Omega(D)$

$$(D \prec D',\ \Omega(D) \neq 0,\ \Omega(D') < \infty)$$

is bounded by 2. In this note, it is shown that, in multiple-input case, not only the difference but also the quotient is unbounded.

For the background and fundamental facts concerning codes, precodes and complexity, see [1] and [2].

1.

$\mathbf{N}$ and $\mathbf{N}_0$ mean the sets of positive integers and of nonnegative integers, respectively. For $t, k \in \mathbf{N}_0$, we denote $[t:k] = \langle i \in \mathbf{N}_0 | t \leqq i \leqq k \rangle$. For $n, m \in \mathbf{N}$, we write $X_{(n)} = \langle x_1, \ldots, x_n \rangle$ and $Y_{(m)} = \langle y_1, \ldots, y_m \rangle$. A *partial automaton* is a 5-tuple $A = ([1:v], X_{(n)}, Y_{(m)}, \delta, \lambda)$ where:

(1) v, n and m are positive integers. $[1:v]$, $X_{(n)}$ and $Y_{(m)}$ are called the state set, the input set and the output set of A, respectively.

(2) δ is a partial mapping of $[1:v]\times X_{(n)}$ into $[1:v]$ called a state transition function (δ is extended as usual to a partial mapping of $[1:v]\times(X_{(n)})^*$ into $[1:v]$).
(3) λ is a mapping of $[1:v]$ *onto* $Y_{(m)}$ called an output function.
(4) For any $a\in[1:v]$ there exists a $p\in X^*$ such that $\delta(1,p)=a$.

If δ is defined for any element of $[1:v]\times X_{(n)}$, then A is said to be an *(initially connected finite) Moore automaton.*

Let $A=([1:v], X_{(n)}, Y_{(m)}, \delta, \lambda)$ be a Moore automaton. If $\lambda(\delta(a,p))\neq\lambda(\delta(b,p))$ holds for $a, b\in[1:v]$ and $p\in X^*$, then we say that p distinguishes between a and b. $\omega(a,b)$ is the minimal length of p which distinguishes between a and b. If there is no word which distinguishes between a and b, then we denote $\omega(a,b)=\infty$. Especially, $a=b$ implies $\omega(a,b)=\infty$. The *complexity* $\Omega(A)$ of A is defined by

$$\Omega(A)=\min\langle\omega(a,b)|a,b\in[1:v], a\neq b\rangle.$$

If $v=1$ then $\Omega(A)=0$.

The notions of codes and precodes were introduced in [1] as tools to describe Moore automata constructively. The following definition is from [6, 7]. It is of course essentially equivalent to Ádám's definition in [1].

Let $n\in\mathbf{N}$. A 6-tuple $D=(r,s,\beta,\gamma,\varphi,\mu)$ is said to be an *n-input precode* if the following eight postulates are fulfilled:

(A) r, s are nonnegative integers.
(B) β and φ are mappings of $[2:r+s+1]$ into $[1:r+1]$.
γ is a mapping of $[2:r+s+1]$ into $[1:n]$.
μ is a mapping of $[1:r+1]$ into $\mathbf{N}$.
(C) $\beta(a)<a$ for any $a\in[2:r+1]$.
(D) For $a, b\in[2:r+1]$, if $a<b$ then $(\beta(a),\gamma(a))<(\beta(b),\gamma(b))$ in the lexicographic order.
(E) For $a\in[r+2:r+s+1]$, $(\beta(a),\gamma(a))$ is the lexicographically smallest element in $([1:r+1]\times[1:n])-\langle(\beta(b),\gamma(b))|b\in[2:a-1]\rangle$.
(F) For $a\in[2:r+1]$, $\varphi(a)=a$.
(G) For $a\in[r+2:r+s+1]$, $\varphi(a)=1$ or $(\beta(\varphi(a)),\gamma(\varphi(a)))<(\beta(a),\gamma(a))$ in the lexicographic order.
(H) $\mu(a)\in\langle 1\rangle\cup\langle\mu(b)+1|b\in[1:a-1]\rangle$.

We denote $\mu(D)=\max\langle\mu(a)|a\in[1:r+1]\rangle$. If $m=\mu(D)$ then D is said to be an m-output precode.

It can be easily be seen that $r+s\leqq n(r+1)$ i.e., $s\leqq nr+n-r$. If $s=nr+n-r$, then the precode is said to be a *code.*

Let $D=(r,s,\beta,\gamma,\varphi,\mu)$ and $D'=(r',s',\beta',\gamma',\varphi',\mu')$ be n-input precodes. If $r+s\leqq r'+s'$ and β', γ', φ', μ' are extensions of β, γ, φ, μ then we denote $D\leqq D'$. We denote $D<D'$ if $D\leqq D'$ and $r+s<r'+s'$. If $D<D'$ and $r'+s'=r+s+1$ then we write $D\prec D'$.

It can easily be seen that, for any precode D, there exists a code C such that $D\leqq C$.

Let $D=(r,s,\beta,\gamma,\varphi,\mu)$ be an n-input m-output precode. Define a partial mapping δ_D of $[1:r+1]\times X_{(n)}$ into $[1:r+1]$ by

$$\delta_D(\beta(a), x_{\gamma(a)})=\varphi(a) \quad \text{for any} \quad a\in[2:r+s+1].$$

Define a mapping λ_D of $[1:r+1]$ onto $Y_{(m)}$ by

$$\lambda_D(a) = y_{\mu(a)} \quad \text{for any} \quad a \in [1:r+1].$$

Then it is easy to verify that $\Psi(D)=([1:r+1], X_{(n)}, Y_{(m)}, \delta_D, \lambda_D)$ is a partial automaton. $\Psi(D)$ is an automaton iff D is a code.

The *complexity* $\Omega(D)$ of a precode D is defined by

$$\Omega(D) = \min \langle \Omega(\Psi(C)) | C \text{ is a code such that } D \leqq C \rangle.$$

2.

Let n, w, t be positive integers such that $n \geqq 2$ and $w \geqq 2$. Define an n-input precode $D=(r, s, \beta, \gamma, \varphi, \mu)$ as follows:

(1) $r = 4t+4w-2$ and $s = nr+n-r-1$.

(2) $(\beta(2b), \gamma(2b), \varphi(2b)) = (b, 1, 2b)$ and $(\beta(2b+1), \gamma(2b+1), \varphi(2b+1)) = (b, n, 2b+1)$ for any $b \in [1:2t+2w-1]$.

(3) $\mu(a) = a$ for any $a \in [1:3t+3w-1]$.

$\mu(a) = a-w-t$ for any $a \in [3t+3w:4t+4w-1]$.

(4) For each $a \in [r+2:r+s+1]$, the a-th row is determined as follows:

(a) $\beta(a)$, $\gamma(a)$ are determined uniquely by Postulate (E).

(b) If $\beta(a) \in [2t+2w:3t+3w-2] \cup [3t+3w:4t+4w-2]$ and $\gamma(a) = 1$ then $\varphi(a) = a+1$.

If $(\beta(a), \gamma(a)) = (3t+3w-1, 1)$ then $\varphi(a) = 2t+2w$.

If $(\beta(a), \gamma(a)) = (4t+4w-1, 1)$ then $\varphi(a) = 3t+3w$.

(c) If $\beta(a) \in [2t+2w:3t+2w-1]$ and $\gamma(a) = n$ then $\varphi(a) = 3t+3w-1$.

If $\beta(a) \in [3t+3w:4t+3w-2]$ and $\gamma(a) = n$ then $\varphi(a) = 4t+4w-1$.

(d) Otherwise, $\varphi(a)=1$.

It is easy to verify that D satisfies Postulates (A)—(H).

The state transition function and the output function of the partial automaton $\Psi(D)=([1:4t+4w-1], X_{(n)}, Y_{(m)}, \delta_D, \lambda_D)$ is shown in the following table:

a	$\delta_D(a, x_1)$	$\delta_D(a, x_j)$ $(j\in[2: n-1])$	$\delta_D(a, x_n)$	$\lambda_D(a)$
1	2	1	3	1
2	4	1	5	2
3	6	1	7	3
⋮	⋮	⋮	⋮	⋮
$2t+2w-2$	$4t+4w-4$	1	$4t+4w-3$	$2t+2w-2$
$2t+2w-1$	$4t+4w-2$	1	$4t+4w-1$	$2t+2w-1$
$2t+2w$	$2t+2w+1$	1	$3t+3w-1$	$2t+2w$
$2t+2w+1$	$2t+2w+2$	1	$3t+3w-1$	$2t+2w+1$
⋮	⋮	⋮	⋮	⋮
$3t+2w-2$	$3t+2w-1$	1	$3t+3w-1$	$3t+2w-2$
$3t+2w-1$	$3t+2w$	1	$3t+3w-1$	$3t+2w-1$
$3t+2w$	$3t+2w+1$	1	1	$3t+2w$
$3t+2w+1$	$3t+2w+2$	1	1	$3t+2w+1$
⋮	⋮	⋮	⋮	⋮
$3t+2w-2$	$3t+3w-1$	1	1	$3t+3w-2$
$3t+3w-1$	$2t+2w$	1	1	$3t+3w-1$
$3t+3w$	$3t+3w+1$	1	$4t+4w-1$	$2t+2w$
$3t+3w+1$	$3t+3w+2$	1	$4t+4w-1$	$2t+2w+1$
⋮	⋮	⋮	⋮	⋮
$4t+3w-2$	$4t+3w-1$	1	$4t+4w-1$	$3t+2w-2$
$4t+3w-1$	$4t+3w$	1	1	$3t+2w-1$
$4t+3w$	$4t+3w+1$	1	1	$3t+2w$
$4t+3w+1$	$4t+3w+2$	1	1	$3t+2w+1$
⋮	⋮	⋮	⋮	⋮
$4t+4w-2$	$4t+4w-1$	1	1	$3t+3w-2$
$4t+4w-1$	$3t+3w$	1	—	$3t+3w-1$

Let $D'=(r, s+1, \beta, \gamma, \varphi, \mu)$ be a precode such that $D\prec D'$. Then D' is a code, i.e., $\Psi(D')$ is a Moore automaton. We have $(\beta(r+s+2), \gamma(r+s+2))=(4t+4w-1, n)$ and D' is determined only by the value $\varphi(r+s+2)$. It can easily be seen that arbitrary choice of $\varphi(r+s+2)\in[1:4w+4t-1]$ makes D' to satisfy the postulates for codes. We shall show that $\varphi(r+s+2)\neq 1$ implies $\Omega(D')=w$, and $\varphi(r+s+2)=1$ implies $\Omega(D')=t+w$.

Case 1: $\varphi(r+s+2)\neq 1$, i.e., $\delta_{D'}(4t+4w-1, x_n)\neq 1$.

Let $a, b\in[1:4t+4w-1]$ such that $a<b$. We have $\omega(a, b)\neq 0$ iff $\lambda_{D'}(a)=\lambda_{D'}(b)$ iff $a=2t+2w+i, b=3t+3w+i$ for some $i\in[0:t+w-1]$. Let $i\in[0:t+w-1]$. Since

$$\lambda_{D'}(2t+2w+i) = \lambda_{D'}(3t+3w+i),$$

$$\delta_{D'}(2t+2w+i, x_j) = \delta_{D'}(3t+3w+i, x_j) \quad \text{for any} \quad j\in[2:n-1],$$

we have

$$\omega(2t+2w+i, 3t+3w+i) =$$

$$= \min\langle \omega(\delta_{D'}(2t+2w+i, x_1), \delta_{D'}(3t+3w+i, x_1))+1,$$

$$\omega(\delta_{D'}(2t+2w+i, x_n), \delta_{D'}(3t+3w+i, x_n))+1\rangle.$$

Thus we have

$$\omega(3t+2w-1, 4t+3w-1) =$$

$$= \min\langle \omega(3t+2w, 4t+3w)+1, \omega(3t+3w-1, 1)+1\rangle = 1.$$

$$\omega(3t+3w-1, 4t+4w-1) =$$

$$= \min\langle \omega(2t+2w, 3t+3w)+1, \omega(1, \delta_{D'}(4t+4w-1, x_n))+1\rangle = 1.$$

For $i\in[0:t-2]$,

$$\omega(2t+2w+i, 3t+3w+i) =$$

$$= \min\langle \omega(2t+2w+i+1, 3t+3w+i+1)+1, \omega(3t+3w-1, 4t+4w-1)+1\rangle = 2.$$

For $i\in[0:w-2]$,

$$\omega(3t+2w+i, 4t+3w+i) = \omega(3t+2w+i+1, 4t+3w+i+1)+1.$$

Hence,

$$\omega(3t+3w-2, 4t+4w-2) = 2,$$

$$\omega(3t+3w-3, 4t+4w-3) = 3,$$

$$\dots$$

$$\omega(3t+2w, 4t+3w) = w.$$

Consequently, $\Omega(D')=\max\langle 0, 1, 2, \dots, w\rangle = w$.

Case 2: $\varphi(r+s+2)=1$, i.e., $\delta_{D'}(4t+4w-1, x_n)=1$.
Let $a, b\in[1:4t+4w-1]$ such that $a<b$. Just as in Case 1, we have

$\omega(a, b) \neq 0$ iff $a = 2t+2w+i$, $b = 3t+3w+i$ for some $i\in[0:t+w-1]$.

$\omega(3t+2w-1, 4t+3w-1) = \min\langle \omega(3t+2w, 4t+3w)+1, \omega(3t+2w-1, 1)+1\rangle = 1.$

$\omega(3t+2w+i, 4t+3w+i) = \omega(3t+2w+i+1, 4t+3w+i+1)+1$ for any $i\in[0:w-2]$.

We have

$$\omega(3t+3w-1, 4t+4w-1) = \omega(2t+2w, 3t+3w)+1.$$

For $i\in[0:t-2]$,

$$\omega(2t+2w+i, 3t+3w+i) =$$

$$= \min\langle \omega(2t+2w+i+1, 3t+3w+i+1)+1, \omega(3t+3w-1, 4t+4w-1)+1\rangle =$$

$$= \min\langle \omega(2t+2w+i+1, 3t+3w+i+1)+1, \omega(2t+2w, 3t+3w)+2\rangle.$$

It follows that

$$\omega(3t+2w-1, 4t+3w-1) = 1,$$
$$\omega(3t+2w-2, 4t+3w-2) = 2,$$
$$\omega(3t+2w-3, 4t+3w-3) = 3,$$
$$\cdots$$
$$\omega(2t+2w, 3t+3w) = t,$$
$$\omega(3t+3w-1, 4t+4w-1) = t+1,$$
$$\omega(3t+3w-2, 4t+4w-2) = t+2,$$
$$\cdots$$
$$\omega(3t+2w, 4t+3w) = t+w.$$

Consequently, $\Omega(D')=\max\langle 0, 1, 2, \ldots, t+w\rangle=t+w$.

We have shown that $\varphi(r+s+2)\neq 1$ implies $\Omega(D')=w$ and $\varphi(r+s+2)=1$ implies $\Omega(D')=w+t$. It follows that $\Omega(D)=\min\langle w, w+t\rangle=w$. We have shown the following.

Theorem 1. For any $n, w, t\in\mathbf{N}$ with $n\geqq 2$ and $w\geqq 2$, there exist n-input precodes D and D' such that $D\prec D'$, $\Omega(D)=w$ and $\Omega(D')=t+w$. □

In autonomous case, Problem 3 of Ádám is solved in [8] as follows:

Proposition 1. The set

$\langle\Omega(D')-\Omega(D)|D$ and D' are 1-input precodes such that $D\prec D'$ and $\Omega(D')<\infty\rangle$

coincides with all nonnegative integers. □

In multiple-input case, we have the following similar result which is an immediate consequence of Theorem 1.

Corollary 1. For any $n\in\mathbf{N}$ with $n\geqq 2$, the set

$\langle\Omega(D')-\Omega(D)|D$ and D' are n-input precodes such that $D\prec D'$ and $\Omega(D')<\infty\rangle$

coincides with all nonnegative integers. □

Consider the quotient $\Omega(D')/\Omega(D)$ instead of the difference $\Omega(D')-\Omega(D)$. In autonomous case, we have the following result [8].

Proposition 2. The set

$\langle\Omega(D')/\Omega(D)|D$ and D' are 1-input precodes such that $D\prec D'$, $\Omega(D)\neq 0$ and

$$\Omega(D')<\infty\rangle$$

coincides with all rational numbers between 1 and 2. □

Though the quotient is bounded in autonomous case, it is unbounded in multiple-input case. The following is also immediate from Theorem 1.

Corollary 2. For any $n\in\mathbf{N}$ with $n\geqq 2$, the set

$\langle \Omega(D')/\Omega(D) | D$ and D' are n-input precodes such that $D\prec D'$, $\Omega(D)\neq 0$ and $\Omega(D')<\infty\rangle$

coincides with all rational numbers not less than 1. □

Contrary to expectation, the solution of the problem does not contribute to our investigation, especially to the Basic Problem. If we wish to proceed further in this line, we should make refinements of the problem, e.g., not only n but also r, s and/or m should be taken into account.

3.

In this section, we consider a modification of our problem in the sense that, instead of the cardinality n of the input set, the cardinality m of the output set is taken into account. Analogous to Theorem 1, we have the following result:

Theorem 2. For any $m, w, t\in\mathbf{N}$ with $m\geqq 2$ and $w\geqq 2$, there exist m-output precodes D and D' such that $D\prec D'$, $\Omega(D)=w$, $\Omega(D')=t+w$.

Proof. Define a $(2t+2w)$-input precode $D=(r, s, \beta, \gamma, \varphi, \mu)$ as follows:

(1) $r=2t+2w+m-2$ and $s=(2t+2w)r+(2t+2w)-r-1$.

(2) $(\beta(a), \gamma(a), \varphi(a))=(a-1, 1, a)$ for any $a\in[2:m-1]$.

(3) $(\beta(a), \gamma(a), \varphi(a))=(m-1, a-m+1, a)$ for any $a\in[m:2t+2w+m-1]$.

(4) $\mu(a)=a$ for any $a\in[1:m-1]$.

$\mu(a)=m$ for any $a\in[m:2t+2w+m-1]$.

(5) For each $a\in[r+2, r+s+1]$, the a-th row is determined as follows:

(a) $\beta(a)$, $\gamma(a)$ are determined uniquely by Postulate (E).

(b) If $\beta(a)\in[m:t+w+m-2]\cup[t+w+m:2t+2w+m-2]$ and $\gamma(a)=1$ then $\varphi(a)=a+1$.

If $(\beta(a), \gamma(a))=(t+w+m-1, 1)$ then $\varphi(a)=m$.

If $(\beta(a), \gamma(a))=(2t+2w+m-1, 1)$ then $\varphi(a)=t+w+m$.

(c) If $\beta(a)-(m-2)=\gamma(a)\in[2:t+w+1]$ then $\varphi(a)=\beta(a)$.

If $\beta(a)-(t+w+m-2)=\gamma(a)\in[2:t+w+1]$ then $\varphi(a)=\beta(a)$.

(d) If $\beta(a)\in[m:t+m-1]$ and $\gamma(a)=2t+2w$ then $\varphi(a)=t+w+m-1$.

If $\beta(a)\in[t+w+m:2t+w+m-2]$ and $\gamma(a)=2t+2w$ then $\varphi(a)=2t+2w+m-1$.

(e) Otherwise, $\varphi(a)=1$.

Let $D'=(r, s+1, \beta, \gamma, \varphi, \mu)$ be a precode such that $D \prec D'$. Let

$$a, b \in [1:2t+2w+m-1]$$

such that $a<b$. If $a\in[1:m-1]$ then $\lambda_{D'}(a)\neq\lambda_{D'}(b)$ and thus $\omega(a, b)=0$. If $a\in[m:2t+2w+m-1]$ then there exist $i, j\in[0: t+w-1]$ such that

$$a = m+i \quad \text{or} \quad a = t+w+m+i,$$

$$b = m+j \quad \text{or} \quad b = t+w+m+j.$$

If $i\neq j$ then $\lambda_{D'}(a)=m=\lambda_{D'}(b)$ and

$$\lambda_{D'}(\delta_{D'}(a, x_{t+2})) = \lambda_{D'}(a) = m \neq 1 = \lambda_{D'}(1) = \lambda_{D'}(\delta_{D'}(b, x_{t+2})).$$

Hence $\omega(a, b)=1$. Consequently $\omega(a, b)\geqq 2$ implies that $a=m+i$ and $b=t+$ $+w+m+i$.

Similarly as in Theorem 1, we have, for any $i\in[0:t+w-1]$,

$$\omega(m+i, t+w+m+i) =$$

$$= \min\langle \omega(\delta_{D'}(m+i, x_1), \delta_{D'}(t+w+m+i, x_1))+1,$$

$$\omega(\delta_{D'}(m+i, x_n), \delta_{D'}(t+w+m+i, x_n))+1\rangle.$$

Since $\delta_{D'}(t+m-1, x_{2t+2w})=t+w+m-1$ and $\delta_{D'}(t+w+m-1, x_{2t+2w})=1$, we have $\omega(t+m-1, t+w+m-1)=1$. In a similar way as in Theorem 1, we can verify the following:

If $\varphi(r+s+2)\neq 1$ then $\Omega(D') = \omega(t+m, 2t+w+m) = w$.

If $\varphi(r+s+2) = 1$ then $\Omega(D') = \omega(t+m, 2t+w+m) = t+w$. □

The followings results are immediate from the above theorem.

Corollary 3. For any $m\in\mathbf{N}$ with $m\geqq 2$, the set

$\langle \Omega(D')-\Omega(D) | D$ and D' are m-output precodes such that $D\prec D'$ and $\Omega(D')<\infty\rangle$

coincides with all nonnegative integers. □

Corollary 4. For any $m\in\mathbf{N}$ with $m\geqq 2$, the set

$\langle \Omega(D')/\Omega(D) | D$ and D' are m-output precodes such that $D\prec D'$, $\Omega(D)\neq 0$ and $\Omega(D')<\infty\rangle$

coincides with all rational numbers not less than 1. □

DEPT. OF MATHEMATICS
KYOTO SANGYO UNIVERSITY
KYOTO 603
JAPAN

References

[1] ÁDÁM, A., On the question of description of the behavior of finite automata, Studia Sci. Math. Hungar. 13 (1978), 105—124.
[2] ÁDÁM, A., On the complexity of codes and pre-codes assigned to finite Moore automata, Acta Cybernet. 5 (1981), 117—133.
[3] ÁDÁM, A., On the simplicity of finite autonomous Moore automata, Studia Sci. Math. Hungar., 16 (1981), 427—436.
[4] ÁDÁM, A., On certain partitions of finite directed graphs and of finite automata, Acta Cybernet. 6 (1984), 331—346.
[5] KATSURA, M., On complexity of finite Moore automata, Acta Cybernet. 7 (1986), 281—292.
[6] KATSURA, M., Finiteness of complexity for precodes assigned to finite Moore automata, Topics in the Theoretical Bases and Applications of Computer Sci. (Proc. Conf., Győr, July 1985), Akadémiai Kiadó, Budapest, 1986; 91—102.
[7] KATSURA, M., Leaves of precodes assigned to finite Moore automata, Studia Sci. Math. Hungar. (To appear)
[8] KATSURA, M., On the complexity of finite autonomous Moore automata. Periodica Math. Hungar. (To appear)

(Received June 8, 1986)

An infinite hierarchy of tree transformations in the class $\mathcal{NDR}$

S. VÁGVÖLGYI and Z. FÜLÖP

Introduction

Let $S=\{\mathcal{DR}, \mathcal{LDR}, \mathcal{NDR}, \mathcal{LNDR}, \mathcal{H}, \mathcal{LH}, \mathcal{NH}\}$ where $\mathcal{DR}$ is the class of all deterministic root-to-frontier tree transformations, $\mathcal{H}$ is the class of all homomorphism tree transformations, moreover, for both $\mathcal{DR}$ and $\mathcal{H}$, their linear, nondeleting and linear-nondeleting subclasses are denoted by prefixing them by $\mathcal{L}$, $\mathcal{N}$ and $\mathcal{LN}$, respectively. Let $[S]$ be the set of classes of tree transformations generated by S with composition $\circ$: $[S]=\{\mathcal{K}_1\circ\ldots\circ\mathcal{K}_n | n\geqq 1,\ \mathcal{K}_i\in S,\ 1\leqq i\leqq n\}$. The set $[S]$ was introduced and examined in [1] where several equalities and inclusions were obtained with respect to elements of $[S]$. However, the question that whether $[S]$ is a finite or an infinite set was only raised and not answered.

In Section 2 of this paper we show that, in fact, $[S]$ is infinite by proving that $(\mathcal{LNDR}\circ\mathcal{NH})^m\subset(\mathcal{LNDR}\circ\mathcal{NH})^{m+1}$ for each $m\geqq 1$. This infinite proper hierarchy was already suggested by Theorem 12 of [1].

It is well known that $\mathcal{NDR}$ is closed under composition (proof, for example, in [1]). Thus we have $(\mathcal{LNDR}\circ\mathcal{NH})^m\subset\mathcal{NDR}$ for each $m\geqq 1$. In the second half of Section 2 we show that the stronger proper inclusion $\bigcup_{m=1}^{\infty}(\mathcal{LNDR}\circ\mathcal{NH})^m\subset\mathcal{NDR}$ is also valid.

The paper, apart from some simple reference to [1], is self-containing. Both in [1] and this paper, most of the notions and notations are adopted from [2].

1. Notions and notations

For an arbitrary set Y, we denote by Y^* the free monoid generated by Y, with empty word λ. The prefix ordering $\leqq$ in Y^* is meant as usual: for any $\alpha,\beta\in Y^*$, $\alpha\leqq\beta$ if and only if α is a prefix of β, that is, there exists a $\gamma\in Y^*$ such that $\beta=\alpha\gamma$. The relation $\alpha<\beta$ is defined by $\alpha\leqq\beta$ and $\alpha\neq\beta$.

The set of nonnegative integers is denoted by N. For each $n \in N$, $[n]$ denotes the set $\{1, ..., n\}$. Thus $[0]=\emptyset$.

By a ranked alphabet we mean an ordered pair (F, ν) where F is a finite set and $\nu: F \to N$ is the arity function. Elements of F are called function symbols, more exactly, if $f \in F$ and $\nu(f)=n$ then f is an n-ary function symbol. For any $n \in N$ we put $F_n=\{f \in F | \nu(f)=n\}$. Hence, for any ranked alphabet (F, ν), we have the equivalent notation $F=\bigcup_{n \in N} F_n$, where F_n are pairwise disjoint finite sets.

Let $F=\bigcup_{n \in N} F_n$ be a ranked alphabet and Y be a set, disjoint with F. Then the set of all terms or trees over Y of type F is defined as the smallest set $T_F(Y)$ satisfying:

(a) $Y \subseteq T_F(Y)$ and

(b) $f(p_1, ..., p_n) \in T_F(Y)$ whenever $f \in F_n$ and $p_1, ..., p_n \in T_F(Y)$.

For $f( )$ we write f. If $Y=\emptyset$ then $T_F(Y)$ is written as T_F.

We shall need a few of the usual functions on the elements of $T_F(Y)$: for any $p \in T_F(Y)$ the frontier $\mathrm{fr}(p) \in Y^*$, the set of subtrees or subterms $\mathrm{sub}(p) \subseteq T_F(Y)$, the paths $\mathrm{path}(p) \subseteq N^*$ and for each $m \in N$ the m-rank $\mathrm{rn}_m(p) \in N$ of p are defined by induction as follows:

(a) if $p \in Y$ then

$$\mathrm{fr}(p) = p, \quad \mathrm{sub}(p) = \{p\}, \quad \mathrm{path}(p) = \{\lambda\} \quad \text{and} \quad \mathrm{rn}_m(p) = 0;$$

(b) if $p = f(p_1, ..., p_n)$ for some $n \in N$, $f \in F_n$ and $p_1, ..., p_n \in T_F(Y)$ then

$$\mathrm{fr}(p) = \mathrm{fr}(p_1) \ldots \mathrm{fr}(p_n),$$

$$\mathrm{sub}(p) = \Big(\bigcup_{i \in [n]} \mathrm{sub}(p_i)\Big) \cup \{p\},$$

$$\mathrm{path}(p) = \{\lambda\} \cup \{i\alpha | i \in [n], \alpha \in \mathrm{path}(p_i)\} \quad \text{and}$$

$$\mathrm{rn}_m(p) = \begin{cases} \sum_{i \in [n]} \mathrm{rn}_m(p_i) & \text{if} \quad n \neq m \\ 1 + \sum_{i \in [n]} \mathrm{rn}_m(p_i) & \text{if} \quad n = m. \end{cases}$$

We mention that $\mathrm{rn}_m(p)$ means the number of occurrences of the m-ary function symbols in p. Moreover we define $\mathrm{rn}(p)=\sum_{m \in N} \mathrm{rn}_m(p)$.

Now let $p \in T_F(Y)$ and $\alpha \in \mathrm{path}(p)$. We introduce the notion of the subtree $\mathrm{str}(p, \alpha)$ and the symbol $\mathrm{lab}(p, \alpha)$ of p determined by α, moreover, the two length $|\alpha|_2$ of α in p in the following way:

(a) if $p \in Y$ then
$\mathrm{str}(p, \alpha) = p$, $\mathrm{lab}(p, \alpha) = p$ and $|\alpha|_2 = 0$;

(b) if $p=f(p_1, ..., p_n)$ for some $n \in N$, $f \in F_n$ and $p_1, ..., p_n \in T_F(Y)$ then α is either λ or of the form $i\alpha'$ for some $i \in [n]$ and $\alpha' \in \mathrm{path}(p_i)$. Thus

we define

$$\mathrm{str}(p, \alpha) = \begin{cases} p & \text{if } \alpha = \lambda \\ \mathrm{str}(p_i, \alpha') & \text{if } \alpha = i\alpha', \end{cases}$$

$$\mathrm{lab}(p, \alpha) = \begin{cases} f & \text{if } \alpha = \lambda \\ \mathrm{lab}(p_i, \alpha') & \text{if } \alpha = i\alpha', \end{cases}$$

$$|\alpha|_2 = \begin{cases} 0 & \text{if } \alpha = \lambda \text{ and } n < 2, \\ 1 & \text{if } \alpha = \lambda \text{ and } n \geqq 2, \\ |\alpha'|_2 & \text{if } \alpha = i\alpha' \text{ and } n < 2, \\ 1 + |\alpha'|_2 & \text{if } \alpha = i\alpha' \text{ and } n \geqq 2. \end{cases}$$

We note that in this latter definition $|\alpha'|_2$ is meant in p_i. We mention what the above three functions informally mean. It is well known that p can be considered as an ordered tree labelled by elements of $F \cup Y$, moreover α can be thought of as a path leading from the root to a node x of p. Now, $\mathrm{str}(p, \alpha)$ is the subtree of p the root of which is x, $\mathrm{lab}(p, \alpha)$ is the symbol in $F \cup Y$ x is labelled by, finally $|\alpha|_2$ is the number of the occurrences of function symbols with arity $m \geqq 2$ along the path α. We also note that α may be in $\mathrm{path}(q)$ for some $q \neq p$ and $|\alpha|_2$ in p may differ from $|\alpha_2|$ in q. However it will always be clear from the context in what p $|\alpha|_2$ is meant.

The countably infinite set $X = \{x_1, x_2, \ldots\}$ of variable symbols will be kept fix throughout this paper. The set of the first m elements $x_1, \ldots, x_m$ of X is denoted by X_m. The set $T_F(X_m)$ will be written as $T_{F,m}$.

$\hat{T}_{F,m}$ is the linear-nondeleting subset of $T_{F,m}$: for $p \in T_{F,m}$, $p \in \hat{T}_{F,m}$ iff each x_i appears exactly once in p ($i \in [m]$).

For $p, q \in T_{F,m}$ and $i \in [m]$, by the i product $p \cdot_i q$ of p by q we mean the tree obtained from p by substituting each occurrence of x_i in p by q.

Let $p \in T_{F,m}$ and $y_1, \ldots, y_m \in Y$. We denote by $p(y_1, \ldots, y_m)$ the tree obtained from p by substituting each occurrence of x_i in p by y_i for each $i \in [m]$. Of course we have $p(y_1, \ldots, y_m) \in T_F(Y)$.

We introduce one more definition concerning $T_{F,m}$. For $p \in T_{F,m}$ and $i \in [m]$, the set of i paths $\mathrm{path}_i(p)$ of p is given as follows:

(a) if $p = x_j$ for some $j \in [m]$ then

$$\mathrm{path}_i(p) = \begin{cases} \{\lambda\} & \text{if } i = j \\ \emptyset & \text{if } i \neq j, \end{cases}$$

(b) if $p = f(p_1, \ldots, p_n)$ for some $n \geqq 0$, $f \in F_n$ and $p_1, \ldots, p_n \in T_{F,m}$ then

$$\mathrm{path}_i(p) = \{j\alpha \mid j \in [n], \alpha \in \mathrm{path}_i(p_j)\}.$$

It is clear that $\mathrm{path}_i(p) \subseteq \mathrm{path}(p)$, moreover $\mathrm{path}_i(p)$ consists of all the elements of $\mathrm{path}(p)$ leading from the root to a terminal node of p labelled by x_i.

A tree transformation τ is defined as a subset of $T_F \times T_G$ where F and G are arbitrary ranked alphabets. In this way, τ can alternatively be considered as a relation from T_F to T_G.

For the sake of convenient proofs, we introduce the concept of the extended tree transformation. It is a subset τ of $T_F(X) \times T_G(X)$.

Since (extended) tree transformations are in fact relations, for any (extended) tree transformations τ and σ, the domain $\mathrm{dom}\,\tau$ and the composition $\tau\circ\sigma$ of τ and σ are defined as it is usual for relations. Moreover, for any two classes $\mathscr{K}_1$ and $\mathscr{K}_2$ of tree transformations we put:

$$\mathscr{K}_1\circ\mathscr{K}_2=\{\tau_1\circ\tau_2|\tau_1\in\mathscr{K}_1 \quad \text{and} \quad \tau_2\in\mathscr{K}_2\} \quad \text{and}$$

$$\mathscr{K}_1^n=\begin{cases}\mathscr{K}_1 & \text{if} \quad n=1\\ \mathscr{K}_1^{n-1}\circ\mathscr{K}_1 & \text{if} \quad n>1.\end{cases}$$

We are interested only in tree transformations which can be induced by deterministic root-to-frontier tree transducers.

A deterministic root-to-frontier tree transducer (DR transducer in the sequel) is a system

$$\mathfrak{A}=(F, A, G, P, a_0) \quad \text{where} \tag{1}$$

(a) F and G are ranked alphabets;
(b) A, the state set of $\mathfrak{A}$, is a ranked alphabet consisting of 1-ary function symbols, disjoint with F, G and X;
(c) a_0, the initial state of $\mathfrak{A}$, is a distinguished element of A;
(d) P is a finite set of so called rewriting rules (or simply rules) of the form

$$af(x_1, \ldots, x_n)\to q \tag{2}$$

where $a\in A$, $n\geqq 0$, $f\in F_n$ and $q\in T_G(AX_n)$;
(e) different rules of P have different left-hand sides.

We mention that above and in what follows we use the following notations. If A is the state set of a DR transducer and T is a set of terms then $AT=\{a(t)|a\in A, t\in T\}$. Moreover, for any $a\in A$ and $t\in T$, $a(t)$ is written as at.

Then it is clear that each rule (2) of P can also be written in both of the following two forms:

$$af(x_1, \ldots, x_n)\to\bar{q}(a_1x_{i_1}, \ldots, a_mx_{i_m}) \tag{3}$$

for some $m\geqq 0$, $\bar{q}\in\hat{T}_{G,m}$, $a_j\in A$ and $x_{i_j}\in X_n$ $(j\in[m])$;

$$af(x_1, \ldots, x_n)\to\bar{q}(a_{1_1}x_1, \ldots, a_{1_{m_1}}x_1, \ldots, a_{n_1}x_n, \ldots, a_{n_{m_n}}x_n) \tag{4}$$

where $m_i\geqq 0$, $a_{i_j}\in A$ $(i\in[n], j\in[m_i])$ and $\bar{q}\in\hat{T}_{G,m}$ $(m=m_1+\ldots+m_n)$.

Next we show how $\mathfrak{A}$ can be used to rewrite (or transform) terms of T_F to terms of T_G. To this end, we define the relation $\underset{\mathfrak{A}}{\Rightarrow}$ called direct derivation on the set $T_G(AT_F(X))$ in the following manner: for $p, q\in T_G(AT_F(X))$, $p\underset{\mathfrak{A}}{\Rightarrow}q$ if and only if q can be obtained from p by replacing an occurrence of a subtree of the form $af(p_1, \ldots, p_n)$ in p by the tree $\bar{q}(a_1p_{i_1}, \ldots, a_mp_{i_m})$ and the rule (3) is in P. The reflexive-transitive closure of $\underset{\mathfrak{A}}{\Rightarrow}$ is denoted by $\overset{*}{\underset{\mathfrak{A}}{\Rightarrow}}$ and called derivation. The tree transformation induced by $\mathfrak{A}$ with the state $a\in A$ is introduced as

$$\tau_{\mathfrak{A}(a)}=\{(p, q)|p\in T_F, q\in T_G \quad \text{and} \quad ap\overset{*}{\underset{\mathfrak{A}}{\Rightarrow}}q\},$$

moreover, the tree transformation $\tau_{\mathfrak{A}}$ induced by $\mathfrak{A}$ is meant $\tau_{\mathfrak{A}(a_0)}$:

$$\tau_{\mathfrak{A}}=\{(p, q)|p\in T_F, q\in T_G \quad \text{and} \quad a_0p\overset{*}{\underset{\mathfrak{A}}{\Rightarrow}}q\}.$$

Now we define the extended tree transformation induced by $\mathfrak{A}$. To this end, we need the following concept. Let $q' \in T_G(AX)$. We say that $q \in T_G(X)$ belongs to q' if it satisfies the following conditions:

(a) if $q' = ax_i$ for some $a \in A$ and $i \in N$ then $q = x_i$,

(b) if $q' = g(q'_1, \ldots, q'_n)$ for some $n \geqq 0$, $g \in G_n$ and $q'_1, \ldots, q'_n \in T_G(AX)$ then $q = g(q_1, \ldots, q_n)$ where q_j belongs to q'_j for each $j \in [n]$.

Informally, we say that q belongs to q' if and only if q can be obtained from q' by substituting each subtree of the form ax_i of q' by x_i.

The extended tree transformation $\tau'_{\mathfrak{A}(a)}$ induced by $\mathfrak{A}$ with the state $a \in A$ is given as follows: for any $p \in T_F(X)$ and $q \in T_G(X)$, $(p, q) \in \tau'_{\mathfrak{A}(a)}$ if and only if $\exists q' \in T_G(AX)$ such that $ap \overset{*}{\underset{\mathfrak{A}}{\Rightarrow}} q'$ and q belongs to q'. The extended tree transformation $\tau'_{\mathfrak{A}}$ induced by $\mathfrak{A}$ is defined as $\tau'_{\mathfrak{A}(a_0)}$.

We say that a tree transformation τ can be induced by some DR transducer $\mathfrak{A}$ if $\tau = \tau_{\mathfrak{A}}$ holds.

The tree transformations which can be induced by DR transducers are in fact partial mappings. This follows from (e) of the definition of the DR transducer.

Next we introduce some restrictions on DR transducers. We say that a DR transducer (1) is

(a) totally defined if for each $a \in A$ and $f \in F$ there is a rule (2) in P;

(b) linear (L) if for each rule (4) of P and $i \in [n]$, $m_i \leqq 1$;

(c) nondeleting (N) if for each rule (4) of P and $i \in [n]$, $m_i \geqq 1$;

(d) linear-nondeleting (LN) if it is linear and nondeleting;

(e) uniform (U) if for each rule (4) of P and $i \in [n]$, $a_{i_1} = \ldots = a_{i_{m_i}}$.

It is obvious that if a DR transducer (1) is a UDR transducer then each rule of P can also be written in the form $af(x_1, \ldots, x_n) \to \bar{q}(a_1 x_1, \ldots, a_n x_n)$ for some $\bar{q} \in T_{G,n}$ and $a_1, \ldots, a_n \in A$. Any LDR transducer is a UDR transducer, too.

A DR transducer (1) is an H transducer if it is totally defined and has only one state, i.e., $A = \{a_0\}$ holds. The LH, NH and LNH subclasses of the class of H transducers are defined in a natural way. Each H transducer is a UDR transducer, by definition.

The class of all tree transformations which can be induced by K transducers is denoted by $\mathcal{K}$ where K stands for any of DR, LDR, NDR, LNDR, UDR, H, LH, NH and LNH.

2. The problems and the solutions

Following [1], let S consist of $\mathcal{DR}$, $\mathcal{H}$ and their linear, nondeleting and linear nondeleting subclasses, that is, let $S = \{\mathcal{DR}, \mathcal{LDR}, \mathcal{NDR}, \mathcal{LNDR}, \mathcal{H}, \mathcal{LH}, \mathcal{NH}\}$. Moreover, define $[S]$ as the set of all the classes of tree transformations which can be obtained as compositions of elements of S: $[S] = \{\mathcal{K}_1 \circ \ldots \circ \mathcal{K}_n | n \geqq 1, \mathcal{K}_i \in S, 1 \leqq i \leqq n\}$.

In [1], it was raised the problem that whether $[S]$ is an infinite set. We shall prove that $[S]$ is infinite by showing that $[S]$ contains an infinite proper hierarchy of classes of tree transformations. Namely, we prove, in Theorem 3, that $(\mathcal{LNDR} \circ \mathcal{NH})^m \subset \subset (\mathcal{LNDR} \circ \mathcal{NH})^{m+1}$ for each $m \geqq 1$.

In connection with this hierarchy one more problem can be raised. By definition, $\mathcal{LNDR}$ and $\mathcal{NH}$ are subclasses of $\mathcal{NDR}$, moreover it is not difficult to

see that $\mathscr{NDR}$ is closed under composition (a proof is given, e.g., in [1]). These, together with the infinite proper hierarchy mentioned above yield the proper inclusion $(\mathscr{LNDR} \circ \mathscr{NH})^m \subset \mathscr{NDR}$ for each $m \geqq 1$. In the second half of this section we show that the proper inclusion $\bigcup_{m=1}^{\infty} (\mathscr{LNDR} \circ \mathscr{NH})^m \subset \mathscr{NDR}$ also holds. Namely, in Lemma 17, we give an NDR transducer $\mathfrak{A}$ for which there does not exist m with $\tau_{\mathfrak{A}} \in (\mathscr{LNDR} \circ \mathscr{NH})^m$.

We set out to solve the first problem.

First we make a trivial observation on UNDR transducers. Let $\mathfrak{A} = (F, A, G, P, a_0)$ be a UNDR transducer and the rule $af(x_1, \ldots, x_n) \to q(a_1 x_1, \ldots, a_n x_n)$ in P. Then for each $j \in [n]$ and $\gamma \in \mathrm{path}_j(q)$ the condition

$$\text{if} \quad n > 1 \quad \text{then} \quad |\gamma|_2 \geqq 1$$

holds, since from $|\gamma|_2 = 0$ it would follow that $\mathfrak{A}$ is a deleting DR transducer. Our first Lemma is essentially a consequence of this observation.

Lemma 1. Let $\mathfrak{A} = (F, A, G, P, a_0)$ be a UNDR transducer, moreover, $m \geqq 0$, $p \in T_{F,m}$, $q \in T_{G,m}$ and $a \in A$ be such that $(p, q) \in \tau'_{\mathfrak{A}(a)}$. Then

(a) for each $j \in [m]$ and $\alpha \in \mathrm{path}_j(p)$ there exists a $\beta \in \mathrm{path}_j(q)$ for which $|\alpha|_2 \leqq |\beta|_2$ and

(b) for each $j \in [m]$ and $\alpha \in \mathrm{path}_j(q)$ there exists a $\beta \in \mathrm{path}_j(p)$ with $|\beta|_2 \leqq |\alpha|_2$.

Proof. We prove only the part (a) of our statement since (b), as a converse of (a), can be shown in a similar way. We follow an induction on p.

If $p = x_i$ for some $i \in [m]$ then $q = x_i$ hence (a) trivially holds.

Now let $p = f(p_1, \ldots, p_n)$ for some $n \geqq 0$, $f \in F_n$ and $p_1, \ldots, p_n \in T_{F,m}$. By our supposition, there exists a rule $af(x_1, \ldots, x_n) \to \bar{q}(a_1 x_1, \ldots, a_n x_n)$ in P and there are $q_1, \ldots, q_n \in T_{G,m}$ for which $(p_i, q_i) \in \tau'_{\mathfrak{A}(a_i)}$ $(i \in [n])$ and $q = \bar{q}(q_1, \ldots, q_n)$. Since the case $n = 0$ is again trivial we may suppose that $n \geqq 1$. Then $\alpha = i\alpha'$ for some $i \in [n]$ and $\alpha' \in \mathrm{path}_j(p_i)$.

Let γ be an arbitrary element of $\mathrm{path}_i(\bar{q})$, which is not empty since $\mathfrak{A}$ is nondeleting, and let $\beta' \in \mathrm{path}_j(q_i)$ be such that $|\alpha'|_2 \leqq |\beta'|_2$. Put $\beta = \gamma\beta'$. We mention that β' exists because of the induction hypothesis and, obviously, $\beta \in \mathrm{path}_j(q)$. Now we distinguish the cases $n = 1$ and $n > 1$.

If $n = 1$ then

$$|\alpha|_2 = |i\alpha'|_2 = |\alpha'|_2 \leqq |\beta'|_2 \leqq |\gamma|_2 + |\beta'|_2 = |\gamma\beta'|_2 = |\beta|_2.$$

On the other hand, in the case $n > 1$, by our above note on $|\gamma|_2$ we have

$$|\alpha|_2 = |i\alpha'|_2 = 1 + |\alpha'|_2 \leqq |\gamma|_2 + |\beta'|_2 = |\gamma\beta'|_2 = |\beta|_2.$$

The proof is complete. □

Now we recall what we mean by the syntactic composition of two DR transducers. The exact definition can be found in [1].

Let $\mathfrak{A}_1 = (F^0, A_1, F^1, P_1, a_1)$ and $\mathfrak{A}_2 = (F^1, A_2, F^2, P_2, a_2)$ be two arbitrary DR transducers. Their syntactic composition is the DR transducer $\mathfrak{A}_1 \circ \mathfrak{A}_2 = (F^0, A_1 \times A_2, F^2, P, (a_1, a_2))$ where P is constructed in the following way. When-

ever $bf(x_1, \ldots, x_n) \to \bar{q}(b_1 x_{i_1}, \ldots, b_m x_{i_m})$ is a rule in P_1 and it holds that $c\bar{q} \overset{*}{\underset{\mathfrak{A}_2}{\Rightarrow}} q(c_{1_1}x_1, \ldots, c_{1_{v_1}}x_1, \ldots, c_{m_1}x_m, \ldots, c_{m_{v_m}}x_m)$ we put the rule $(b, c)f(x_1, \ldots, x_n) \to q((b_1, c_{1_1})x_{i_1}, \ldots, (b_1, c_{1_{v_1}})x_{i_1}, \ldots, (b_m, c_{m_1})x_{i_m}, \ldots, (b_m, c_{m_{v_m}})x_{i_m})$ in P. It is well known that this construction yields the application of $\mathfrak{A}_2$ after $\mathfrak{A}_1$ in a "step by step" way. A very useful property of the syntactic composition is the following: if $\mathfrak{A}_2$ is an NDR transducer then $\tau_{\mathfrak{A}_1 \circ \mathfrak{A}_2} = \tau_{\mathfrak{A}_1} \circ \tau_{\mathfrak{A}_2}$ (for a proof, see Lemma 3 in [1]).

We shall need the generalisation of the syntactic composition and the above equality for any $m \geqq 2$. Therefore we make the following definition.

Definition 1. Let $m \geqq 2$ and let $\mathfrak{A}_i$ be a DR transducer for each $i \in [m]$. By the syntactic composition of $\mathfrak{A}_1, \ldots, \mathfrak{A}_m$ we mean the DR transducer defined above if $m=2$ and the DR transducer $(\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_{m-1}) \circ \mathfrak{A}_m$ if $m>2$.

Then, using Lemma 3 of [1] as a basis, the following statement can be verified by an induction on m: if $\mathfrak{A}_2, \ldots, \mathfrak{A}_m$ are NDR transducers then $\tau_{\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_m} = \tau_{\mathfrak{A}_1} \circ \ldots \circ \tau_{\mathfrak{A}_m}$.

The next lemma says that this equality is also valid for extended tree transformations.

Lemma 2. Let $m \geqq 2$ and let $\mathfrak{A}_i = (F^{i-1}, A_i, F^i, P_i, a_i)$ be a DR transducer for each $i \in [m]$ such that $\mathfrak{A}_2, \ldots, \mathfrak{A}_m$ are NDR transducers. Then $\tau'_{\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_m} = \tau'_{\mathfrak{A}_1} \circ \ldots \circ \tau'_{\mathfrak{A}_m}$.

Proof. Induction on m. For $m=2$ it is enough to show that for each $n \geqq 0$, $p \in T_{F^0,n}$, $q \in T_{F^2,n}$, $b_1 \in A_1$ and $b_2 \in A_2$ the following equivalence holds:

$$(p, q) \in \tau'_{\mathfrak{A}_1 \circ \mathfrak{A}_2((b_1, b_2))} \Leftrightarrow (\exists r \in T_{F^1,n})((p, r) \in \tau'_{\mathfrak{A}_1(b_1)} \text{ and } (r, q) \in \tau'_{\mathfrak{A}_2(b_2)}).$$

This can be verified by an induction on p. The detailed proof is omitted.

Finally, the induction step of m is shown by the following computation

$$\tau'_{\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_m} = \tau'_{(\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_{m-1}) \circ \mathfrak{A}_m} = \tau'_{\mathfrak{A}_1 \circ \ldots \circ \mathfrak{A}_{m-1}} \circ \tau'_{\mathfrak{A}_m} = \tau'_{\mathfrak{A}_1} \circ \ldots \circ \tau'_{\mathfrak{A}_m}. \quad \square$$

At this point we declare our main theorem.

Theorem 3. For any $m \geqq 2$ and $1 \leqq k < m$ $(\mathcal{LNDR} \circ \mathcal{NH})^k \subset (\mathcal{LNDR} \circ \mathcal{NH})^m$.

Proof. Because the complete proof is rather long we structured it in the following way. First we give a tree transformation τ_m which is in $(\mathcal{LNDR} \circ \mathcal{NH})^m$. Then we present Lemma 4 which concers any $\mathcal{NDR}$ transducer which induces τ_m. After this we suppose that $\tau_m \in (\mathcal{LNDR} \circ \mathcal{NH})^k$ for some $k \leqq m$ and, during a series of lemmas from 5 to 14, show, in Lemma 14, that $k<m$ is impossible.

Take an arbitrary integer $m \geqq 2$ and keep it fixed in the rest of the proof of this theorem. To define τ_m we introduce an LNDR transducer $\mathfrak{A}$ and an NH transducer $\mathfrak{B}$ as follows.

Let the LNDR transducer $\mathfrak{A} = (F, \{a, d\}, F', P, a)$ be determined by the following conditions:

(a) $F = F_0 \cup F_2 \cup F_3$, $F_0 = \{\#\}$, $F_2 = \{f_2\}$ and $F_3 = \{g_3\}$;

(b) $F'=F'_0\cup F'_2\cup F'_3$, $F'_0=\{\#\}$, $F'_2=\{f_2,f'_2\}$ and $F'_3=\{g_3\}$;

(c) P consists of the rules (i)—(vi) listed below

(i) $a\# \to \#$

(ii) $af_2(x_1, x_2)\to f'_2(dx_1, dx_2)$

(iii) $ag_3(x_1, x_2, x_3)\to g_3(dx_1, ax_2, dx_3)$

(iv) $d\# \to \#$

(v) $df_2(x_1, x_2)\to f_2(dx_1, dx_2)$

(vi) $dg_3(x_1, x_2, x_3)\to g_3(dx_1, dx_2, dx_3)$.

Moreover, introduce the NH transducer $\mathfrak{B}=(F', \{b\}, F, P', b)$ with P' containing the following rules:

(i) $b\# \to \#$

(ii) $bf'_2(x_1, x_2)\to g_3(bx_1, bx_1, bx_2)$

(iii) $bf_2(x_1, x_2)\to f_2(bx_1, bx_2)$

(iv) $bg_3(x_1, x_2, x_3)\to g_3(bx_1, bx_2, bx_3)$.

It is not difficult to see how $\mathfrak{A}$ and after that $\mathfrak{B}$ works on a tree $p\in T_F$. First $\mathfrak{A}$, with its state $a\in A$, searches for the first occurrence of f_2 on the path of p leading along the "middle branches" of a (possibly empty) sequence of g_3's and if it is found then rewrites it to f'_2 producing a tree $p'\in T_{F'}$. Any other symbol of p stays as it was. Then $\mathfrak{B}$ looks for this f'_2 in p' and duplicates the subtree on the first branch of f'_2 by substituting f'_2 by g_3. The other symbols of p' remain unchanged.

We put $\tau_m=(\tau_{\mathfrak{A}}\circ\tau_{\mathfrak{B}})^m$. Of course, $\tau_m\in(\mathscr{LNDR}\circ\mathscr{NH})^m$.

Now, for each $i\geqq 1$, we define a pair of trees $P_i, Q_i\in T_{F,i+1}$ recursively as follows:

(a) $P_1=f_2(x_2,x_1)$, $Q_1=g_3(x_2, x_2, x_1)$,

(b) $P_i=f_2\left(P_{i-1}(x_2, \ldots, x_{i+1}), x_1\right)$, $Q_i=g_3\big(P_{i-1}(x_2, \ldots, x_{i+1}), Q_{i-1}(x_2, \ldots, x_{i+1}), x_1\big)$ if $i>1$.

To make it clearer, P_m and Q_m are visualized in Fig. 1.

Let us introduce the notation $\tau'_m=(\tau'_{\mathfrak{A}}\circ\tau'_{\mathfrak{B}})^m$. By the definitions of $\mathfrak{A}$ and $\mathfrak{B}$, it can easily be verified that $(P_m, Q_m)\in\tau'_m$ moreover, that for each $t_1, \ldots, t_{m+1}\in T_F$ it holds

$$\big(P_m(t_1, \ldots, t_{m+1}), Q_m(t_1, \ldots, t_{m+1})\big)\in\tau_m. \tag{5}$$

Lemma 4. Let $\mathfrak{L}=(F, C, F, P'', c_0)$ be an NDR transducer with $\tau_{\mathfrak{L}}=\tau_m$. Then we have $(P_m, Q_m)\in\tau'_{\mathfrak{L}}$.

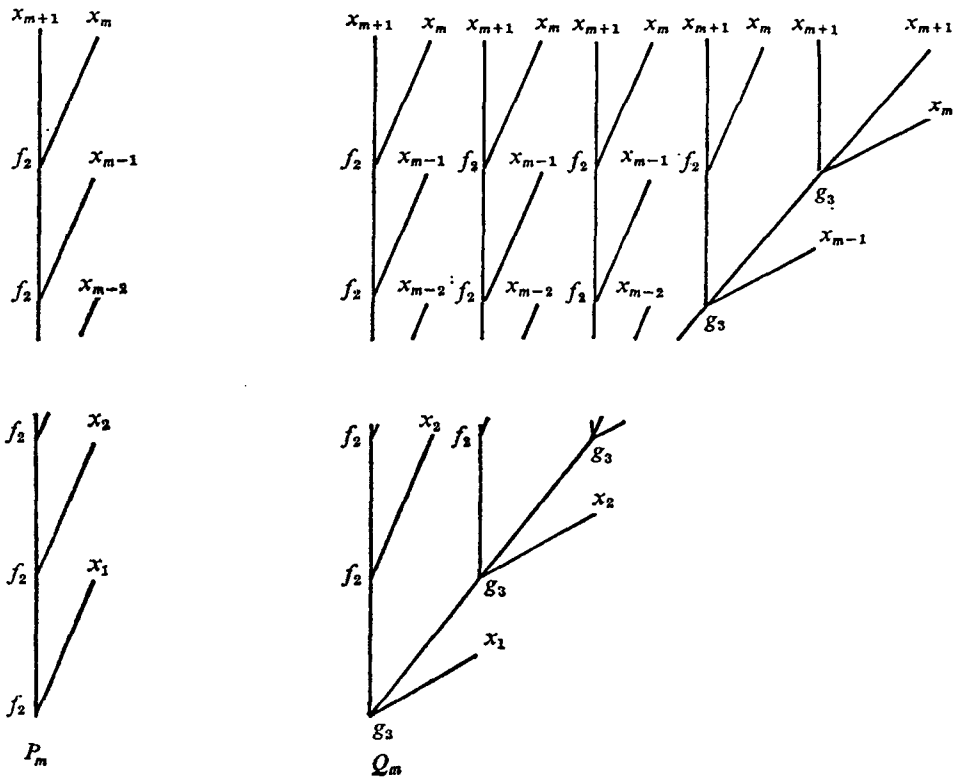


Figure 1.

Proof. First we note that $P_m \in \operatorname{dom} \tau'_{\mathfrak{L}}$ since, among others,

$$P_m(\#, \ldots, \#) \in \operatorname{dom} \tau_m = \operatorname{dom} \tau_{\mathfrak{L}}$$

thus (P_m, R_m) must be in $\tau'_{\mathfrak{L}}$ for some $R_m \in T_{F,m+1}$. It is obvious that R_m can be written in the form $\bar{R}_m(\overbrace{x_1, \ldots, x_1}^{n_1 \text{ times}}, \ldots, \overbrace{x_{m+1}, \ldots, x_{m+1}}^{n_{m+1} \text{ times}})$ for some $n_i \geqq 1$ $(i \in [m+1])$, $\bar{R}_m \in \hat{T}_{F,n}$ where $n = n_1 + \ldots + n_{m+1}$. Then it follows that for each $t_1, \ldots, t_{m+1} \in T_F$ the tree $\tau_{\mathfrak{L}}(P_m(t_1, \ldots, t_{m+1}))$ can be written in the form $\bar{R}_m(t_{1_1}, \ldots, t_{1_{n_1}}, \ldots, t_{m+1_1}, \ldots, t_{m+1_{n_{m+1}}})$ where for each $i \in [m+1]$ and $j \in [n_i]$ $(t_i\, t_{i_j}) \in \tau_{\mathfrak{L}(c_{i_j})}$ for some $c_{i_j} \in C$. Using these notations we have that for each $t_1, \ldots, t_{m+1} \in T_F$

$$Q_m(t_1, \ldots, t_{m+1}) = \bar{R}_m(t_{1_1}, \ldots, t_{1_{n_1}}, \ldots, t_{m+1_1}, \ldots, t_{m+1_{n_{m+1}}}).$$

We shall use this equality under different choices of $t_1, \ldots, t_{m+1}$ in the sequel of this proof.

Now suppose that $Q_m \neq R_m$. This means that for some $\alpha \in \operatorname{path}(Q_m) \cap \operatorname{path}(R_m)$, $\operatorname{lab}(Q_m, \alpha) \neq \operatorname{lab}(R_m, \alpha)$. Then four different cases are possible, each of which yields a contradiction:

(a) $\mathrm{lab}(Q_m, \alpha)=f$, $\mathrm{lab}(R_m, \alpha)=g$ for some $f, g\in F$ such that $f\neq g$. But then

$$f=\mathrm{lab}(Q_m, \alpha)=\mathrm{lab}(Q_m(\#, \ldots, \#), \alpha)=\mathrm{lab}(\bar{R}_m(\#_{1_1}, \ldots, \#_{1_{n_1}}, \ldots, \#_{m+1_1}, \ldots$$

$$\ldots, \#_{m+1_{n_{m+1}}}), \alpha) = \mathrm{lab}(R_m, \alpha)=g$$

which is impossible.

(b) $\mathrm{lab}(Q_m, \alpha)=x_i$, $\mathrm{lab}(R_m, \alpha)=g$ for some $i\in[m+1]$ and $g\in F$. Now, on the one hand $g=\#$, by $\#=\mathrm{lab}(Q_m(\#, \ldots, \#), \alpha)=\mathrm{lab}(\bar{R}_m(\#_{1_1}, \ldots, \#_{1_{n_1}}, \ldots,$ $\#_{m+1_1}, \ldots, \#_{m+1_{n_{m+1}}}), \alpha)=\mathrm{lab}(R_m, \alpha)=g$. On the other hand, for any $t\in T_F$

$$t = \mathrm{lab}(Q_m(\#, \ldots, \overset{i-\mathrm{th}}{t}, \ldots, \#), \alpha) =$$

$$= \mathrm{lab}(\bar{R}_m(\#_{1_1}, \ldots, \#_{1_{n_1}}, \ldots, t_{i_1}, \ldots, t_{i_l}, \ldots, \#_{m+1_1}, \ldots, \#_{m+1_{n_{m+1}}}), \alpha) =$$

$$= \mathrm{lab}(R_m, \alpha) = g = \#,$$

a contradiction.

(c) $\mathrm{lab}(Q_m, \alpha)=f$, $\mathrm{lab}(R_m, \alpha)=x_i$ for some $f\in F$ and $i\in[m+1]$. Then it can be seen from the definition of Q_m (see Fig. 1) that in this case $\mathrm{str}(Q_m, \alpha)$ contains at least one x_j with $j\neq i$ whatever i be. But then for any $t\in T_F$ it holds that t is a subtree of $\mathrm{str}(Q_m(\#, \ldots, \overset{j-\mathrm{th}}{t}, \ldots, \#), \alpha)$. It also holds that

$$\mathrm{str}(Q_m, (\#, \ldots, \overset{j-\mathrm{th}}{t}, \ldots, \#), \alpha) =$$

$$= \mathrm{str}(\bar{R}_m(\#_{1_1}, \ldots, \#_{1_{n_1}}, \ldots, t_{j_1}, \ldots, t_{j_{n_j}}, \ldots, \#_{m+1_1}, \ldots, \#_{m+1_{n_{m+1}}}), \alpha) = \#_{i_l}$$

for some $l\in[n_i]$. Contradiction since $\#_{i_l}$ does not depend on t chosen arbitrarily.

(d) $\mathrm{lab}(Q_m, \alpha)=x_i$, $\mathrm{lab}(R_m, \alpha)=x_j$ for some $i, j\in[m+1]$ such that $i\neq j$. Let t be an arbitrary element of T_F with $\mathrm{rn}(t)>0$. We have that

$$\# = \mathrm{lab}(Q_m(\#, \ldots, \overset{j-\mathrm{th}}{t}, \ldots, \#), \alpha) =$$

$$= \mathrm{lab}(\bar{R}_m(\#_{1_1}, \ldots, \#_{1_{n_1}}, \ldots, t_{j_1}, \ldots, t_{j_{n_j}}, \ldots, \#_{m+1_1}, \ldots, \#_{m+1_{n_{m+1}}}), \alpha) = t_{j_l}$$

for some $l\in[n_j]$, moreover $(t, t_{j_l})\in\tau_{\mathfrak{L}(c_{j_l})}$ for some $c_{j_l}\in C$. But $\mathfrak{L}$ is an NDR transducer hence $\mathrm{rn}(t_{j_l})>0$ which is again impossible. □

Let $k\leqq m$ and assume that $\tau_m\in(\mathscr{LNDR}\circ\mathscr{NH})^k$. This means in a more detailed form that for each $i\in[2k]$ there exists a DR transducer $\mathfrak{A}_i=(F^{i-1}, A_i, F^i, P_i, a_i)$ such that the following conditions hold

(a) $F^0=F^{2k}=F$,
(b) if i is odd then $\mathfrak{A}_i$ is an LNDR transducer (6)
(c) if i is even then $\mathfrak{A}_i$ is an NH transducer
(d) $\tau_{\mathfrak{A}_1}\circ\ldots\circ\tau_{\mathfrak{A}_{2k}}=\tau_m$.

Since each $\mathfrak{A}_i$ is an NDR transducer too, combining Lemmas 2 and 4 we have that

$$(P_m, Q_m) \in \tau'_{\mathfrak{A}_1} \circ \ldots \circ \tau'_{\mathfrak{A}_{2k}}, \tag{7}$$

or in other words, for each $i \in [2k]$ there exists a pair of trees $r_{i-1} \in T_{F^{i-1}, m+1}$ and $r_i \in T_{F^i, m+1}$ for which $r_0 = P_m$, $r_{2k} = Q_m$ and $(r_{i-1}, r_i) \in \tau'_{\mathfrak{A}_i}$. In fact, (7) is the relation which leads us to a contradiction in Lemma 14.

Lemma 5. For each $i \in [2k]$ it holds that $\mathrm{rn}_0(r_{i-1}) = 0$, that is, r_{i-1} does not contain any symbol of arity 0.

Proof. We observe that if $\mathrm{rn}_0(r_{i-1}) \neq 0$ then $\mathrm{rn}_0(r_i) \neq 0$ since $\mathfrak{A}_i$ is an NDR transducer. Now if for some $i \in [2k]$ $\mathrm{rn}_0(r_{i-1}) \neq 0$ then we obtain that $\mathrm{rn}_0(r_{2k}) \neq 0$ which, by the definition of r_{2k}, is a contradiction. □

Next we make a remark on the paths of r_0 and r_{2k} leading to x'_js ($j \in [m+1]$). Namely, we observe that whenever $j \in [m+1]$ and α is an element of either $\mathrm{path}_j(r_0)$ or $\mathrm{path}_j(r_{2k})$, by the definition of r_0 and r_{2k}, we have

$$|\alpha|_2 = \begin{cases} j & \text{if } j \in [m] \\ m & \text{if } j = m+1. \end{cases} \tag{8}$$

This can easily be read from Figure 1.

Lemma 6. For each $i \in [2k]$, $j \in [m+1]$ and $\alpha \in \mathrm{path}_j(r_{i-1})$, $|\alpha|_2$ is the same as in (8).

Proof. By part (a) of Lemma 1, for each $\alpha \in \mathrm{path}_j(r_{i-1})$ there exists a $\beta \in \mathrm{path}_j(r_i)$ with $|\alpha|_2 \leqq |\beta|_2$. Hence, if for some $i \in [2k]$ $j \in [m+1]$ and $\alpha \in \mathrm{path}_j(r_{i-1})$, $|\alpha|_2 > j$ when $j \in [m]$ and $|\alpha|_2 > m$ when $j = m+1$ holds then we obtain that for some $\beta \in \mathrm{path}_j(r_{2k})$, $|\beta|_2 > j$ if $j \in [m]$ and $|\beta|_2 > m$ if $j = m+1$. This, however, contradicts (8).

In a similar way, using part (b) of Lemma 1, the condition $|\alpha|_2 < j$ if $j \in [m]$ and $|\alpha|_2 < m$ if $j = m+1$ yields the existence of a $\beta \in \mathrm{path}_j(r_0)$ with the same property as α has, contradicting again (8). □

Lemma 7. Let $i \in [2k]$ and $r'_i \in T_{F^i}(A_i X_{m+1})$ be such that

$$a_i r_{i-1} \overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r'_i.$$

Suppose that the rule $cf(x_1, \ldots, x_n) \to q(c_1 x_1, \ldots, c_n x_n)$ was applied in the above derivation, where $f \in F_n^{i-1}$ for some $n \geqq 1$, $c, c_1, \ldots, c_n \in A_i$ and $q \in T_{F^i, n}$. Then for each $j \in [n]$ and $\gamma \in \mathrm{path}_j(q)$ it holds

$$|\gamma|_2 = \begin{cases} 0 & \text{if } n = 1, \\ 1 & \text{if } n > 1. \end{cases}$$

(We mention that r_i belongs to r'_i.)

Proof. By the conditions of our lemma, there exist the terms $s_{i-1} \in T_{F^{i-1}, m+2}$, $t_1, \ldots, t_n \in T_{F^{i-1}, m+1}$, $s_i \in T_{F^i, m+2}$ and $q_1, \ldots, q_n \in T_{F^i, m+1}$ such that each of the following conditions is satisfied.

(a) s_{i-1} contains exactly one occurrence of x_{m+2},

(b) $r_{i-1} = s_{i-1} \cdot_{m+2} f(t_1, \ldots, t_n)$,

(c) $r_i = s_i \cdot_{m+2} q(q_1, \ldots, q_n)$,

(d) $(s_{i-1}, s_i) \in \tau'_{\mathfrak{A}_i}$, $(t_j, q_j) \in \tau'_{\mathfrak{A}_i(c_j)}$ $(j \in [n])$.

Let us suppose that for some $j \in [n]$ and $\gamma \in \mathrm{path}_j(q)$ $|\gamma|_2$ violates the condition stated by our lemma. By Lemma 5 and (a) we can choose an $l \in [m+1]$ such that for some $\alpha \in \mathrm{path}_l(r_{i-1})$ α can be written in the form $\alpha = \alpha_1 j \alpha_2$ where $\alpha_1 \in \mathrm{path}_{m+2}(s_{i-1})$ and $\alpha_2 \in \mathrm{path}_l(t_j)$. Moreover, by Lemma 1, there exist $\beta_1 \in \mathrm{path}_{m+2}(s_i)$ and $\beta_2 \in \mathrm{path}_l(q_j)$ with $|\alpha_1|_2 \leqq |\beta_1|_2$ and $|\alpha_2|_2 \leqq |\beta_2|_2$. Letting $\beta = \beta_1 \gamma \beta_2$ we obviously have that $\beta \in \mathrm{path}_l(r_i)$.

First consider the case $n=1$. By our indirect assumption, $|\gamma|_2 > 0$, from which we have

$$|\alpha|_2 = |\alpha_1 j \alpha_2|_2 = |\alpha_1|_2 + |\alpha_2|_2 < |\beta_1|_2 + |\gamma|_2 + |\beta_2|_2 = |\beta_1 \gamma \beta_2|_2 = |\beta|_2$$

contradicting Lemma 6.

Now assume that $n>1$. In this case $|\gamma|_2 = 0$ is impossible by our observation made at the beginning of this section hence the indirect assumption is $|\gamma|_2 > 1$. But then

$$|\alpha|_2 = |\alpha_1 j \alpha_2|_2 = |\alpha_1|_2 + 1 + |\alpha_2|_2 < |\beta_1|_2 + |\gamma|_2 + |\beta_2|_2 = |\beta_1 \gamma \beta_2|_2 = |\beta|_2,$$

a contradiction. □

Lemma 8. For each $i \in [2k]$ and $n \geqq 4$, $\mathrm{rn}_n(r_{i-1}) = 0$.

Proof. Suppose it does not hold. Let $i \in [2k]$ be the greatest integer for which $\mathrm{rn}_n(r_{i-1}) > 0$ for some $n \geqq 4$. Then in the derivation $a_i r_{i-1} \overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r'_i (r'_i \in T_{F^i}(A_i X_{m+1}))$ it has to be applied at least one rule $cf(x_1, \ldots, x_n) \to q(cx_1, \ldots, cx_n)$ for which $n \geqq 4$ and $\mathrm{rn}_l(q) = 0$ for each $l \geqq 4$. Since $\mathfrak{A}_i$ is an NDR transducer it can be possible only if $|\gamma|_2 > 1$ for some $j \in [n]$ and $\gamma \in \mathrm{path}_j(q)$. This is a contradiction, by Lemma 7. □

At this point of the proof we can declare that for each $i \in [2k]$, every function symbol of r_{i-1} is in $F_1^{i-1} \cup F_2^{i-1} \cup F_3^{i-1}$.

Lemma 9. Let $i \in [2k]$ and $r'_i \in T_{F^i}(A_i X_{m+1})$ be such that

$$a_i r_{i-1} \overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r'_i.$$

Suppose that in the above derivation it was applied a rule $cf(x_1, \ldots, x_n) \to q$ where $f \in F_n^{i-1}$, $n \geqq 1$, $c \in A_i$ and $q \in T_{F^i}(A_i X_n)$. Then $n \in [3]$ and q can be written in one of the following three forms, for some suitable $u_0, u_1, u_2, u_3 \in T_{F_1^i, 1}$ $c_1, c_2, c_3 \in A_i$, $g \in F_2^i$ and $h \in F_3^i$:

(a) if $n = 1$ then $q = u_0(c_1 x_1)$,

(b) if $n = 2$ then either

$q = u_0(g(u_1(c_1x_{i_1}), u_2(c_2x_{i_2})))$ where $\{i_1, i_2\} = [2]$ or (9)

$q = u_0(h(u_1(c_1x_{i_1}), u_2(c_2x_{i_2}), u_3(c_3x_{i_3})))$ where

$\{i_1, i_2, i_3\} = [2]$,

(c) if $n = 3$ then

$q = u_0(h(u_1(c_1x_{i_1}), u_2(c_2x_{i_2}), u_3(c_3x_{i_3})))$ with

$\{i_1, i_2, i_3\} = [3]$.

(We note that in the notation $T_{F_1^i,1}$, F_1^i is considered a ranked alphabet. Thus the condition $u_j \in T_{F_1^i,1}$ means that every function symbol of u_j is in F_1^i $(j=0,1,2,3)$.)

Proof. Immediate from Lemmas 5, 7, 8 and from the fact that $\mathfrak{A}_i$ is an NDR transducer. □

Definition. We say that for some $i \in [2k]$ r_{i-1} has property (10) if

(a) for some $f \in F_3^{i-1}$ and $p_1, p_2, p_3 \in T_{F^{i-1}, m+1}$, $f(p_1, p_2, p_3) \in \mathrm{sub}(r_{i-1})$

and (10)

(b) for each $j \in [3]$, $n_j > 0$ where $n_j = \max\{|\alpha|_2 \mid \alpha \in \mathrm{path}_l(p_j),\ l \in [m+1]\}$.

Lemma 10. There exists no $i \in [2k]$ for which r_{i-1} has property (10).

Proof. It is enough to show that whenever r_{i-1} has property (10) then so does r_i. This proves our lemma since r_{2k}, by its definition, does not have property (10).

To this end, let us suppose that r_{i-1} has property (10) $(i \in [2k])$. Then, from Lemma 9, it follows that for some suitable $s_{i-1} \in T_{F^{i-1}, m+2}$, $s_i \in T_{F^i, m+2}$, $u_0, u_1, u_2, u_3 \in T_{F_1^i,1}$, $c, c_1, c_2, c_3 \in A_i$ and $q_1, q_2, q_3 \in T_{F^i, m+1}$ the following relations hold:

(a) $r_{i-1} = s_{i-1} \cdot_{m+2} f(p_1, p_2, p_3)$,

(b) $r_i = s_i \cdot_{m+2} u_0(h(u_1(q_1), u_2(q_2), u_3(q_3)))$,

(c) $cf(x_1, x_2, x_3) \to u_0(h(u_1(c_1x_{i_1}), u_2(c_2x_{i_2}), u_3(c_3x_{i_3}))) \in P_i$, $\{i_1, i_2, i_3\} = [3]$,

(d) $(s_{i-1}, s_i) \in \tau'_{\mathfrak{A}_i}$, $(p_{i_j}, q_j) \in \tau'_{\mathfrak{A}_i(c_j)}$ for each $j \in [3]$.

Moreover, for each $j \in [3]$, there exist $l_j \in [m+1]$ and $\alpha_j \in \mathrm{path}_{l_j}(p_{i_j})$ with $|\alpha_j|_2 = n_{i_j}$. By Lemma 1, there exist $\beta_j \in \mathrm{path}_{l_j}(q_j)$ such that $|\alpha_j|_2 \leqq |\beta_j|_2$. This shows that r_i has property (10) with $h(u_1(q_1), u_2(q_2), u_3(q_3))$. □

Definition. Let $i \in [2k]$. We say that r_{i-1} has property (11) if

(a) for some $f \in F_3^{i-1}$ and $p_1, p_2, p_3 \in T_{F^{i-1}, m+1}$, $f(p_1, p_2, p_3) \in \mathrm{sub}(r_{i-1})$

and (11)

(b) there exists exactly one $j \in [3]$ with $n_j > 0$ where n_j is the same as in the definition of property (10).

Lemma 11. There exist no $i \in [2k]$ such that r_{i-1} has property (11).

Proof. Since r_{2k} does not have property (11), we can use the same technique as in the proof of Lemma 10. Assume that r_{i-1} has property (11). Then, using the notations of Lemma 10, we again have (a)—(d) as in Lemma 10 and, without loss of generality, may suppose that $|\alpha_1|_2>0$, $|\alpha_2|_2=|\alpha_3|_2=0$ or, in other words, $p_{i_2}, p_{i_3}\in T_{F_1^{i-1},m+1}$. Hence, from Lemmas 1 and 9 it follows that $|\beta_1|_2>0$ and $q_2, q_3\in T_{F_1^i,m+2}$ meaning that r_i has property (11). □

We shall need one further property.

Definition. We say that for some $i\in[2k]$, r_{i-1} has property (12) if there exist $\alpha, \beta\in\text{path}(r_{i-1})$ satisfying the following conditions:

(a) $\alpha \not\leqslant \beta$ and $\beta \not\leqslant \alpha$,

(b) $\text{str}(r_{i-1}, \alpha) = f(p_1, p_2, p_3)$ for some $f\in F_3^{i-1}$,

$$p_1, p_2, p_3\in T_{F^{i-1},m+1}, \tag{12}$$

(c) $\text{str}(r_{i-1}, \beta) = f'(p_1', p_2', p_3')$ for some $f'\in F_3^{i-1}$,

$$p_1', p_2', p_3'\in T_{F^{i-1},m+1}.$$

Lemma 12. There exist no $i\in[2k]$ for which r_{i-1} has property (12).

Proof. If r_{i-1} has property (12) then, by Lemma 9, so does r_i. This proves our lemma since r_{2k} does not have property (12). □

Lemma 13. Let $i\in[2k]$ be an odd integer. Then $\text{rn}_3(r_{i-1})=\text{rn}_3(r_i)$.

Proof. It obviously follows from Lemma 9 that $\text{rn}_3(r_{i-1})\leqq\text{rn}_3(r_i)$. Let us assume that $\text{rn}_3(r_{i-1})<\text{rn}_3(r_i)$. Then in the derivation $a_i r_{i-1}\overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r_i' \left(r_i'\in T_{F^i}(A_i X_{m+1})\right)$ it has to be applied at least one rule of the form (9). However, this is impossible since, for odd i, $\mathfrak{A}_i$ is an LNDR transducer. □

Lemma 14. $k=m$.

Proof. On the contrary, assume that $k<m$. Then, since $\text{rn}_3(r_0)=0$ and $\text{rn}_3(r_{2k})=m$, it follows from Lemma 13 that for some even integer $i\in[2k]$, $\text{rn}_3(r_{i-1})\leqq$ $\leqq\text{rn}_3(r_i)-2$. It means that there exist $\alpha, \beta\in\text{path}(r_{i-1})$ such that $\alpha\neq\beta$, $\text{str}(r_{i-1}, \alpha)=$ $=f(p_1, p_2)$, $\text{str}(r_{i-1}, \beta)=f'(p_1', p_2')$ for some $f, f'\in F_2^{i-1}$, $p_j, p_j'\in T_{F^{i-1},m+1}$ $(j\in[2])$ moreover, in the derivation $a_i r_{i-1}\overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r_i' \left(r_i'\in T_{F^i}(A_i X_{m+1})\right)$ both f and f' were rewritten by applying a rule of the form (9).

First we claim that either $\alpha<\beta$ or $\beta<\alpha$. Really, from $\alpha\not<\beta$, $\beta\not<\alpha$ and $\alpha\neq\beta$ it would follow that r_i has property (12) contradicting Lemma 12.

Suppose that $\alpha<\beta$ and that the rule (9) was applied to rewrite f in the derivation $a_i r_{i-1}\overset{*}{\underset{\mathfrak{A}_i}{\Rightarrow}} r_i'$. Then, without loss of generality, we may assume that the following relations hold for some suitable $s_{i-1}, t_{i-1}\in T_{F^{i-1},m+2}$, $s_i\in T_{F^i,m+2}$ and $q_1, q_2, q_3\in T_{F^i,m+1}$:

(a) $r_{i-1} = s_{i-1} \cdot_{m+2} f(p_1, p_2)$,

(b) $p_1 = t_{i-1} \cdot_{m+2} f'(p_1', p_2')$,

(c) $r_i = s_i \cdot_{m+2} u_0(h(u_1(q_1), u_2(q_2), u_3(q_3)))$,

(d) $(s_{i-1}, s_i) \in \tau'_{\mathfrak{A}_i}$, $(p_{i_j}, q_j) \in \tau'_{\mathfrak{A}_i(c_j)}$ for each $j \in [3]$.

Let us introduce the following notations:

$$m_j = \max \{|\alpha|_2 | \alpha \in \text{path}_l(p_j) \text{ for some } l \in [m+1]\} \quad (j \in [2]),$$

$$n_j = \max \{|\alpha|_2 | \alpha \in \text{path}_l(q_j) \text{ for some } l \in [m+1]\} \quad (j \in [3]).$$

We know, by (b), that $m_1 > 0$. Moreover $m_2 = 0$ since from $m_2 > 0$ it would follow, by (d) and Lemma 1, that $n_1 > 0$, $n_2 > 0$ and $n_3 > 0$. This, however would mean that r_i has property (10) which is impossible by Lemma 10. Hence we have $p_2 \in T_{F_1^{k-1}, m+1}$.

We also know, by (9), that $\{i_1, i_2, i_3\} = [2]$ which means that $\mathfrak{A}_i$ duplicates either p_1 or p_2. We show that both cases are impossible.

First let us suppose that 1 appears once and 2 appears twice in the sequence i_1, i_2, i_3. Then we obtain, by Lemmas 1 and 9, that for exactly one $j \in [3]$, $n_j > 0$, contradicting Lemma 11.

Next assume that 1 appears twice and 2 appears once in the sequence i_1, i_2, i_3. But then, since f' was also rewritten by a rule of type (9) we have that r_i has property (12) yielding again a contradiction, by Lemma 12.

Hence we have $k = m$. □

With this we also completed the proof of Theorem 3. □

Now we present Theorem 3 in an alternative form. It is not difficult to see that $\mathcal{LNDR} \circ \mathcal{NH} = \mathcal{UNDR}$. Really, for any LNDR transducer $\mathfrak{A}$ and NH transducer $\mathfrak{B}$, by Lemma 3 of [1], $\tau_{\mathfrak{A} \circ \mathfrak{B}} = \tau_{\mathfrak{A}} \circ \tau_{\mathfrak{B}}$ and it can easily be verified that in this case $\mathfrak{A} \circ \mathfrak{B}$ is a UNDR transducer. Conversely, given a UNDR transducer $\mathfrak{L}$, with the help of the usual relabeling technique (see, for example, Lemma 3.1 in [2], pp. 155) we can construct an LNDR transducer $\mathfrak{A}$ and an NH transducer $\mathfrak{B}$ with $\tau_{\mathfrak{L}} = \tau_{\mathfrak{A}} \circ \tau_{\mathfrak{B}}$. Thus Theorem 3 can be given in the following form as well.

Theorem 15. For any $m \geqq 1$, $\mathcal{UNDR}^m \subset \mathcal{UNDR}^{m+1}$.

The first problem, presented at the beginning of this section is answered by

Theorem 16. $[S]$ is an infinite set.

Proof. Immediately follows from Theorem 3. □

Now we deal with the second problem. The following lemma can be proved.

Lemma 17. There exists an NDR transducer $\mathfrak{A} = (F, \{a, b\}, F', P, a)$ such that $\tau_{\mathfrak{A}} \notin (\mathcal{LNDR} \circ \mathcal{NH})^k$ for any $k \geqq 1$.

Proof. Let $\mathfrak{A}$ be determined by the following conditions:

(a) $F = F_0 \cup F_2$, $F_0 = \{\#\}$, $F_2 = \{f_2\}$,

(b) $F' = F'_0 \cup F'_2 \cup F'_3$, $F'_0 = \{\#\}$, $F'_2 = \{f_2\}$, $F'_3 = \{g_3\}$,

(c) P is the set of the rules:

(i) $a\# \to \#$, $b\# \to \#$,

(ii) $af_2(x_1, x_2) \to g_3(bx_1, ax_1, bx_2)$, $bf_2(x_1, x_2) \to f_2(bx_1, bx_2)$.

Let P_m and Q_m be defined in the same way as in the proof of Theorem 3. It is not difficult to verify that for each $m \geqq 1$, $(P_m, Q_m) \in \tau'_{\mathfrak{A}}$.

Moreover, let Q_m be written in the form

$$\bar{Q}_m(x_1, x_2, x_2, \ldots, \overbrace{x_{m+1}, \ldots, x_{m+1}}^{m+1 \text{ times}}) \quad \text{where} \quad \bar{Q}_m \in \hat{T}_{F,n} \quad (n = 1+2+\ldots+m+1).$$

Then we can say that for any $t_1, \ldots, t_{m+1} \in T_F$

$$(P_m(t_1, \ldots, t_{m+1}),\ \bar{Q}_m(t_1, t_2, t_2, \ldots, \overbrace{t_{m+1}, \ldots, t_{m+1}}^{m \text{ times}}, t'_{m+1})) \in \tau_{\mathfrak{A}}$$

holds where $t'_{m+1} = \tau_{\mathfrak{A}}(t_{m+1})$. Using this notation, the following lemma can be proved in a similar way as Lemma 4. Therefore we omit the proof.

Lemma 18. If $\mathfrak{L}$ is an NDR transducer with $\tau_{\mathfrak{A}} = \tau_{\mathfrak{L}}$ then for each $m \geqq 1$ $(P_m, Q_m) \in \tau'_{\mathfrak{L}}$ holds. □

Now we can complete the proof of Lemma 17. Suppose that

$$\tau_{\mathfrak{A}} \in (\mathscr{LNDR} \circ \mathscr{NH})^k$$

for some $k \geqq 1$. Then for each $i \in [2k]$ there exists a DR transducer $\mathfrak{A}_i = (F^{i-1}, A_i, F^i, P_i, a_i)$ with properties (a)—(c) of (6) and $\tau_{\mathfrak{A}} = \tau_{\mathfrak{A}_1} \circ \ldots \circ \tau_{\mathfrak{A}_{2k}}$. Let m be chosen such that $k < m$. It follows from Lemmas 2 and 18 that $(P_m, Q_m) \in \tau'_{\mathfrak{A}_1} \circ \ldots \circ \tau'_{\mathfrak{A}_{2k}}$. However, if we follow the proof of Theorem 3 from (7) then we see in Lemma 14 that this is a contradiction. This ends the proof of Lemma 17. □

The last theorem is an immediate consequence of Lemma 17.

Theorem 19. $\bigcup_{k=1}^{\infty} (\mathscr{LNDR} \circ \mathscr{NH})^k \subset \mathscr{NDR}$.

RESEARCH GROUP ON THEORY OF AUTOMATA
HUNGARIAN ACADEMY OF SCIENCES
SOMOGYI B. U. 7.
SZEGED, HUNGARY
H—6720

References

[1] FÜLÖP, Z. and S. VÁGVÖLGYI, Results on compositions of deterministic root-to-frontier tree transformations, Acta Cybernetica, v. 8, f. 1, 1987, 49—61.

[2] GÉCSEG, F. and M. STEINBY, Tree Automata, Akadémiai Kiadó, Budapest, 1984.

(Received Sept. 28, 1986)

Evaluated grammars

ALEXANDR MEDUNA

1. Introduction

Mechanisms which regulate the application of the rules belong to the most important devices in order to enlarge the generative capacity of context free grammars. A common idea is that not every derivation leading from the start symbol to a terminal word is acceptable, but there is a control device which lets through acceptable derivations only. For instance, an application of some production determines which productions are applicable in the next step (this is called a programmed grammar), or some productions can never be applied if any other applicable (an ordered grammar). In a matrix grammar one has to apply only certain previously specified strings of productions or, more generally, the string of productions corresponding to a derivation must belong to a set of previously specified strings (a grammar with a control set) — see [3].

In this paper, the notion of evaluated grammar is introduced. The derivation process in this generative mechanism is regulated by a certain evaluation of some symbols occurring in sentential forms.

We believe that the introduction of the new type of grammar with a restriction in derivation introduced here is very useful because of three reasons:

(i) evaluated grammars represent a simple and very natural extension of context-free grammars;

(ii) evaluated grammars are considerably more powerful than context-free grammars;

(iii) some classes of languages generated by parallel rewriting systems (e.g. EOL languages) can be characterized by evaluated grammars in a natural way.

2. Preliminaries

We introduce here only briefly the notions needed in this paper. For a more detailed discussion, as well as for background material and motivation, the reader is referred to [2, 3].

Let α be a word over an alphabet Σ. The alphabet of α, alph α, is the set of all symbols (from Σ) that appear at least once in α.

A context free grammar is quadruple $G=(\Sigma, P, S, \Delta)$ where, as usual, Σ is a finite alphabet, $\Delta \subseteq \Sigma$ is the terminal alphabet and $\Sigma \setminus \Delta$ is the nonterminal alphabet, $P \subseteq (\Sigma \setminus \Delta) \times \Sigma^*$ is a finite set of productions, where a production (A, α) is usually written as $A \to \alpha$, and S in $\Sigma \setminus \Delta$ is the start symbol. For arbitrary words $x, y \in \Sigma^*$ and production $A \to \alpha$ we write $xAy \Rightarrow x\alpha y$, and denote the reflexive, transitive closure of $\Rightarrow$ by $\Rightarrow^*$. The language generated by G, denoted $L(G)$, is defined by $L(G)=\{x \in \Delta^*: S \Rightarrow^* x\}$.

A context free grammar $G=(\Sigma, P, S, \Delta)$ is called regular if every production $A \to \alpha$ from P satisfies $\alpha \in \Delta(\Sigma \setminus \Delta) \cup \Delta$.

An ETOL system G consits of $m+3 (m \geq 1)$ components $G=(\Sigma, P_1, \ldots, P_m, S, \Delta)$ where Σ, Δ, S are defined identically as for context free grammars, and where every P_i is a finite subset of $\Sigma \times \Sigma^*$ such that for every $a \in \Sigma$ at least one pair (a, α) occurs in P_i. The pairs in P_i are again called productions and usually written as $a \to \alpha$. For an arbitrary word $x=a_1 a_2 \ldots a_k$, $a_i \in \Sigma$, and productions $a_1 \to \alpha_1, \ldots, a_k \to \alpha_k$ of the same set P_j we write $a_1 a_2 \ldots a_k \Rightarrow \alpha_1 \alpha_2 \ldots \alpha_k$ and denote the reflexive, transitive closure of $\Rightarrow$ by $\Rightarrow^*$. The language generated by G, $L(G)$, is defined by $L(G)=\{x \in \Delta^*: S \Rightarrow^* x\}$. An ETOL system with a single set of productions is called EOL system.

For $n>0$, an n-parallel right linear grammar (see [1]) is a quintuple $G=(\Sigma, P, S, \Delta, n)$ where Σ, Δ, S are defined identically as for context free grammars, and $P \subseteq (\Sigma \setminus (\Delta \cup \{S\})) \times (\Delta^* (\Sigma \setminus (\Delta \cup \{S\}))) \cup \Delta^+) \cup \{S\} \times (\Delta^* \cup (\Sigma \setminus (\Delta \cup \{S\}))^n)$ is a finite set of rules, where a production (A, α) is usually written as $A \to \alpha$. The yield relation is defined as follows: for $x, y \in \Sigma^*$, $x \Rightarrow y$ if and only if either $x=S$ and $S \to y \in P$ or $x=y_1 X_1 \ldots y_n X_n$ and $y=y_1 x_1 \ldots y_n x_n$, where $y_i \in \Delta^*$, $x_i \in \Delta^* (\Sigma \setminus (\{S\} \cup \Delta)) \cup \Delta^+$, $X_i \in \Sigma \setminus \Delta$ and $X_i \to x_i \in P$, $1 \leq i \leq n$. The relation $\Rightarrow$ can be extended to give $\Rightarrow^*$ as above. The notion of the language generated by G can be introduced just as for context free grammars. (An n-parallel right linear grammar G is in normal form if

$$G = (\Delta \cup K_1 \cup \ldots \cup K_n \cup \{S\}, P, S, \Delta, n),$$

S is not in $\Delta \cup K_1 \cup \ldots \cup K_n$, K_i are mutually disjoint nonterminal sets, if $S \to X_1 \ldots X_n \in P$ and $X_1 \ldots X_n \in (K_1 \cup \ldots \cup K_n)^*$ then $X_i \in K_i$, $1 \leq i \leq n$, and if $X_i \to y Y_j \in P$, $X_i \in K_i$ and $Y_j \in K_j$ then $i=j$.)

The families of languages generated by context free, regular and n-parallel right linear grammars are denoted by $\mathscr{L}(\mathsf{CF})$, $\mathscr{L}(\mathsf{RG})$ and $\mathscr{L}(n-\mathsf{PRL})$, respectively. Let $\mathscr{L}(\mathsf{PRL}) = \bigcup_{i=1}^{\infty} \mathscr{L}(i-\mathsf{PRL})$. Families of languages generated by ETOL systems, ETOL systems of finite index (see [2]) and EOL systems are denoted by $\mathscr{L}(\mathsf{ETOL})$, $\mathscr{L}_{\mathsf{FIN}}(\mathsf{ETOL})$ and $\mathscr{L}(\mathsf{EOL})$, respectively.

3. Definition of evaluated grammars

Intuitively, an evaluated grammar is very much like a context free grammar. However, some symbols (including terminals) in a given sentential form of an evaluated grammar can have a certain value associated (a non-negative integer). In

one derivation step either a nonterminal without associated value in a usual "context free" way or a nonterminal with the least value that occurs in the given sentential form is rewritten. In the latter case the nonterminal is rewritten again in a usual way but, in addition new evaluation is assigned to apriori specified (on the right side of the rule applied) symbols.

Formally, let N be the set of non-negative integers and let Σ be an alphabet. We denote members of $V=\Sigma\times N$ by $a_{(i)}$ where a is in Σ and i is in N, then in a natural way we can define V^*. Define the letter-to-letter homomorphism $v: (V\cup\Sigma)^*\to\Sigma^*$ by $v(a_{(i)})=a$ for all $a_{(i)}$ in V and $v(a)=a$ for all $a\in\Sigma$. An evaluated grammar, EG, is a construct $G=(\Sigma, P, S, \Delta)$ where $\Delta\subseteq\Sigma$ is a terminal alphabet, $P\subseteq(\Sigma\setminus\Delta)\times(V\cup\Sigma)^*$ is a finite set of productions, where $(A,\alpha)\in P$ is usually written as $A\to\alpha$, and $S\in\Sigma\setminus\Delta$ is a start symbol. For all $\alpha,\beta\in(V\cup\Sigma)^*$ we write $\alpha\underset{G}{\Rightarrow}\beta$ (or simply $\alpha\Rightarrow\beta$ if G is understood) if $\beta=\alpha_1\gamma\alpha_2$ for some $\alpha_1,\alpha_2\in(V\cup\Sigma)^*$ and either $\gamma\in\Sigma^*$, $\alpha=\alpha_1 A\alpha_2$ and $A\to\gamma\in P$ or $\alpha=\alpha_1 A_{(i)}\alpha_2$ for some $A_{(i)}\in V$,

$$\gamma=\delta_0 B_{1(i+k_1)}\delta_1\ldots B_{n(i+k_n)}\delta_n, A\to\delta_0 B_{1(k_1)}\delta_1\ldots B_{n(k_n)}\delta_n\in P$$

where $\delta_j\in\Sigma^*$, $B_{j(k)_j}$, $B_{j(i+k_j)}\in V$ (i.e. $B_j\in\Sigma$, $i, k_j\in N$), $0\leqq j\leqq n$ for some $n\geqq 0$ (where $n=0$ implies $\gamma=\delta_0$ and $A\to\delta_0\in P$) and $i\leqq m$ for every $X_{(m)}\in\text{alph}\,\alpha_1\alpha_2\cap V$. The language generated by G, $L(G)$, is defined by $L(G)=\{v(x): v(x)\in\Delta^*$ and $S_{(0)}\Rightarrow^* x\}$ where $\Rightarrow^*$ is the transitive,reflexive closure of $\Rightarrow$.

Now we introduce some special cases of evaluated grammars. Let $G=(\Sigma, P\ S, \Delta)$ be an EG and let n be a positive integer. We say that G is n-regular if it has the following properties:

(1) if $S\to\alpha\in P$ then $v(\alpha)=X_1\ldots X_n$
with $X_i\in\Sigma\setminus(\Delta\cup\{S\})$, $1\leqq i\leqq n$;

(2) if $A\to\alpha\in P$ and $A\neq S$ then
$v(\alpha)\in\Delta(\Sigma\setminus(\Delta\cup\{S\}))\cup\Delta$.

We say that the EG G is regular, RGEG, if it is n-regular for some positive integer n.

We say that an EG G is binary, BEG, if

$$P\subseteq(\Sigma\setminus\Delta)\times((\Delta\times\{0\})\cup((\Sigma\setminus\Delta)\times\{1\})\cup\Delta)^*.$$

A binary regular EG (i.e. an EG which is as regular, as binary) will be denoted by BRGEG.

We use $\mathscr{L}$(E), $\mathscr{L}$(RGE), $\mathscr{L}$(BE) and $\mathscr{L}$(BRGE) to denote the families of languages generated by evaluated, regular evaluated, binary evaluated and binary regular evaluated grammars, respectively.

4. Examples

We now consider some examples to give insight into evaluated grammars. We usually define evaluated grammars by simply listing their productions in Backus-Naur form. In this case we use S to denote the start symbol, early upper case Roman letters to denote nonterminals and early lower case Roman letters to denote terminals.

Example 1. Let

$$G_1\colon\ S \to A_{(1)}B_{(1)}C_{(1)};\quad A \to aA_{(1)}|a_{(0)};$$

$$B \to bB_{(1)}|b_{(0)};\qquad C \to cC_{(1)}|c_{(0)}$$

be a BRGEG. Then, e.g., for the word *aabbcc* there exists a derivation in G:

$$S_{(0)} \Rightarrow A_{(1)}B_{(1)}C_{(1)} \Rightarrow aA_{(2)}B_{(1)}C_{(1)} \Rightarrow aA_{(2)}bB_{(2)}C_{(1)} \Rightarrow$$

$$\Rightarrow aA_{(2)}bB_{(2)}cC_{(2)} \Rightarrow aA_{(2)}bb_{(2)}cC_{(2)} \Rightarrow$$

$$\Rightarrow aA_{(2)}bb_{(2)}cc_{(2)} \Rightarrow aa_{(2)}bb_{(2)}cc_{(2)}$$

and thus we get

$$v(aa_{(2)}bb_{(2)}cc_{(2)}) = aabbcc.$$

Clearly, G generates a well-known context-sensitive language:

$$L(G_1) = \{a^n b^n c^n\colon\ n \geqq 1\}.$$

Example 2. Consider the RGEG

$$G_2\colon\ S \to A_{(0)}B_{(1)}A_{(0)};\quad A \to aA_{(2)}|a_{(1)};$$

$$B \to bB_{(2)}|b_{(0)}C|b_{(0)};\quad C \to bc|b.$$

The reader can easily check that

$$L(G_2) = \{a^k b^l a^k\colon\ l \geqq k \geqq 1\}.$$

It is well-known (see [2]) that $L(G_2)$ is not an EOL language.

Example 3. Let

$$G_3\colon\ S \to S_{(1)}S_{(1)}|a_{(0)}$$

$$L(G_3) = \{a^{2^n}\colon\ n \geqq 1\}$$

be a BEG. It is not difficult to show that which is not an ETOL language of finite index (see [2]).

5. Generating power of evaluated grammars

From the definition of an EG it is easy to see that every context free language can be obtained as the language of some EG. Moreover, from examples in the previous section it follows that the class of context free languages is properly contained in $\mathscr{L}$(E). The purpose of this section is to show that $\mathscr{L}$(E) is included in $\mathscr{L}$(ETOL).

Theorem 1. $\mathscr{L}(\mathrm{E}) \subseteq \mathscr{L}(\mathrm{ETOL})$.

Proof. Let

$$G = (\Sigma, P, S, \Delta)$$

be an EG and let k be an arbitrary but fixed non-negative integer such that for every production $A \to \alpha_1 B_{(i)} \alpha_2 \in P$, where $A \in \Sigma \setminus \Delta_1$, $\alpha_1 \alpha_2 \in (V \cup \Sigma)^*$, $B_{(i)} \in V$, it holds that $i \leqq k$. Consider a new alphabet $\bar{\Sigma} = \{[A, i]\colon A \in \Sigma \text{ and } 0 \leqq i \leqq k\}$ and let F be a new "block" symbol. Let $\bar{\Delta} = \{[a, i]\colon a \in \Delta \text{ and } 0 \leqq i \leqq k\}$; clearly $\bar{\Delta} \subseteq \bar{\Sigma}$. Now we define four new tables of productions P_i, $1 \leqq i \leqq 4$, as follows:

$$P_1 = \{[A, 0] \to x_0[B_1, k_1]x_1 \ldots [B_n, k_n]x_n\colon$$

$$A \to x_0 B_{1(k_1)} x_1 \ldots B_{n(k_n)} x_n \in P,$$

$$A \in \Sigma \setminus \Delta, \quad x_j \in \Sigma^*, \quad B_{j(k_j)} \in V,$$

$$0 \leqq j \leqq n \text{ for some } n \geqq 0\} \cup$$

$$\cup \{X \to X\colon X \in \Sigma \cup \{F\} \cup \{[a, 0]\colon a \in \Delta\} \cup \{[A, i]\colon A \in \Sigma \text{ and } 1 \leqq i \leqq k\}\} \cup$$

$$\cup \{[A, 0] \to F\colon A \in \Sigma \setminus \Delta\};$$

$$P_2 = \{[A, i] \to [A, i-1]\colon A \in \Sigma \text{ and } 1 \leqq i \leqq k\} \cup$$

$$\cup \{[A, 0] \to F\colon A \in \Sigma\} \cup \{X \to X\colon X \in \Sigma \cup \{F\}\};$$

$$P_3 = \{A \to \alpha\colon A \to \alpha \in P \text{ and } \alpha \in \Sigma^*\} \cup \{X \to X\colon X \in \Sigma \cup \{F\} \cup \bar{\Sigma}\}$$

and

$$P_4 = \{X \to F\colon X \in \Sigma \setminus \Delta \cup \bar{\Sigma} \setminus \bar{\Delta} \cup \{F\}\} \cup \{a \to a\colon a \in \Delta\} \cup$$

$$\cup \{[a, i] \to a\colon [a, i] \in \bar{\Delta} (a \in \Delta,\ 0 \leqq i \leqq k)\}.$$

Consider the ETOL system

$$\bar{G} = (\bar{\Sigma} \cup \Sigma \cup \{F\}, P_1, P_2, P_3, P_4, [S, 0], J, \Delta).$$

From the construction it is clear that $L(G) = L(\bar{G})$; hence the theorem holds. □

In the end we want to mention that it is not known whether or not the inclusion $\mathscr{L}(\mathsf{E}) \subseteq \mathscr{L}(\mathsf{ETOL})$ is proper.

6. Subfamilies of $\mathscr{L}(\mathsf{E})$

In this section we prove a few results about some special cases of evaluated grammars which were defined in Section 3.

Theorem 2. $\mathscr{L}(\mathsf{EOL}) = \mathscr{L}(\mathsf{BE})$.

Proof. 1. $\mathscr{L}(\mathsf{EOL}) \subseteq \mathscr{L}(\mathsf{BE})$: Let

$$G = (\Sigma, P, S, \Delta)$$

be an EOL system. Define a new alphabet $\bar{\Delta}=\{\bar{a}\colon a\in\Delta\}$ and a coding

$$h:\Sigma^*\to((\Sigma\setminus\Delta\times\{1\})\cup(\bar{\Delta}\times\{1\}))^* \quad \text{defined by}$$

(i) $h(X)=X_{(1)}$ for all $X\in\Sigma\setminus\Delta$;

(ii) $h(X)=\bar{X}_{(1)}$ for all $X\in\Delta$.

We define a new set of productions

$$\bar{P}=\{h(A)\to h(\alpha)\colon A\to\alpha\in P\}\cup\{\bar{a}_{(1)}\to a_{(0)}\colon a\in\Delta(\bar{a}\in\bar{\Delta})\}.$$

Consider the BEG

$$\bar{G}=(\Sigma\cup\bar{\Delta},\bar{P},S,\Delta).$$

Clearly $L(G)=L(\bar{G})$ and thus $\mathscr{L}(\mathsf{EOL})\subseteq\mathscr{L}(\mathsf{BV})$.

2. $\mathscr{L}(\mathsf{EOL})\supseteq\mathscr{L}(\mathsf{BE})$: Let

$$G=(\Sigma,P,S,\Delta)$$

be a BEG and, clearly, we may assume without loss of generality that every nonterminal in G is useful i.e. that for every $A\in\Sigma\setminus\Delta$ there exists a word $\alpha\in(((\Sigma\setminus\Delta)\times\{1\})\cup(\Delta\times\{0\})\cup\Delta)^*$ such that $A\to\alpha\in P$. Define a new alphabet $\Sigma'=\{A'\colon A\in\Sigma\}$ and a new "block" symbol F. We define the substitution

$$g\colon(\Delta\cup((\Sigma\setminus\Delta)\times\{1\})\cup(\Delta\times\{0\}))^*\to(\Delta\cup\Sigma')^*$$

by

(i) $g(a)=\{a',a\}$ for all $a\in\Delta$;

(ii) $g(A_{(1)})=A'$ for all $A_{(1)}\in(\Sigma\setminus\Delta)\times\{1\}$;

(iii) $g(a_{(0)})=a$ for all $a_{(0)}\in\Delta\times\{0\}$.

Let

$$P'=\{A'\to g(\alpha)\colon A\to\alpha\in P\}\cup\cup\{a'\to a',a'\to a,a\to F\colon a\in\Delta\}\cup\cup\{F\to F\}.$$

Now, let

$$G'=(\Sigma'\cup\Delta\cup\{F\},P',S',\Delta)$$

be an EOL system, then, clearly, $L(G)=L(G')$ and thus $\mathscr{L}(\mathsf{EOL})\supseteq\mathscr{L}(\mathsf{BE})$. Hence, we have $\mathscr{L}(\mathsf{EOL})=\mathscr{L}(\mathsf{BE})$ and the theorem holds. □

From this proof we obtain:

Corollary 1. For every $L\in\mathscr{L}(\mathsf{BE})$ there exists a BEG $G=(\Sigma,P,S,\Delta)$ such that $L(G)=L$ and $P\subseteq(\Sigma\setminus\Delta)\times((\Delta\times\{0\})\cup((\Sigma\setminus\Delta)\times\{1\}))^*$.

It is a straightforward to prove the following four lemmas.

Lemma 1. $\mathscr{L}(\mathsf{BE})\subset\mathscr{L}(\mathsf{E})$.

Proof. The inclusion $\mathscr{L}(\mathsf{BE})\subseteq\mathscr{L}(\mathsf{E})$ is an immediate consequence of the definitions of BEG and EG. That the inclusion is strict follows from Example 2 in Section 4 and Theorem 2. □

Lemma 2. $\mathscr{L}(\mathsf{BE}) \not\subset \mathscr{L}(\mathsf{RGE})$.

Proof. From the construction in the proof of Theorem 1 follows that $\mathscr{L}(\mathsf{RGE}) \subseteq \subseteq \mathscr{L}_{\mathsf{FIN}}(\mathsf{ETOL})$. On the other hand, $L(G_3) \in \mathscr{L}(\mathsf{BE}) / \mathscr{L}_{\mathsf{FIN}}(\mathsf{ETOL})$; see Example 3 in Sect. 4. Hence, the lemma holds. □

Lemma 3. $\mathscr{L}(\mathsf{RGE}) \not\subset \mathscr{L}(\mathsf{BE})$.

Proof. It is an immediate consequence of Example 2 in Sect. 4 and Theorem 2. □

Lemma 4. $\mathscr{L}(\mathsf{RGE}) \subset \mathscr{L}(\mathsf{E})$.

Proof. Clear. □

It is quite clear that $\mathscr{L}(\mathsf{RG}) \subset \mathscr{L}(\mathsf{BRGE})$; see the definition and Example 1 in Sect. 4. We now prove this result:

Lemma 5. $\mathscr{L}(\mathsf{BRGE}) \subseteq \mathscr{L}(\mathsf{PRL})$.

Proof. Let

$$G = (\Sigma, P, S, \Delta)$$

be n-regular BRGEG for some $n \geqq 1$.

From the definition of BRGEG it follows that in any sentential form (except the last one) of a derivation of any word from $L(G)$, no symbol $a_{(0)}$, $a \in \Delta \times \{0\}$, is contained. Thus, we can construct the following $n-$parallel right linear grammar:

$$\bar{G} = (\Sigma \cup \bar{\Delta}, \bar{P}, S, \Delta)$$

where

$$\bar{\Delta} = \{\bar{a} \colon a \in \Delta\}$$

and

$$\begin{aligned}\bar{P} = {} & \{S \to X_1 \dots X_n \colon S \to X_{1_{(1)}} \dots X_{n_{(1)}} \in P,\ X_{i_{(1)}} \in \Sigma \setminus (\Delta \cup \{S\}) \times \{1\},\ 1 \leqq i \leqq n,\ n > 0\} \cup \\ & \cup \{A \to aB \colon A \to aB_{(1)} \in P,\ a \in \Delta,\ A \in \Sigma \setminus (\Delta \cup \{S\}),\ B_{(1)} \in ((\Sigma \setminus (\Delta \cup \{S\})) \times \{1\})\} \cup \\ & \cup \{A \to a \colon A \to a_{(0)} \in P,\ A \in \Sigma \setminus (\Delta \cup \{S\}),\ a \in (\Delta \times \{0\})\} \cup \\ & \cup \{A \to a,\ A \to \bar{a},\ \bar{a} \to \bar{a},\ \bar{a} \to a \colon A \to a \in P,\ A \in \Sigma \setminus (\Delta \cup \{S\}),\ a \in \Delta\}.\end{aligned}$$

Clearly $L(G) = L(\bar{G})$. The lemma is proved. □

Lemma 6. $\mathscr{L}(\mathsf{CF})$ and $\mathscr{L}(\mathsf{BRGE})$ are incomparable but not disjoint.

Proof. The lemma is a direct consequence of Example 1 (see Sect. 4), Lemma 5 and a diagram from Sect. 6 in [1]. □

Lemma 7. $\mathscr{L}(\mathsf{CF}) \not\subset \mathscr{L}(\mathsf{RGE})$.

Proof. By proof of Lemma 2, $\mathscr{L}(\mathsf{RGE}) \subseteq \mathscr{L}_{\mathsf{FIN}}(\mathsf{ETOL})$. But it is well-known that $\mathscr{L}(\mathsf{CF}) \not\subset \mathscr{L}_{\mathsf{FIN}}(\mathsf{ETOL})$ (see, e.g., [2]). Thus the lemma holds. □

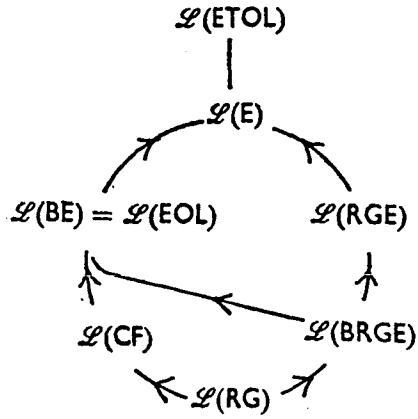
Lemma 8. $\mathscr{L}(\mathsf{BRGE}) \subseteq \mathscr{L}(\mathsf{BE}) \cap \mathscr{L}(\mathsf{RGE})$.

Proof. From definition. □

7. The relationship diagram

The aim of this section is to establish the relationship diagram among various classes of languages considered in this paper. We get the following theorem.

Theorem 3.



(If there is a directed chain of edges in the diagram leading from a class X to a class Y then $X \subset Y$, an undirected chain means that we do not know whether the inclusion is proper. Otherwise X and Y are incomparable but not disjoint.)

Proof. From the results of Sect. 5 and 6 together with the fact that $\mathscr{L}(\mathrm{RG}) \subset \subset \mathscr{L}(\mathrm{CF}) \subset \mathscr{L}(\mathrm{EOL})$ — see [2]. □

Abstract

Evaluated grammars are based on context free grammars but the derivation process in these grammars is regulated by a certain evaluation of some symbols occurring in their sentential forms.

Fundamental properties of the family of languages generated by evaluated grammars are investigated. This family of languages is contained in the family of ETOL languages and properly contains the family of EOL languages.

In addition, we propose and study some special cases of evaluated grammars.

COMPUTING CENTRE
TECHNICAL UNIVERSITY BRNO
OBRÁNCŮ MÍRU 21
BRNO, CZECHOSLOVAKIA

References

[1] ROSEBRUGH, R. D., WOOD, D., Restricted parallelism and right linear grammars. Utilitas Mathematica 7, 151—186, 1975.

[2] ROZENBERG, G., SALOMAA, A., The Mathematical Theory of L Systems, New York: Academic Press, 1980.

[3] SALOMAA, A., Formal Languages, New York: Academic Press, 1973.

(Received Aug. 12, 1986)

EBE: a language for specifying the expected behavior of programs during debugging

NGUYEN HUU CHIEN

1. Introduction

In [1] Bruegge B. and Hibbard P. used *GPE*s (Generalized Path Expression) for specifying expected behavior of programs. *GPE*s are slightly extended version of a *BPE* (Basic Path Expression) with predicates and counters.

A *BPE* is a regular expression with operators sequencing(;), exclusive selection(+) and repetition(*). The operands, called *PF*s (Path Function), are the names of statements or groups of statements defined in the source program. For each *PF* two counters are defined: the counters *ACT* and *TERM*. These represent the activation and termination number of a *PF* respectively. Predicate is a logical expression involving the counters and the variables of the program and debugger. *BPE* is extended by associating predicates with *PF*s.

In this paper we extended *GPE* by adding the operator shuffle (Δ). This does not increase the power of *GPE*s, but we can describe the expected behavior of a program in a simpler way. In the next sections we define the syntax and semantics of the extended *GPE*s, called *EBE*s (Expected Behavior Expression). The purpose of *EBE*s is to specify the order of execution of *PF*s, the semantics of *EBE*s therefore can be defined by specifying a set of actual behaviors that are valid with respect to a given *EBE*. In section IV we discuss some properties of *EBE*s. According to the syntax and semantics we introduce the syntactical and semantical equivalence of *EBE*s. A sufficient condition for the semantical equivalence of two *EBE*s is given. It is shown that the syntactical equivalence is more powerful than the semantical equivalence. It is also proved that *EBE*s are not more powerful than *GPE*s. In section V we present an implementation of *EBE*s. The implementation is formally defined omitting details of actual implementation, and then its semantics is also defined similarly to that of *EBE*s, that is, by specifying a set of actual behaviors that are valid with respect to a given implementation. Correctness of the implementation is proved by showing a given *EBE* and its implementation recognize the same set of actual behaviors.

In order to make an implementation effective it is necessary to reduce *EBE*s. We give some rules for reducing *EBE*s in section VI.

II. The syntax of EBEs

Assume that the notions ⟨*identifier*⟩, ⟨*integer number*⟩ and ⟨*arithmetic expression*⟩ are known. The other notions are defined in terms of the above ones.

⟨*path function*⟩::=⟨*procedure name*⟩
⟨*procedure name*⟩::=⟨*identifier*⟩
⟨*counter*⟩::=*ACT*(⟨*procedure name*⟩)|*TERM*(⟨*procedure name*⟩)
⟨*counter exp*⟩::=⟨*counter*⟩|⟨*integer variable*⟩|
⟨*integer constant*⟩|(⟨*counter exp*⟩)|
⟨*counter exp*⟩⟨*binary op*⟩⟨*counter exp*⟩
⟨*binary op*⟩::=+|−|×
⟨*integer variable*⟩::=⟨*identifier*⟩
⟨*integer constant*⟩::=⟨*integer number*⟩
⟨*counter rel*⟩::=⟨*counter exp*⟩⟨*rel*⟩⟨*counter exp*⟩
⟨*arithmetic rel*⟩::=⟨*arithmetic expression*⟩⟨*rel*⟩
⟨*arithmetic expression*⟩
⟨*rel*⟩::=<|>|=|≦|≧
⟨*predicate*⟩::=⟨*counter rel*⟩|⟨*arithmetic rel*⟩|(⟨*predicate*⟩)|
⟨*predicate*⟩⟨*logic op*⟩⟨*predicate*⟩|¬⟨*predicate*⟩
⟨*logic op*⟩::=∧|∨|→
⟨*operand*⟩::=⟨*path function*⟩|⟨*path function*⟩[⟨*predicate*⟩]
⟨*EBE*⟩::=⟨*operand*⟩|(⟨*EBE*⟩)|⟨*EBE*⟩;⟨*EBE*⟩|⟨*EBE*⟩+⟨*EBE*⟩|
⟨*EBE*⟩*|⟨*EBE*⟩Δ⟨*EBE*⟩

Let E be an *EBE*, we define the language $L(E)$ as follows:

If $E=o$, where o an operand, then $L(E)=\{o\}$. Let $L1=L(E1)$, $L2=L(E2)$, then

$$L(E1;E2)=L1\,L2,\ L(E1+E2)=L1+L2,\ L(E1*)=L1*,$$

$L(E1\,\Delta E2)=L1\,\Delta L2=\{o_1o_1'\ldots o_no_n'|o_1\ldots o_n\in L1$ and $o_1'\ldots o_n'\in L2$, it may happen that o_i and o_j' are $\varepsilon\}$.

Now we give some examples of *EBE*s.

Example.

Initstack; (*Push*[*TERM*(*Push*)−*TERM*(*Pop*)<*N*]+
Pop[*TERM*(*Push*)−*TERM*(*Pop*)>*o*]+
Top[*TERM*(*Push*)−*TERM*(*Pop*)>*o*])*.

This *EBE* specifies an expected behavior of the program which states the operational constraints on a bounded stack of length N: first the procedure Initstack has to be called. One of the following can then happen: either procedure *Push* can be called if the size of the stack is smaller than N, or *Top* or *Pop* can be called if the size of the stack is larger than o.

Example. The *EBE*

$$(p;\ q)\Delta(r;\ s)$$

is used to look for activation of the procedure p when p has been called 5 times and the value of the variable A is 4.

Example. The *EBE*

$$p; q \Delta r; s$$

permits possible sequences of the execution of the procedures p, q, r and s as follows:

$$pqrs, prqs, prsq, rpsq, rpqs, rspq.$$

III. The semantics of EBEs

First we define some notions.

Let OB be an arbitrary set (representing a set of all data objects), P a finite set of procedures, and $P' \subset P$.

A *state* is a pair $\langle S, cou \rangle$, where $S \subset OB$, and $cou = \{a_p, t_p | p \in P'\} \subset N+ = \{0, 1, 2, \ldots\}$ (the numbers a_p and t_p represent the activation and termination number of the procedure p), and the "*cou*" is called *counter-state*.

A *concrete (actual) event* is an activation of the procedure p at a state $\langle S, cou \rangle$. We denote it by $e_c = \langle p, S, cou \rangle$.

A *concrete behavior* B is a sequence of concrete events $e_c^1 \ldots e_c^n$. Let $\mathbf{B}$ be the *set* of all concrete behaviors.

A *computational system* is a 5-tuple $\langle OB, P, P', f_a, f_t \rangle$, where f_a and f_t are maps: $\mathbf{B} \to \{g | g \text{ is function, } g: P' \to N+\}$ which are defined as follows:

The definition of f_a: $f_a(\emptyset)(p) = 0$ for all $p \in P'$,

$$f_a(B\langle p, S, cou \rangle)(p') = f_a(B)(p) + 1 \quad \text{if} \quad p' = p$$
$$= f_a(B)(p') \quad \text{otherwise,} \quad p' \in P', \quad B \in \mathbf{B}.$$

The definition of f_t: $f_t(\emptyset)(p) = 0$ for all $p \in P'$,

$$f_t(B\langle p, S, cou \rangle)(p') = f_t(B)(p) + 1 \quad \text{if} \quad p' = p$$
$$= f_t(B)(p') \quad \text{otherwise,} \quad p' \in P', \quad B \in \mathbf{B} \quad (\emptyset \text{ is the empty sequence}).$$

Let E be an *EBE*, then

$$P_E = \{p | p \text{ is a path function in } E\},$$
$$V_E = \{v | v \text{ is variable in } E, \text{ and } v \neq ACT \text{ and } v \neq TERM\},$$
$$C_E = \{c | c \text{ is constant in } E\}, \text{ assume that } C_E \subset OB,$$
$$AT_E = \{ACT(p), TERM(p) | p \in P'\},$$
$$Q_E = \{q | q \text{ is predicate in } E\}.$$

An *abstract event* e_a is a 4-tuple $\langle p, q, V_E, AT_E \rangle$, where $p \in P_E$, $q \in Q_E$.

An *abstract event expression* Ea of E is an expression obtained as follows. All operands $p[q]$ or p in E are substituted by abstract events $e_a = \langle p, q, V_E, AT_E \rangle$ or $e_a = \langle p, true, V_E, AT_E \rangle$ respectively.

Let $e_c = \langle p, s, cou \rangle$, then the counter-state "*cou*" and the maps f_a and f_t *match* under a concrete behavior B, if $a_p = f_a(Be_c)(p)$, $a_{p'} = f_a(B)(p')$, $p' \in P_E \setminus \{p\}$, and $t_{p'} = f_t(B)(p')$, $p' \in P_E$. This fact is denoted by $Matchs(cou, f_a, f_t, B)$.

An *Interpretation* is a function I: $V_E \cup C_E \cup AT_E \to OB \cup N+$ such that $I(v) \in OB$ for $v \in V_E$, $I(c) \in OB$ for $c \in C_E$, $I(v) \in N+$ for $v \in AT_E$ and I preserves constants and usual arithmetic operators, that is

(1) $I(c) = c$ for all $c \in C_E$,

(2) $I(exp1 \text{ op } exp2) = I(exp1) \text{ op } I(exp2)$, where $op \in \{+, -, \times, /, \uparrow\}$.

A concrete event $e_c = \langle p, S, cou \rangle$ and an abstract event $e_a = \langle p', q, V_E, AT_E \rangle$ match under an interpretation I, if $p = p'$ and $\{I(v) | v \in V_E\} \subset S$ and $I(ACT(p')) = a_{p'}$, $I(TERM(p')) = t_{p'}$ for all $p' \in P_E$. This is denoted by $Matche(e_c, e_a, I)$.

Now we introduce the sets R, BE and EN for Ea. First we supply the abstract events of Ea with indexes 1, 2, ... continuously, in such a manner that any e_a should receive different indexes at different occurrences. If the index of e_a is i, then $e_a(i)$ denotes an *indexed event* of e_a, and the resulting expression is called an *indexed expression* of Ea and denoted $\hat{E}$. Then the sets $R(\hat{E})$, $BE(\hat{E})$ and $EN(\hat{E})$ are defined as follows.

(1) If $\hat{E} = e_a(k)$ then $R(\hat{E}) = \emptyset$, $BE(\hat{E}) = EN(\hat{E}) = \{e_a(k)\}$.

(2) Assume that $Ri = R(\hat{E}i)$, $BEi = BE(\hat{E}i)$ and $ENi = EN(\widehat{Ei})$, $i = 1, 2$,

then

$$R(\widehat{E1; E2}) = R1 \cup R2 \cup (EN1 \times BE2); \quad BE(\widehat{E1; E2}) = BE1,$$

$$BE(\widehat{E1*; E2}) = BE1 \cup BE2,$$

$$EN(\widehat{E1; E2}) = EN2, \qquad EN(\widehat{E1; E2*}) = EN1 \cup EN2,$$

$$R(\widehat{E1+E2}) = R1 \cup R2, \qquad BE(\widehat{E1+E2}) = BE1 \cup BE2,$$

$$EN(\widehat{E1+E2}) = EN1 \cup EN2,$$

$$R(\widehat{E1*}) = R1 \cup (EN1 \times BE1), \qquad BE(\widehat{E1*}) = BE1, \quad EN(\widehat{E1*}) = EN1,$$

$$R(\widehat{E1 \Delta E2}) = R1 \cup R2 \cup (\vec{R}1 \times \overleftarrow{R}2) \cup (\overleftarrow{R}2 \times \vec{R}1)$$

where $\bar{R} = \vec{R} \cup \overleftarrow{R}$, and $\vec{R} = \{a | (a', a) \in R\}$ and $\overleftarrow{R} = \{a | (a, a') \in R\}$,

$$BE(\widehat{E1 \Delta E2}) = BE1 \cup BE2, \qquad EN(\widehat{E1 \Delta E2}) = EN1 \cup EN2.$$

In the following if $(e_a(i), e'_a(k)) \in R(\hat{E})$, then it is written $e_a(i) > e'_a(k)$.

Let $\text{Exp}(\hat{E}) = \{e_a(i) | e_a(i) \text{ is an indexed event in } \hat{E}\}$.

Let $e_a(i) \in \text{Exp}(\hat{E})$ and $M \subset \text{Exp}(\hat{E})$, then $\tilde{e}_a(i) = \{e'_a(k) | e_a(i) > e'_a(k)\}$, and $\vec{M} = \bigcup_{e_a \in M} \tilde{e}_a(i)$.

From the construction of the sets $R(\hat{E})$, $EN(\hat{E})$ and $BE(\hat{E})$ it is easy to see the following properties.

Statement 1.

a) $e_a(k) \in BE(\hat{E})$ iff there is a u such that $e_a(k)u \in L(\hat{E})$,
$e_a(k) \in EN(\hat{E})$ iff there is a u such that $ue_a(k) \in L(\hat{E})$,
$e_a(k) \succ e'_a(n)$ iff there are u, v such that $ue_a(k)e'_a(n)v \in L(\hat{E})$,

b) $e'_a(k_1) \succ \ldots \succ e^n_a(k_n)$, $e^1_a(k_1) \in BE(\hat{E})$ iff there is u such that $e^1_a(k_1) \ldots e^n_a(k_n)u \in L(\hat{E})$.

Example. Let $E((p[q]+g[r]);f*)*$. Then

$$Ea = ((e^1_a + e^2_a);\, e^3_a *)*,$$

$$\hat{E} = ((e^1_a(1) + e^2_a(2));\, e^3_a(3) *)*,$$

$$BE(\hat{E}) = \{e^1_a(1), e^2_a(2)\}, \quad EN(\hat{E}) = \{e^1_a(1), e^2_a(2), e^3_a(3)\},$$

$$R(\hat{E}) = \{(e^1_a(1), e^3_a(3)), (e^2_a(2), e^3_a(3)), (e^3_a(3), e^3_a(3)), (e^1_a(1), e^1_a(1)),$$
$$(e^2_a(2), e^2_a(2)), (e^3_a(3), e^1_a(1)), (e^3_a(3), e^2_a(2)), (e^2_a(2), e^1_a(1)), (e^1_a(1), e^2_a(2))\},$$

where $e^1_a = \langle p, q, V_E, AT_E\rangle$, $e^2_a = \langle g, r, V_E, AT_E\rangle$ $e^3_a = \langle f, \text{true}, V_E, AT_E\rangle$.

Definition. Let $R = \langle OB, P, P', f_a, f_t\rangle$ be a computational system and E an *EBE* such that $P' = P_E$. The *semantics* of E is defined by the predicate $Valid_E$: $\mathbf{B} \to \{\text{true, false}\}$ with the partial map $Next_E$: $\mathbf{B} \to \{\vec{M} \mid M \subset Exp(\hat{E})\}$, in such a way that $Next_E(B)$ is defined iff $Valid_E(B) = \text{true}$. The $Valid_E$ and $Next_E$ are defined recursively as follows.

(1) Let $e_c = \langle p, S, cou\rangle$, then $Valid_E(e_c) = Matchs(cou, f_a, f_t, \emptyset) \& M \neq \emptyset$, where

$$M = \{e_a(i) \mid e_a(i) \in BE(\hat{E}) \,\&\, e_a = \langle p, q, V_E AT_E\rangle \,\&\, (\exists I)(Matche(e_c, e_a, I) \,\&\, Sat(q, I)) = true\}$$

(*Sat* is defined later). And $Next_E(e_c)$ *is* defined iff $Valid_E(e_c) = \text{true}$, and then $Next_E(e_c) = \vec{M}$.

(2) Let $e_C = \langle p, S, cou\rangle$ and $B \in \mathbf{B}$, then

$$Valid_E(Be_c) = Valid_E(B) \& Next_E(B) = \vec{N} \& Matchs(cou, f_a, f_t, B) \& M \neq \emptyset, \text{ where}$$

$$M = \{e_a(i) \mid e_a(i) \in \vec{N} \,\&\, e_a = \langle p, q, V_E, AT_E\rangle \,\&\, (\exists I)(Matche(e_c, e_a, I) \,\&\, Sat(q, I)) = \text{true}\}.$$

And $Next_E(Be_c)$ is defined iff $Valid_E(Be_c) = \text{true}$, and then $Next_E(Be_c) = \vec{M}$.

The definition of the predicate Sat. $Sat(q, I)$ is defined according to the syntax of the predicate q.

$$Sat(\langle counter\ exp\rangle \langle rel\rangle \langle counter\ exp\rangle, I) =$$
$$= I(\langle counter\ exp\rangle) \langle rel\rangle I(\langle counter\ exp\rangle)$$
$$Sat(\langle arithmetic\ exp\rangle \langle rel\rangle \langle arithmetic\ exp\rangle, I) =$$
$$= I(\langle arithmetic\ exp\rangle) \langle rel\rangle I(\langle arithmetic\ exp\rangle)$$
$$Sat(\langle predicate\rangle \langle logic\ op\rangle \langle predicate\rangle, I) =$$
$$= Sat(\langle predicate\rangle, I) \langle logic\ op\rangle Sat(\langle predicate\rangle, I)$$
$$Sat(\neg\langle predicate\rangle, I) = \neg Sat(\langle predicate\rangle, I).$$

Let $\mathbf{B}(E)=\{B|B\in\mathbf{B}$ and $Valid_E(B)=\text{true}\}$.

From the definition of the semantics of *EBE*s it is easy to see the following fact.

Fact 1. Let $Bn=e_c^1\ldots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$, $i=1, \ldots, n$, then

$Valid_E(Bn) = \text{true}$ iff

$Matchs(cou_i, f_a, f_t, B_{i-1})$, $i = 1, \ldots, n$, $B_0 = \emptyset$, and there is a sequence $\{Mi\}_{i=1}^n$ such that

$$Mi = \{e_a(k)|e_a(k)\in\vec{M}_{i-1}\,\&\, e_a =$$

$$= \langle p_i, q, V_E, AT_E\rangle\,\&\,\exists I(Matche(e_c^i, e_a, I)\,\&\, Sat(q, I) = \text{true}\} \neq \emptyset,$$

and $Next_E(Bi) = \overrightarrow{Mi}$, $i = 1, \ldots, n$, $\overrightarrow{Mo} = BE(\hat{E})$.

IV. Some properties of EBE

Definition. Two *EBE*s E and E' are *syntactically equivalent* iff $L(E)=L(E')$.

Definition. Two *EBE*s E and E' are *semantically equivalent* iff $\mathbf{B}(E)=\mathbf{B}(E')$.

Theorem 1. If E and E' are *EBE*s such that $L(E)\subset L(E')$ and for all $u\in L(E')\setminus L(E)$ there are $v\in L(E)$ and w for which $v=uw$ then E and E' semantically equivalent.

Proof. According to the construction of Ea we can identify Ea with E, thus $L(Ea)$ with $L(E)$. First we prove the following facts.

For any E and $Bn=e_c^1\ldots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$.

Fact 2. If there is a sequence $\{Mi\}_{i=1}^n$ such that

$$Mi = \{e_a(k)|e_a(k)\in\vec{M}_{i-1}\,\&\, e_a =$$

$$= \langle p_i, q, V_E, AT_E\rangle\,\&\,\exists I(Matche(e_c^i, e_a, I)\,\&\, Sat(q, I)) = \text{true}\} \neq \emptyset,$$

$$i = 1, \ldots, n, \quad \overrightarrow{Mo} = E(B\hat{E}),$$

then there is a sequence $\{e_a^i(k_i)\}_{i=1}^n$, $e_a^i=\langle p_i, q_i, V_E, AT_E\rangle$, for which $e_a^i(k_i)\in Mi$, $i=1, \ldots, n$ and $e_a^i(k_1)>\ldots>e_a^n(k_n)$.

The existence of the desired sequence is shown by induction as follows.

Since $Mn\neq\emptyset$, thus there is an $e_a^n(k_n)\in Mn$, $e_a^n=\langle p_n, q_n, V_E, AT_E\rangle$. From the definition of Mn there is an $e_a^{n-1}(k_{n-1})\in M_{n-1}$ for which $e_a^{n-1}(k_{n-1})>e_a^n(k_n)$, $e_a^{n-1}=\langle p_{n-1}, q_{n-1}, V_E, AT_E\rangle$. Assume that the sequence $\{e_a^j(k_j)\}_{j=i}^n$, $i>1$, is constructed. Then from the definition of Mi there is an $e_a^{i-1}(k_{i-1})\in M_{i-1}$ for which $e_a^{i-1}(k_{i-1})>e_a^i(k_i)$. So we get the desired sequence.

Fact 3. If there is a sequence $\{e_a^i(k_i)\}_{i=1}^n$, $e_a^i=\langle p_i, q_i, V_E, AT_E\rangle$, such that there is a u for which $e_a^1(k_1)\dots e_a^n(k_n)u\in L(\hat{E})$, and for each $i\leqq n$ there is an I for which $Matche(e_c^i, e_a^i, I)$ *and* $Sat(q_i, I)=$true, then $e_a^i(k_i)\in Mi$, $i=1, \dots, n$ (Mi is defined in Fact 2, $i=1, \dots, n$).

This can easily be proved by induction on $i\leqq n$ (using Statement 1).

Now we prove Theorem 1.

We have to prove that $\mathbf{B}(E)=\mathbf{B}(E')$.

From Fact 1 it is sufficient to prove that for any $Bn=e_c^1\dots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$, $i=1, \dots, n$, the following holds.

$$(+)\begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i=1, \dots, n, Bo=\emptyset, \text{ and there is} \\ \text{a sequence } \{Mi\}_{i=1}^n \text{ such that} \\ Mi=\{e_a(k)|e_a(k)\in\vec{M}_{i-1}\,\&\, e_a=\langle p_i, q, V_E, AT_E\rangle\,\&\,\exists I(Matche(e_c^i, e_a, I)\,\& \\ \qquad Sat(q, I))=\text{true}\}\neq\emptyset, \\ \text{and } Next_E(Bi)=\overrightarrow{Mi},\ i=1, \dots, n,\ \overrightarrow{Mo}=BE(\hat{E}). \end{cases}$$

iff

$$(++)\begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i=1, \dots, n,\ Bo=\emptyset, \text{ and there is} \\ \text{a sequence } \{Ni\}_{i=1}^n \text{ such that} \\ Ni=\{e_a(1)|e_a(1)\in\vec{N}_{i-1}\,\&\, e_a=\langle p_i, q, V_{E'}, AT_{E'}\rangle\,\&\,\exists I(Matche(e_c^i, e_a, I)\,\& \\ \qquad Sat(q, I))=\text{true}\}\neq\emptyset, \\ \text{and } Next_{E'}(Bi)=\overrightarrow{Ni},\ i=1, \dots, n,\ \overrightarrow{No}=BE(\hat{E}'). \end{cases}$$

This is shown by induction on n.

1) It is easy to show that the statement holds for $n=1$.

2) Assume that the statement holds for n. Now we prove that the statement holds for $n+1$ too.

$(+)\Rightarrow(++)$. Assume that $(+)$ holds for $n+1$. Then $(++)$ holds for n. We have yet to prove that $N_{n+1}\neq\emptyset$ and $Next_{E'}(B_{n+1})=\vec{N}_{n+1}$.

According to Fact 2 there is a sequence $\{e_a^i(k_i)\}_{i=1}^{n+1}$, $e_a^i=\langle p_i, q_i, V_E, AT_E\rangle$, for which $e_a^i(k_i)\in Mi$, $i=1, \dots, n+1$, and $e_a^1(k_1)>\dots>e_a^{n+1}(k_{n+1})$. Since $e_a^1(k_1)\in BE(\hat{E})$, thus, by Statement 1, there is a u for which $e_a^1(k_1)\dots e_a^{n-1}(k_{n+1})u\in L(\hat{E})$ which implies that there is a v for which $e_a^1\dots e_a^{n+1}v\in L(Ea)$. Since $L(Ea)\subset L(E'a)$, thus $e_a^1\dots e_a^{n+1}v\in L(E'a)$ which implies that there are a sequence $\{l_i\}_{i=1}^{n+1}$ and a u' for which $e_a^1(l_1)\dots e_a^{n+1}(l_{n+1})u'\in L(\hat{E}')$. Then, by Fact 3, we have $e_a^i(l_i)\in Ni$, $i=1, \dots, n+1$. So $N_{n+1}\neq\emptyset$. Since $(++)$ holds for n, thus $Next_{E'}(Bn)=N_n$ and, by Fact 1, $Valid_{E'}(Bn)=$true, therefore $Valid_{E'}(B_{n+1})=$true (by the definition of Semantics of *EBE*s) which implies $Next_{E'}(B_{n+1})$ is defined and is $\vec{N}_{n+1}$.

$(++)\Rightarrow(+)$. Assume that $(++)$ holds for $n+1$. Then $(+)$ holds for n. We have yet to prove that $M_{n+1}\neq\emptyset$ and $Next_E(B_{n+1})=\vec{M}_{n+1}$. Similarly to the above argument we have the sequence $\{e_a^i(k_i)\}_{i=1}^{n+1}$, $e_a^i=\langle p_i, q_i, V_{E'}, AT_{E'}\rangle$, for which $e_a^i(k_i)\in Ni$, $i=1, \dots, n+1$, and there is a v such that $e_a^1\dots e_a^{n+1}v\in L(E'a)$. We have two cases:

$$\text{either}\quad e_a^1\dots e_a^{n+1}v\in L(Ea)$$

$$\text{or}\quad e_a^1\dots e_a^{n+1}v\in L(E'a)\setminus L(Ea).$$

In the second case there is a v' for which $e_a^1 \ldots e_a^{n+1} vv' \in L(Ea)$. So in both cases we have that there are a sequence $\{l_i\}_{i=1}^{u+1}$ and w for which $e_a^1(l_1) \ldots e_a^{n+1}(l_{n+1}) w \in L(\hat{E})$. Therefore by Fact 3 $e_a^i(l_i) \in Mi$, $i=1, \ldots, n+1$. So $M_{n+1} \neq \emptyset$. Again by the same argument seen above we get that $Next_E(B_{n+1}) = \vec{M}_{n+1}$.

Theorem 2. a) if E and E' are syntactically equivalent then they are semantically equivalent too.

b) There exist two *EBE*s E and E' which are semantically equivalent but not syntactically equivalent.

Proof. a) It is a corollary of Theorem 1.

b) In order to prove this we give an example.

Let $E = p_1 + (p_1; p_2)$ and $E' = p_1; p_2$, it is clear that E and E' satisfy Theorem 1, therefore E and E' are semantically equivalent but not syntactically equivalent because $L(E) \neq L(E')$.

An *EBE* is a GPE *(Generalised Path Expression)* if the operator Δ does not occur in it.

Theorem 3. For every *EBE* E there exists a *GPE* E' such that E and E' are semantically equivalent.

Proof. First we construct an automaton M for which $L(M) = L(E)$. In order to do this we define the sets $R(\hat{E})$, $BE(\hat{E})$ and $EN(\hat{E})$ similarly to those of Section III. The automaton $M = (\Sigma, St, s_0, \delta, F)$ is then constructed as follows. Let $\Sigma = \{e_a | e_a \text{ is in } Ea\} = \{e_a^1, \ldots, e_a^n\}$. Let s_0 be an arbitrary symbol. Then $\delta(s_0, e_a^i) = \{e_a^i(k) | e_a^i(k) \in BE(\hat{E})\} = s_1^i$. So we have defined states $s_0, s_1^1, s_1^2, \ldots, s_1^n$ of St. Suppose that a state s of St is defined, then

$$\delta(s, e_a^i) = \{e_a^i(k) | \exists e_a^j(m)(e_a^j(m) \in s \,\&\, e_a^j(m) \succ e_a^i(k)) = \text{true}\}, \quad i = 1, 2, \ldots, n.$$

Finally let

$$F' = \{s | s \cap EN(\hat{E}) \neq \emptyset\} \quad \text{and} \quad F = F' \cup \{s_0\} \quad \text{if} \quad \varepsilon \in L(Ea) = F,$$

and $F = F'$, otherwise.

It is easy to see that $L(M) = L(Ea)$. It is known that there is a regular expression E' over Σ for which $L(M) = L(E')$. Thus we have $L(Ea) = L(E')$. From the construction of Ea we can identify Ea with E, thus $L(Ea)$ with $L(E)$. Therefore, by Theorem 2, E and E' are semantically equivalent.

V. Implementation of EBE

The implementation of *EBE* is defined using the concept of automaton.

Let $R = \langle OB, P, P', f_a, f_t, \rangle$ be a computational system. Let $Q = \{q | q$ is predicate, $q: OB' \times \{f_a(B)(p) | B \in \mathbf{B}, p \in P'\}^m \times \{f_t(B)(p) | B \in \mathbf{B}, p \in P'\}^n \to \{\text{true, false}\}$,

and $M = (\Sigma, St, s_0, \delta, F)$ a deterministic finite automaton, where $\Sigma \subset P' \times Q$. For all $p \in P'$ the set Condition $(p) = \{\langle s, q \rangle | \delta(s, \langle p, q \rangle)$ is defined$\}$ is called *condition* of the procedure p.

Definition. An *Implementation* is a set $I(M)=\{\langle p, Condition(p)\rangle | p\in P'\}$. For simplicity we often omit the argument M.

Restriction.
It is assumed about the automaton M that if $s_i=\delta(s_{i-1}, b)$, $b_i=\langle p_i, q_i\rangle$, $b_i=\langle p_i, q_i\rangle$, $i=1, 2, \ldots, n$, then there is a u such that $b_1b_2\ldots b_nu\in L(M)$.
Now we define the semantics of Implementations.

Definition. Let I be an Implementation. The *semantics* of I is defined by the predicate $Valid_I$ with the partial map $Next_I$, where $Valid_I$: $\mathbf{B}\to\{\text{true, false}\}$ and $Next_I$: $\mathbf{B}\to 2^{St}$, in such a way that $Next_I(B)$ is defined iff $Valid_I(B)=\text{true}$. The $Valid_I$ and $Next_I$ are defined recursively as follows.
(1) Let $e_c=\langle p, S, cou\rangle$ be an actual event, then

$$Valid_I(e_c) = Matchs(cou, f_a, f_t, \emptyset)\,\&\,\exists\langle s_0, q\rangle(\langle s_0, q\rangle\in Condition(p)\,\&\,Sat(q, e_c, \emptyset)),$$

(*Sat* is defined later).
$Next_I(e_c)$ is defined iff $Valid_I(e_c)=\text{true}$, and then
$Next_I(e_c) = \{s | s\in St\ \&\ \exists q(q\in Q\ \&\ s=\delta(s_0,\langle p, q\rangle)\&Sat(q, e_c, \emptyset))=\text{true}\}$.
(2) Let $e_c=\langle p, S, cou\rangle$ and $B\in\mathbf{B}$, then

$$Valid_I(Be_c) = Valid_I(B)\,\&\,Next_I(B) =$$

$$= G\,\&\,Matchs(cou, f_a, f_t, B)\,\&\,\exists s\exists q(s\in G\,\&\,q\in Q\,\&\,\langle s, q\rangle\in Condition(p)\,\&$$

$$Sat((q, e_c, B)).$$

$Next_I(Be_c)$ is defined iff $Valid_I(Be_c)=\text{true}$, and then $Next_I(Be_c)=H$, where $H=\{s | s\in St\ \&\ \exists s'\,\exists q(s'\in G\ \&\ q\in Q\ \&\ s=\delta(s', \langle p, q\rangle)\ \&\ Sat(q, e, B))=\text{true}\}$.

The definition of Sat. $Sat(q, e_c, B)$ is defined simply as follows.

$$\mathrm{Sat}(q, e_c, B) = q(S, f_a(Be_c)(p), \{f_a(B)(p') | p'\in P'\setminus\{p\}\}, \{f_t(B)(p') | p'\in P'\}).$$

Similarly to Fact 1 it is easy to see the following fact (from the definition of the semantics of Implementation).

Fact 4. For any $Bn=e_c^1\ldots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$, $Valid_I(Bn)=\text{true}$ iff

$Matchs(cou_i, f_a, f_t, B_{i-1})$, $i=1, \ldots, n$, $Bo=\emptyset$, and there is a sequence $\{H_i\}_{i=1}^n$ so that

$$Hi = \{s | \exists s'\exists q(s'\in H_{i-1}\,\&\,q\in Q\,\&\,s = \delta(s', \langle p_i, q\rangle)\,\&\,\mathrm{Sat}(q, e_c^i, B_{i-1})) = \text{true}\} \neq \emptyset,$$

and $Next_I(Bi)=Hi$, $i=1, \ldots, n$, $Ho=\{s_0\}$.
Let $\mathbf{B}(I)=\{B | B\in\mathbf{B}$ and $Valid_I(B)=\text{true}\}$.

Definition. An *Implementation of an EBE E* is an Implementation

$$I = \{\langle p, Condition(p)\rangle | p\in P'\} \text{ such that } P' = P_E \text{ and } \mathbf{B}(I) = \mathbf{B}(E).$$

Now we give an algorithm for transforming an *EBE E* to its Implementation.

Algorithm.

1. Transforming E to the following: we substitute all operands $p[q]$ or p of E by $e=\langle p, q\rangle$ or $e=\langle p, \text{true}\rangle$ respectively. The resulting expression is denoted by Ee.

2. From Ee constructing an automaton $M=\langle \Sigma, St, s_0, \delta, F\rangle$ as that of Theorem 3, where $\Sigma=\{e|e \text{ is in } Ee\}$.

3. For all $p\in P_E$ constructing the set $Condition(p)$, obtaining

$$I = \{\langle P, Condition(p)\rangle | p\in P_E\}.$$

Theorem 4. I is an Implementation of E.

Proof. First we prove the following facts. For any implementation I and actual behavior $Bn=e_c^1 \ldots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$

Fact 5. If there is a sequence $\{Hi\}_{i=1}^n$ such that

$$Hi = \{s|\exists s' \exists q(s'\in H_{i-1} \& q\in Q \& s = \delta(s', \langle p_i, q\rangle) \& \mathrm{Sat}(q, e_c^i, B_{i-1})) = \text{true}\} \neq \emptyset,$$

$$i = 1, \ldots, n, \quad Ho = \{s_0\},$$

then there are sequences $\{s_i\}_{i=0}^n$ and $\{e_i\}_{i=1}^n$, $e^i=\langle p_i, q_i\rangle$, for which $s_i\in Hi$, $s_i=\delta(s_{i-1}, e^i)$ and $Sat(q_i, e_c^i, B_{i-1})=\text{true}$, $i=1, \ldots, n$.

This can be proved by induction as follows. Since $Hn\neq\emptyset$, there is an $s_n\in Hn$. From the definition of Hn there are $s_{n-1}\in H_{n-1}$ and $e^n=\langle p_n, q_n\rangle$ for which $s_n=\delta(s_{n-1}, e^n)$ and $Sat(q_n, e^n, B_{n-1})=\text{true}$. Assume that the sequences $\{s_j\}_{j=i}^n$ and $\{e^j\}_{j=i+1}^n$, are constructed. Then from the definition of Hi there are $s_{i-1}\in H_{i-1}$ and $e^i=\langle p_i, q_i\rangle$, for which $s_i=\delta(s_{i-1}, e^i)$, and $Sat(q_i, e_c^i, B_{i-1})=\text{true}$. So we get the desired sequences.

Fact 6. If there are sequences $\{s_i\}_{i=0}^n$ and $\{e^i\}_{i=1}^n$, $e^i=\langle p_i, q_i\rangle$, for which $s_i=\delta(s_{i-1}, e^i)$ and $Sat(q_i, e^i, B_{i-1})=\text{true}$, $i=1, \ldots, n$, then $s_i\in Hi$, $i=0, 1, \ldots, n$ (Hi is defined in Fact 5).

This can easily be proved by induction on $i\leqq n$.

Now we prove the Theorem. It is easy to see that:

1. The automaton M satisfies the Restriction.

2. $L(M)=L(Ee)$.

Now we show that $\mathbf{B}(E)=\mathbf{B}(I)$. By Fact 1 and Fact 4 it is sufficient to prove that for any EBE E and Implementation I if $Bn=e_c^1 \ldots e_c^n$, $e_c^i=\langle p_i, S_i, cou_i\rangle$, $i=1, 2, \ldots, n$, then

$$(*)\begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i = 1, \ldots, n,\ Bo = \emptyset, \text{ and there is a} \\ \text{sequence } \{Mi\}_{i=1}^n \text{ such that} \\ Mi = \{e_a(k)|e_a(k)\in \vec{M}_{i-1} \& e_a^i = \langle p_i, q, V_E, AT_E\rangle \& \exists \text{ interpretation } I \\ (Matche(e_c^i, e_a, I) \& Sat(q, I)) = \text{true}\} \neq \emptyset, \text{ and } Next_E(Bi) = \vec{Mi}, \\ i = 1, \ldots, n, \quad \vec{Mo} = BE(\hat{E}) \end{cases}$$

iff

$$(**)\begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i=1,\ldots,n, \text{ and there is a} \\ \text{sequence } \{Hi\}_{i=1}^{n} \text{ such that} \\ Hi=\{s \mid s\in St \,\&\, \exists s' \exists q (s'\in H_{i-1} \,\&\, q\in Q \,\&\, s=\delta(s', \langle p_i, q\rangle) \,\&\, Sat(q, e_c^i, B_{i-1})) \\ \quad =\text{true}\}\neq\emptyset, \text{ and } Next_I(Bi)=Hi,\quad i=1,\ldots,n,\quad Ho=\{s_0\}. \end{cases}$$

From the construction of *Ea* and *Ee* we can identify *Ea* with *Ee* and therefore $L(Ea)$ with $L(Ee)$ too, so $L(Ea)=L(Ee)=L(M)$.

Now we prove that $(*)$ iff $(**)$ for any *Bn*. This is shown by induction on n.

(1) It is easy to see that the statement holds for $n=1$.

(2) Assume that the statement holds for n.

$(*)\Rightarrow(**)$. Suppose that $(*)$ holds for $n+1$. Then $(**)$ holds for n. We have yet to prove that $H_{n+1}\neq\emptyset$, and $Next_I(B_{n+1})=H_{n+1}$.

By Fact 2 we have a sequence $\{e_a^i(k_i)\}_{i=1}^{n+1}$, $e_a^i=\langle p, q_i, V_E, AT_E\rangle$, such that $e_a^1(k_1)>\ldots>e_a^{n+1}(k_{n+1})$, and $e_a^i(k_i)\in Mi$, $i=1,\ldots,n+1$. Therefore according to Statement 1 there is a u for which $e_a^1(k_1)\ldots e_a^{n+1}(k_{n+1})u\in L(\hat{E})$ which implies that there is a v such that $e_a^1\ldots e_a^{n+1}v\in L(Ea)$. Since $L(Ea)=L(M)$ thus there exists a sequence $\{s_i\}_{i=0}^{n+1}$ such that $s_i=\delta(s_{i-1}, e^i)$, where $e^i=\langle p_i, q_i\rangle$, $i=1,\ldots. n+1$. It is easy to see that for all $i\leqq n+1$, $Sat(q_i, e_c^i, B_{i-1})=$true (from the definition of *Matche, Matchs,* interpretation *I*, $Sat(q, I)$ and $Sat(q, e_c, B)$). So by Fact 6 we have $s_i\in Hi$, $i=1,\ldots,n+1$, that is $H_{n+1}\neq\emptyset$. Since $(**)$ holds for n thus $Next_I(Bn)=Hn$ and, by Fact 4, $Valid_I(Bn)=$true. Therefore from the definition of the Semantics of Implementation $Valid_I(B_{n+1})=$true which implies that $Next_I(B_{n+1})$ is defined and is H_{n+1}.

$(**)\Rightarrow(*)$. Assume that $(**)$ holds for $n+1$. Then $(*)$ holds for n. We have yet to prove that $M_{n+1}\neq\emptyset$ and $Next_E(B_{n+1})=\vec{M}_{n+1}$.

According to Fact 5 we have the sequence $\{s_i\}_{i=0}^{n+1}$ and $\{e^i\}_{i=1}^{n+1}$ for which $s_i\in Hi$, $s_i=\delta(s_{i-1}, e^i)$, $Sat(q_i, e_c^i, B_{i-1})=$true, and $e^i=\langle p_i, q_i\rangle$, $i=1,\ldots,n+1$. Then, by Restriction, there is a u for which $e^1\ldots e^{n+1}u\in L(M)=L(Ea)$ which implies that there are a v and a sequence $\{k_i\}_{i=1}^{n+1}$ for which $e_a^1(k_1)\ldots e_a^{n+1}(k_{n+1})v\in L(\hat{E})$, $e_a^i=$ $=\langle p_i, q_i, V_E, AT_E\rangle$. It is easy to see that for each $i\leqq n+1$ there is an interpretation *I* for which $Matche(e_e^i, e_a^i, I)$ and $Sat(q_i, I)=$true (again from the definition of *Matche, Matchs, Interpretation I*, $Sat(q, I)$ and $Sat(q, e_c, B)$). Therefere, by Fact 3, $e_a^i(k_i)\in Mi$, $i=1,\ldots,n+1$. So $M_{n+1}\neq\emptyset$. Since $(*)$ holds for n, thus $Next_E(Bn)=Mn$ and, by Fact 1, $Valid_E(Bn)=$true, therefore according to the definition of semantics of *EBE*s we have $Valid_E(B_{n+1})=$true which implies that $Next_I(B_{n+1})$ is defined and is $\vec{M}_{n+1}$.

VI. Reduction of EBEs

Now we give some rules for reducing *EBE*s.

Statement 2. Let $E1$, $E2$ and $E3$ be *EBE*s. Then

(1) $E1+E1 \approx E1$

(2) $E1+E2 \approx E2+E1$

(3) $(E1+E2)+E3 \approx E1+(E2+E3)$

(4) $(E1;E2);\ E3 \approx E1;\ (E2;E3)$

(5) $E1;(E2+E3) \approx E1;\ E2+E1;\ E3$

(6) $(E1+E2);\ E3 \approx E1;\ E3+E2;\ E3$

(7) $E1 \Delta E2 \approx E2 \Delta E1$

(8) $(E1 \Delta E2) \Delta E3 \approx E1 \Delta (E2 \Delta E3)$

(9) $E1 \Delta (E2+E3) \approx E1 \Delta E2+E1 \Delta E3$

(10) $\overset{n}{\underset{i=1}{\Delta}} p_i[q_i] \approx \sum_{\substack{i_1 \ldots i_n \text{ is} \\ \text{permutation} \\ \text{of } \{1,\ldots,n\}}} (p_{i_1}[q_{i_1}];\ \ldots;\ p_{i_n}[q_{i_n}])$

where "$\approx$" means semantical equivalence. This is followed from Theorem 2.

Similarly to *EBE*s we also define the syntactical and semantical equivalence of Implementions.

Definition. Two Implementations $I(M)$ and $I'(M')$ are *syntactically equivalent* if $L(M)=L(M')$, and *semantically equivalent* if $\mathbf{B}(I)=\mathbf{B}(I')$.

Definition. An Implementation $I(M)$ is *minimal* if the automaton M has a minimum number of states.

Theorem 5. There exists an algorithm by means of which we can transform any Implementation $I(M)$ to a minimal Implementation $I'(M')$ so that $I(M)$ and $I'(M')$ are semantically equivalent.

Proof. It is known that there is an algorithm by means of which we can reduce any automaton M to a minimal automaton M' such that $L(M)=L(M')$. The semantical equivalence of $I(M)$ and $I'(M')$ is then followed from the following statement.

Statement 3. If $I(M)$ and $I'(M')$ are Implementations such that $L(M) \subset L(M')$, and for any $u \in L(M') \setminus L(M)$ there are $v \in L(M)$ and w for which $v=uw$, then $I(M)$ and $I'(M')$ are semantically equivalent.

Proof. Let $M=(\Sigma, St, s_0, \delta, F)$ and $M'=(\Sigma', St', s_0', \delta', F')$.

By Fact 4 it is sufficient to prove that for any $Bn=e_c^1 \dots e_c^n$ the following holds:

$$(1)\quad \begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i=1, \dots, n, \text{ and there is a} \\ \text{sequence } \{Hi\}_{i=1}^n \text{ such that} \\ Hi = \{s \mid \exists s' \exists q (s' \in H_{i-1} \& q \in Q \& s = \delta(s', \langle p_i, q \rangle) \& Sat(q, e_c^i, B_{i-1}) = \text{true}\} \neq \emptyset, \\ \text{and } Next_I(Bi) = Hi,\ i=1, \dots, n,\ Ho = \{s_0\}. \end{cases}$$

iff

$$(2)\quad \begin{cases} Matchs(cou_i, f_a, f_t, B_{i-1}),\ i=1, \dots, n, \text{ and there is a sequence } \{H'i\}_{i=1}^n \\ \text{such that} \\ H_i' = \{s \mid \exists s' \exists q (s' \in H_{i-1}' \& q \in Q \& s = \delta'(s', \langle p_i, q \rangle) \& Sat(q, e_c^i, B_{i-1}) = \text{true}\} \neq \emptyset, \\ \text{and } Next_{I'}(Bi) = Hi,\ i=1, \dots, n,\ H'o = \{s_0'\}. \end{cases}$$

This is proved by induction on n.

1) It is easy to see that the statement holds for $n=1$.

2) Assume that the statement holds for n, we prove that it holds for $n+1$, too.

(1)⇒(2). Suppose that (1) holds for $n+1$. Then (2) holds for n. We have yet to prove that $H_{n+1}' \neq \emptyset$ and $Next_{I'}(B_{n+1}) = H_{n+1}'$. By Fact 5 we have the sequences $\{s_i\}_{i=1}^{n+1}$ and $\{e^i\}_{i=1}^{n+1}$ for which $s_i \in Hi$, $s_i = \delta(s_{i-1}, e^i)$, $Sat(q_i, e_c^i, B_{i-1}) = \text{true}$, $e^i = \langle p_i, q_i \rangle$, $i=1, \dots, n+1$. Then according to Restriction there is a u for which $e' \dots e^{n+1}u \in L(M)$. Since $L(M) \subset L(M')$ thus there is a sequence $\{s_i'\}_{i=0}^{n+1}$ for which $s_i' = \delta'(s_{i-1}', e^i)$. So by Fact 6 $s_i' \in H'i$, $i=0, \dots, n+1$, that is $H_{n+1}' \neq \emptyset$. Since (2) holds for n, thus $Next_{I'}(Bn) = H'n$ and, by Fact 4, $Valid_{I'}(Bn) = \text{true}$, so $Valid_{I'}(B_{n+1}) = \text{true}$ (according to the definition of the semantics of Implementation) which implies that $Next_{I'}(B_{n+1})$ is defined and is H_{n+1}'.

(2)⇒(1). Suppose that (2) holds for $n+1$. Then (1) holds for n. We have yet to prove that $H_{n+1} \neq \emptyset$ and $Next_I(B_{n+1}) = H_{n+1}$.

By Fact 5 we have the sequences $\{s_i\}_{i=0}^{n+1}$ and $\{e_i\}_{i=1}^{n+1}$ for which $s_i \in H'i$, $s_i = \delta'(s_i, e^i)$, $Sat(q_i, e_c^i, B_{i-1}) = \text{true}$, $e^i = \langle p_i, q_i \rangle$, $i=1, \dots, n+1$. Then according to Restriction there is a u for which $e^1 \dots e^{n+1}u \in L(M')$. We have two cases:

$$\text{either} \quad e^1 \dots e^{n+1}u \in L(M)$$

$$\text{or} \quad e^1 \dots e^{n+1}u \in L(M') \setminus L(M).$$

In the second case there is u' for which $e^1 \dots e^{n+1}uu' \in L(M)$. So in both cases, we have the sequence $\{s_i'\}_{i=0}^{n+1}$ for which $s_i' = \delta(s_{i-1}', e^i)$, $i=1, \dots, n+1$. Therefore, by Fact 6, $s_i' \in Hi$, $i=0, 1, \dots, n+1$, that is $H_{n+1} \neq \emptyset$. Now by the same argument seen above we get $Next_I(B_{n+1}) = H_{n+1}$.

Abstract

A language, called *EBE*, for specifying the expected behavior of programs during debugging is presented. *EBE* is an extended version of *GPE* (Generalized Path Expressions) [1] with the operator shuffle. The syntax and semantics of *EBE* is formally defined. Some properties of *EBE*s are discussed. Then an implementation of *EBE* is presented. Correctness of implementation is also proved.

DEPARTMENT OF COMPUTER SCIENCE
EÖTVÖS LORÁND UNIVERSITY
MÚZEUM KRT. 6—8
BUDAPEST, HUNGARY
H—1088

References

[1] BRUEGGE, B. and HIBBARD, P., Generalized Path Expressions: A High-Level Debugging Mechanism, The J. of Sys. and Soft. 3, 256—276 (1983).
[2] LAVENTHAL, M. S., Synthesis of synchronization code for data abstractions, M.I.T. Laboratory for Comp. Sci., 1978.
[3] VARGA, L., Rendszerprogramok elmélete és gyakorlata. Akadémiai Kiadó, Budapest 1978 (in Hungarian).
[4] GÉCSEG, F. and PEÁK, I., Algebraic Theory of Automata, Akadémiai Kiadó, Budapest, 1972.
[5] NGUYEN HUU CHIEN, Sequential program debugging with Path Expressions, conference on Automata, Languages and Programming Systems, Salgótarján, May 1986 (Hungary).
[6] NGUYEN HUU CHIEN, EBE: A Language for Specifying the Expected Behavior of Programs in Debugging, 2nd conference of Program Designers, Budapest, L. Eötvös University, July, 1986.

(Received Sept. 3, 1986)

Some remarks on the algorithm of Lucchesi and Osborn

HO THUAN

Let $S=\langle \Omega, F\rangle$ be a relation scheme, where $\Omega=\{A_1, A_2, \ldots, A_n\}$ is the universe of attributes and

$$F = \{L_i \to R_i | L_i, R_i \subseteq \Omega,\ i = 1, 2, \ldots, m\}$$

is the set of functional dependencies. In [2] C. L. Lucchesi and S. L. Osborn provided a very interesting algorithm to determine the set of all keys for any relation scheme $S=\langle \Omega, F\rangle$. Following our notation, the algorithm has time complexity

$$O(|F|\,|\mathscr{K}_S|\,|\Omega|(|\mathscr{K}_S|+|\Omega|)),$$

i.e. its running time is bounded by a polynomial of $|\Omega|$, $|F|$ and $|\mathscr{K}_S|$, where

$|F|$ is the cardinality of F, and

$\mathscr{K}_S$ is the set of all keys for S.

We reproduce here this algorithm with some modifications in accordance with our notation.

Algorithm OL1. Set of all keys for $S=\langle \Omega, F\rangle$;

Comment. $\mathscr{K}_S$ is the set of keys being accumulated in a sequence which can be scanned in the order in which the keys are entered;

$$\mathscr{K}_S \leftarrow \{\text{Key}(\Omega, F, \Omega)\};^{1}$$

for each K in $\mathscr{K}_S$ **do**

for each $FD(L_i \to R_i)$ in F **do**

$T \leftarrow L_i \cup (K \setminus R_i)$;

test $\leftarrow$ true;

for each J in $\mathscr{K}_S$ **do**

[1] Let Key (Ω, F, X) be the algorithm Minimal Key in [2], which determines a key for S that is a subset of a specified superkey X.

if T includes J **then** test ← false;

if test **then** $\mathscr{K}_S \leftarrow \mathscr{K}_S \cup \{\text{Key}(\Omega, F, T)\}$

end

end;

return $\mathscr{K}_S$

The following simple remarks, in some cases can be used to improve the performance of the algorithm of Lucchesi and Osborn.

Remark 1. Let L, R, H be defined as: $R=\bigcup_{i=1}^{m} R_i$, $L=\bigcup_{i=1}^{m} L_i$ $H=\bigcup_{K_j\in\mathscr{K}_S} K_j$. To find the first key for $S=\langle\Omega, F\rangle$, instead of Ω it is better to use the superkey $(\Omega\setminus R)\cup(L\cap R)$ and Algorithm 1 in [1], and instead of the algorithm Key (Ω, F, T) it is better to use Algorithm 2 in [1] for finding one key for S included in a given superkey T.

Remark 2. In [1] it is shown that

$$R\setminus L \subseteq \Omega\setminus H,$$

i.e. $R\setminus L$ consists only of non-prime attributes. Therefore, if $R_i\subseteq R\setminus L$ then $R_i\cap K=\emptyset$, $\forall K\in\mathscr{K}_S$, and $L_i\cup(K\setminus R_i)\supseteq K$. That means, when computing $T=L_i\cup\cup(K\setminus R_i)$, we can neglect all FDs $(L_i\rightarrow R_i)$ with $R_i\subseteq R\setminus L$ for every $K\in\mathscr{K}_S$.

Let us denote

$$\bar{F} = F\setminus\{L_j \rightarrow R_j | L_j \rightarrow R_j \in F \quad \text{and} \quad R_j \subseteq R\setminus L\}.$$

Remark 3. With a fixed K in $\mathscr{K}_S$, it is clear that if $K\cap R_i=\emptyset$ then $L_i\cup(K\setminus R_i)\supseteq K$. In that case, it is not necessary to continue to check whether T includes J for each J in $\mathscr{K}_S$. So, it is better to compute T by the following order

$$T = (K\setminus R_i)\cup L_i.$$

Remark 4. The algorithm of Lucchesi and Osborn is particularly effective when the number of keys for $S=\langle\Omega, F\rangle$ is small. But on what basis can we conclude that the number of keys for S is small? There is no general answer for all cases, and it is shown in [3] that the number of keys for a relation scheme $S=\langle\Omega, F\rangle$ can be factorial in $|F|$ or exponential in $|\Omega|$, and that both of these upper bounds are attainable. However, it is shown in ([1], Corollary 1) that

$$|\mathscr{K}_S| \leqq C_h^{[h/2]}$$

where h is the cardinality of $L\cap R$. Thus, if $L\cap R$ has a few elements only, then it is a good criterion for saying that S has a small number of keys. In the case $L\cap R=\emptyset$, $\Omega\setminus R$ is the unique key for $S=\langle\Omega, F\rangle$ as pointed out in ([1], Corollary 4).

Example. We take up the example in [2], Appendix 1):

$$\Omega = \{a, b, c, d, e, f, g, h\},$$

$$F = \{a \rightarrow b,\ c \rightarrow d,\ e \rightarrow f,\ g \rightarrow h\}.$$

It is clear that for this relation scheme

$$L \cap R = \emptyset,$$

and it has exactly one key, namely $\Omega \setminus R = aceg$.

Taking Remarks 1—3 into account, the algorithm of Lucchesi and Osborn now can be presented as follows:

Algorithm OL2. Set of all keys for $S = \langle \Omega, F \rangle$;

$$\mathcal{K}_S \leftarrow \{\text{Algo. 1}(\Omega, F, (\Omega \setminus R) \cup (L \cap R))\};^2$$

for each K in $\mathcal{K}_S$ **do**

for each $FD(L_i \to R_i)$ in $\bar{F}$ such that $K \setminus R_i \neq K$ **do**

$T \leftarrow (K \setminus R_i) \cup L_i$;

test $\leftarrow$ true;

for each J in $\mathcal{K}_S$ **do**

if T includes J **then** test $\leftarrow$ false;

if test **then** $\mathcal{K}_S \leftarrow \mathcal{K}_S \cup \{\text{Algo. 2}(\Omega, F, T)\}$

end

end;

return $\mathcal{K}_S$.

Remark 5. The time complexity of Algorithm OL2 is

$$0(|\mathcal{K}_S||\Omega|(|\mathcal{K}_S||\bar{F}| + |F||L \cap R|)).$$

Abstract

In [1] we have proposed two algorithms (Algorithm 1 and Algorithm 2) for finding one key of the relation scheme $S = \langle \Omega, F \rangle$ included in a given superkey. In this paper, we show that, using these algorithms and some simple remarks, the performance of the algorithm of Lucchesi and Osborn [2], in general, can be improved.

References

[1] Ho Thuan and Le van Bao, Some results about keys of relational schemes, Acta Cybernetica Tom. 7, Fasc. 1, Szeged, 1985, 99—113.

[2] Lucchesi, C. L. and Osborn, S. L., Candidate keys for relations, J. of Computer and System Sciences, 17, 1978, 270—279.

[3] Osborn, S. L., Normal forms for relational databases, Ph. D. Dissertation, University of Waterloo, 1977.

(Received Oct. 27, 1986)

[2] Algo. 1 and Algo. 2 refer to Algorithm 1 and Algorithm 2 in [1] respectively.

Strong dependencies and s-semilattices

VU DUC THI

1. Introduction

The full family of functional dependencies was first axiomatized by W. W. Armstrong [1]. Different kinds of functional dependencies have also been investigated in relational data base theory. The full family of strong dependencies has been introduced and axiomatized [2], [3], [4].

In this paper s-semilattices and strong operations are defined. We investigate connections between full families of strong dependencies, s-semilattices and strong operations. We prove that there are one-to-one correspondences between them, and s-semilattices completely determine both full families of strong dependencies and strong operations. We give a necessary and sufficient condition for an arbitrary family of sets to be a full family of strong dependencies. A necessary and sufficient condition for a relation to represent a given full family of strong dependencies is also given. Finally, we show that for a given s-semilattice I, we can construct a concrete relation R, the full family of strong dependencies of which is determined by I.

We start with some necessary definitions formulated in [3].

Definition 1.1. Let $R=\{h_1, \ldots, h_m\}$ be a relation over the finite set of attributes Ω, and $A, B \subseteq \Omega$. We say that B strongly depends on A in R (denoted $A \xrightarrow[R]{s} B$) iff

$$(\forall h_i, h_j \in R)\big((\exists a \in A)\big(h_i(a) = h_j(a)\big) \rightarrow (\forall \in B)\big(h_i(b) = h_j(b)\big)\big).$$

Let $S_R = \{(A, B)\colon A \xrightarrow[R]{s} B\}$. S_R is called the full family of strong dependencies of R.

Definition 1.2. Let Ω be a finite set, and denote by $P(\Omega)$ its power set. Let $Y \subseteq P(\Omega) \times P(\Omega)$. We say that Y is a full family of strong dependencies over Ω if for all $A, B, C, D \subseteq \Omega$, $a \in \Omega$,

S1 $(\{a\}, \{a\}) \in Y$;

S2 $(A, B) \in Y, \quad (B, C) \in Y, \quad B \neq \emptyset \rightarrow (A, C) \in Y$;

S3 $(A, B) \in Y, \quad C \subseteq A, \quad D \subseteq B \rightarrow (C, D) \in Y$;

S4 $(A, B) \in Y, \quad (C, D) \in Y \rightarrow (A \cup C, B \cap D) \in Y$;

S5 $(A, B) \in Y, \quad (C, D) \in Y \rightarrow (A \cap C, B \cup D) \in Y$.

Definition 1.3. Let $I \subseteq P(\Omega)$. We say that I is a $\cap$-semilattice over Ω if $\Omega \in I$, and $A, B \in I \rightarrow A \cap B \in I$. Let $M \subseteq P(\Omega)$. Denote by M^+ the set $\{\cap M' : M' \subseteq M\}$. Then we say M generates I if $M^+ = I$.

J. Demetrovics in [3] showed that for a given $\cap$-semilattice I, there is exactly one family N which generates I and has minimal cardinality.

Lemma 1.4. ([3]). Let $I \subseteq P(\Omega)$ be a $\cap$-semilattice over Ω. Let

$$N = \{A \in I : \forall B, C \in I : A = B \cap C \rightarrow A = B \text{ or } A = C\}.$$

Then N generates I and if N' generates I, then $N \subseteq N'$. N is called the minimal generator of I. (It is obvious that $\Omega \in N$.)

It can be seen that if $N_1 (N_2)$ is the minimal generator of $I_1 (I_2)$ and $I_1 \neq I_2$, then $N_1 \neq N_2$ holds.

2. The results

Definition 2.1. Let $I \subseteq P(\Omega)$. We say that I is an s-semilattice over Ω if I satisfies

(1) I is a $\cap$-semilattice,

(2) for all $A \in N \setminus \Omega$

$$(\exists a \in A)((\forall B \in N \setminus \Omega)(A \not\subset B) \rightarrow a \notin B),$$

where N is the minimal generator of I.

Definition 2.2. The mapping $F: P(\Omega) \rightarrow P(\Omega)$ is called a strong operation over Ω if for every $a, b \in \Omega$ and $A \in P(\Omega)$, the following properties hold:

(1) $F(\emptyset) = \Omega$,

(2) $a \in F(\{a\})$,

(3) $b \in F(\{a\}) \rightarrow F(\{b\}) \subseteq F(\{a\})$,

(4) $F(A) = \bigcap_{a \in A} F(\{a\})$.

It is easy to see that the set $\{F(\{a\}) : a \in \Omega\}$ determines the set $\{F(A) : A \in P(\Omega)\}$.

The following theorem shows that there is an one-to-one correspondence between s-semilattices and strong operations.

Theorem 2.3. Let F be a strong operation over Ω. Let $I_F=\{F(A)\colon A\in P(\Omega)\}$. Then I_F is an s-semilattice over Ω. Conversely, if I is an s-semilattice over Ω, then there is exactly one strong operation F so that $I_F=I$, where $F(\emptyset)=\Omega$, and for all $a\in\Omega$,

$$F(\{a\})=\begin{cases}\bigcap\limits_{\substack{a\in A_i\\ A_i\in N\setminus\Omega}} A_i & \text{if } \exists A_i\colon a\in A_i (N \text{ is the minimal generator of } I),\\ \Omega & \text{otherwise.}\end{cases}$$

Proof. It is clear that for arbitrary strong operation F

$$\forall A, B\in P(\Omega)\colon F(A\cup B)=F(A)\cap F(B),\ F(\emptyset)=\Omega$$

and $A\subseteq B\to F(B)\subseteq F(A)$. Consequently, $I_F=\{F(A)\colon A\in P(\Omega)\}$ is a $\cap$-semilattice over Ω. Denote by N_F the minimal generator of I_F. For all $A\in N_F\setminus\Omega$ if there is no attribute a such that $F(\{a\})=A$, then if $A=F(B)(|B|\geqq 2)$ holds, then, according to the definition of strong operation, $A=\bigcap\limits_{b_i\in B} F(\{b_i\})$. This contradicts the definition of minimal generator. Consequently, there is an attribute $a\in\Omega$ so that $F(\{a\})=A$. It is obvious that $a\in A$. It is clear that $A, B\in N_F$ implies $A\neq B$, and by (3) in the definition of strong operation, for all $A\in N_F\setminus\Omega$:

$$(\exists a\in A)((\forall B\in N_F\setminus\Omega)(A\not\subset B)\to a\notin B).$$

Consequently, I_F is an s-semilattice over Ω.

Conversely, we now suppose that I is an s-semilattice over Ω. Denote by N the minimal generator of I. We define the following operation F:

$$F(\emptyset)=\Omega,$$

and for all b,

$$F(\{b\})=\begin{cases}\bigcap\limits_{\substack{b\in A_i\\ A_i\in N\setminus\Omega}} A_i & \text{if } \exists A_i\colon b\in A_i,\\ \Omega & \text{otherwise.}\end{cases}$$

It can be seen that for all $A\in N\setminus\Omega$, where $\exists a\in A\colon A_i\in N\setminus\Omega$ and $A\not\subset A_i\to a\notin A_i$, we have $F(\{a\})=A$. For all different elements $A(A\in N\setminus\Omega)$ it is easy to see that there is an $a\in A$ so that $F(\{a\})=A$. Consequently, $\forall A\in N\setminus\Omega\colon \exists a\in\Omega\colon F(\{a\})=A$. We now show that F is a strong operation over Ω. It can be seen that $b\in F(\{b\})$, and if there is an $A_i\in N\setminus\Omega$ such that $b\in A_i$, then $F(\{b\})\in N^+$. If $a\in F(\{b\})$ holds, then

$$F(\{a\})=\bigcap_{\substack{a\in A_i\\ A_i\in N\setminus\Omega}} A_i\subseteqq\bigcap_{\substack{b\in A_i\\ A_i\in N\setminus\Omega}} A_i=F(\{b\}).$$

On the other hand, it can be seen that the set $\{F(\{b\})\colon b\in\Omega\}$ determines the set $\{F(A)\colon A\in P(\Omega)\}$. Consequently, F is a strong operation over Ω. It is easy to see that $I=\{F(A)\colon A\in P(\Omega)\}$. If we suppose that there is a strong operation F' such that $I_{F'}=I$ then for all $a\in\Omega$ $\exists b\in\Omega\colon F(\{a\})=F'(\{b\})$. It obvious that $a\in F'(\{b\})$. Consequently, $F'(\{a\})\subseteqq F(\{a\})$. On the other hand, there is an attribute c so that $F'(\{a\})=F(\{c\})$. Clearly, $F(\{a\})\subseteqq F'(\{a\})$ by $a\in F(\{c\})$. Consequently, for all $A\in P(\Omega)$, $F'(A)=F(A)$. The proof is complete.

Based on Theorem 2.3, it is easy to see that s-semilattices determine the strong operations, and for arbitrary s-semilattice I over Ω, $|N|$ in not greater than $|\Omega|+1$. Clearly, there is an algorithm to decide for a given family of sets $N \subseteq P(\Omega)$ whether N is the minimal generator of some s-semilattice or not. The following theorem gives necessary and sufficient conditions for an arbitrary family of sets to be a full family of strong dependencies over Ω.

Theorem 2.4. Let $Y \subseteq P(\Omega) \times P(\Omega)$. Y is a full family of strong dependencies over Ω if and only if there is a family $\{E_i: i=1, ..., l; \bigcup_{i=1}^{l} E_i = \Omega\}$ of subsets of Ω such that

(i) for all $A \subseteq \Omega$, $(\emptyset, A) \in Y$,

(ii) for any $A, B \subseteq \bigcup_{E_i \cap A \neq \emptyset} E_i \rightarrow (A, B) \in Y$,

(iii) $((C, D) \in Y,\ C \cap E_i \neq \emptyset) \rightarrow D \subseteq E_i$.

Proof. First we suppose that Y is a full family of strong dependencies over Ω. Then by (S1), (S3), (S5) for each $a \in \Omega$ we can construct an E_i ($E_i \subseteq \Omega$) so that $(\{a\}, E_i) \in Y$, and $\forall E'$: $E \subset E'$ implies $(\{a\}, E') \notin Y$. It is obvious that $a \in E_i$, and we obtain n such E_i's, where $n = |\Omega|$. Thus, we have the set $E = \{E_i: i=1, ..., n; \bigcup_{i=1}^{n} E_i = \Omega\}$. It is easy to see that for all $A \subseteq \Omega$ we have $(\emptyset, A) \in Y$. We now assume that $A = \{a_1, ..., a_k: a_j \in \Omega, j=1, ..., k\} \neq \emptyset$ and B_1 is a set such that $(A, B_1) \in Y$, $\forall B_2: B_1 \subset B_2$ implies $(A, B_2) \notin Y$. According to the construction of E, it is clear that for each a_j there is an $E_{i_j} \in E$ so that $(\{a_j\}, E_{i_j}) \in Y$. By (S4) we have $(\bigcup_{j=1}^{k} a_j, \bigcap_{j=1}^{k} E_{i_j}) = (A, \bigcap_{j=1}^{k} E_{i_j}) \in Y$. By the definition of B_1 we obtain $\bigcap_{j=1}^{k} E_{i_j} \subseteq B_1$. On the other hand, by $(A, B_1) \in Y$ and by (S3), we have $(\{a_j\}, B_1) \in Y$ for all j ($j=i, ..., k$). Consequently, $B_1 \subseteq \bigcap_{j=1}^{k} E_{i_j}$ holds, i.e. $B_1 = \bigcap_{j=1}^{k} E_{i_j}$. It is obvious that $\bigcap_{E_i \cap A \neq \emptyset} E_i \subseteq \bigcap_{j=1}^{k} E_{i_j}$. Thus, for all $B (B \subseteq \bigcap_{E_i \cap A \neq \emptyset} E_i)$: $B \subseteq B_1$. Hence $(A, B) \in Y$ holds. If $(C, D) \in Y$, $C \cap E_i \neq \emptyset$, then we assume that $a_1 \in C \cap E_i$. On the other hand, suppose that a is an attribute such that $(\{a\}, E_i) \in Y$, and $\forall E'$: $E_i \subset E'$ implies $(\{a\}, E') \notin Y$. By $a_1 \in E_i$ and by (S3), $(\{a\}, \{a_1\}) \in Y$ holds. By (S3) and $a_1 \in C$ we obtain $(\{a_1\}, D) \in Y$. Consequently, by (S2), $(\{a\}, D) \in Y$ holds. According to the definition of E we have $D \subseteq E_i$.

The proof of the reverse direction is easy, and so it is omitted. The proof is complete.

Based on Theorem 2.4 we prove the following result, which shows that between full families of strong dependencies and strong operations there exists a one-to-one correspondence.

Theorem 2.5. Let Y be a full family of strong dependencies over Ω. We define the mapping F_Y: $P(\Omega) \rightarrow P(\Omega)$ as follows:

$F_Y(A)=\{a\in\Omega\colon (A,\{a\})\in Y\}$.

Then F_Y is a strong operation over Ω. Conversely, if F is an arbitrary strong operation over Ω, then there is exactly one full family of strong dependencies Y so that $F_Y=F$, where

$$Y=\{(A,B)\colon A,B\in P(\Omega)\colon B\subseteq F(A)\}.$$

Proof. Suppose that Y is a full family of strong dependencies over Ω. It is obvious that $\forall a\in\Omega\colon a\in F_Y(\{a\})$. By Theorem 2.4 we have $(C,D)\in Y$, $C\cap E_i\neq\emptyset$ imply $D\subseteq E_i$. It can be seen that in Theorem 2.4, for any $a\in\Omega$,

$$F_Y(\{a\})\in\{E_i\colon i=1,\ldots,n;\ |\Omega|=n;\ \bigcup_{i=1}^{n}E_i=\Omega\}.$$

Consequently, $(\{b\},F_Y(\{b\}))\in Y$, $b\in F_Y(\{a\})$, i.e. $b\cap F_Y(\{a\})\neq\emptyset$ implies $F_Y(\{b\})\subseteq F_Y(\{a\})$. By (iii) in Theorem 2.4 we obtain $(A,F_Y(A))\in Y$, $\forall a\in A\colon A\cap F_Y(\{a\})\neq\emptyset$ imply $F_Y(A)\subseteq F_Y(\{a\})$. Thus, $F_Y(A)\subseteq\bigcap_{a\in A}F_Y(\{a\})$. On the other hand, by (S5) in the definition of full family of strong dependencies we have $\forall a\in A\colon(\{a\},F_Y(\{a\}))\in Y$ implies $(A,\bigcap_{a\in A}F_Y(\{a\}))\in Y$, i.e. $\bigcap_{a\in A}F_Y(\{a\})\subseteq F_Y(A)$. Consequently, $F_Y(A)=\bigcap_{a\in A}F_Y(\{a\})$ holds. Conversely, assume that F is a strong operation over Ω, and $Y=\{(A,B)\colon B\subseteq F(A)\}$. We have to show that Y is a full family of strong dependencies. By Theorem 2.4 we set $E=\{F(\{a\})\colon a\in\Omega,\ |\Omega|=\ \}$. By the definition of Y and by $\bigcap_{F(\{a\})\cap A\neq\emptyset}F(\{a\})\subseteq F(A)$, it is obvious that $B\subseteq\bigcap_{F(\{a\})\cap A\neq\emptyset}F(\{a\})$ implies $(A,B)\in Y$. On the other hand, if $(C,D)\in Y$ and $C\cap F(\{a\})\neq\emptyset$, then we assume that $b\in C\cap F(\{a\})$, hence by (iii) in the definition of strong operation $b\in F(\{a\})$ implies $F(\{b\})\subseteq F(\{a\})$. It is obvious that $D\subseteq F(C)=\bigcap_{d\in c}F(\{d\})$. By $b\in C$, and $\bigcap_{d\in C}F(\{d\})\subseteq F(\{b\})$ we obtain $D\subseteq F(\{a\})$. It is clear that $\forall A\subseteq\Omega\colon(\emptyset,A),(A,\emptyset)\in Y$. It can be seen that $F=F_Y$. Now, we suppose that there is a full family of strong dependencies Y' so that $F_{Y'}=F$. By the definition of Y and F we obtain $Y'\subseteq Y$. If $(A,B)\in Y$ holds, then $B\subseteq F(A)=F_{Y'}(A)$ holds. By the definition of F_Y we have $(A,B)\in Y'$. Consequently, $Y'=Y$ holds. The proof is complete.

Remark 2.6. Clearly, if F_1 and F_2 are strong operations over Ω $(F_1\neq F_2)$, then $Y_1\neq Y_2$, where $Y_i=\{(A,B)\colon B\subseteq F_i(A)\}$, $i=1,2$.

Based on Theorem 2.3 and Theorem 2.5 the next corollary is obvious.

Corollary 2.7. Let Y be a full family of strong dependencies over Ω. We define the mapping $F\colon P(\Omega)\to P(\Omega)$ as follows:

$$F_Y(A)\colon\{a\in\Omega\colon(A,\{a\})\in Y\}.$$

Let $I_Y=\{F_Y(A)\colon A\in P(\Omega)\}$. Then I_Y is an s-semilattice over Ω. Conversely, if I is an arbitrary s-semilattice over Ω, then there is exactly one full family of strong

dependencies Y such that $I_Y = I$, where

$$Y = \{(A, B)\colon A, B \in P(\Omega), A \neq \emptyset, \exists A_i \in N \setminus \Omega\colon A_i \cap A \neq \emptyset, B \subseteq \bigcap_{\substack{A_i \cap A \neq \emptyset \\ A_i \in N \setminus \Omega}}, N \text{ is the}$$

minimalgenerator of $I\} \cup$

$$\cup \{(A, B)\colon A = \emptyset \text{ or } \exists A_i \in N \setminus \Omega\colon A_i \cap A \neq \emptyset,\ B \in P(\Omega)\}.$$

Corollary 2.7 shows that between full families of strong dependencies and s-semilattices there is a one-to-one correspondence and the s-semilattices determine the full families of strong dependencies.

It is proved (see [2], [3], [4]) that if Y is a full family of strong dependencies over Ω, then there exists a relation R over Ω so that $S_R = Y$.

With the aid of the concept of s-semilattice we can construct for a given full family of strong dependencies Y a simple concrete relation R such that $S_R = Y$.

The equality sets of the relation are defined in [4] as follows:

Definition 2.8. Let $R = \{h_1, \ldots, h_m\}$ be a relation over Ω. For $1 \leqq i < j \leqq m$ denote by E_{ij} the set $\{a \in \Omega\colon h_i(a) = h_j(a)\}$.

Definition 2.9. Let Y be a full family of strong dependencies over Ω. We say that a relation R represents Y iff $S_R = Y$.

We now prove the following theorem which gives a necessary and sufficient condition for a relation to represent a given full family of strong dependencies.

Theorem 2.10. Let Y be a full family of strong dependencies, and $R = \{h_1, \ldots, h_m\}$ be a relation over Ω. Then R represents Y iff for each $a \in \Omega$,

$$F_Y(\{a\}) = \begin{cases} \bigcap_{a \in E_{ij}} E_{ij} & \text{if } \exists E_{ij}\colon a \in E_{ij}, \\ \Omega & \text{otherwise,} \end{cases} \tag{1}$$

where $F_Y(A) = \{a \in \Omega\colon (A, \{a\}) \in Y\}$, and E_{ij} is the equality set of R.

Proof. By Theorem 2.5, $S_R = Y$ if and only if $F_{S_R} = F$ holds. Consequently, first we show that $F_{S_R}(\{a\}) = \bigcap_{a \in E_{ij}} E_{ij}$ if $\exists E_{ij}$; $a \in E_{ij}$, and in other case $F_{S_R}(\{a\}) = \Omega$ holds. Clearly, $F_{S_R}(\{a\}) = \{b \in \Omega\colon \{a\} \xrightarrow[R]{s} \{b\}\}$. According to the definition of strong dependency we know that for any $a \in \Omega$,

$$\{a\} \xrightarrow[R]{s} B \leftrightarrow \{a\} \xrightarrow[R]{f} B,$$

where $a \neq \emptyset$, and $\{a\} \xrightarrow[R]{f} B$ denotes that B functionally depends on $\{a\}$ in R, i.e.

$$(\forall h_i, h_j \in R)(h_i(a) = h_j(a) \to (\forall b \in B)(h_i(b) = h_j(b)))$$

(see [4]). Let us denote by T the set $\{E_{ij}\colon a \in E_{ij}\}$. It is obvious that if $T = \emptyset$, then $\{a\} \xrightarrow[R]{f} \Omega$, i.e. $F_{S_R}(\{a\}) = \Omega$ holds. If $T \neq \emptyset$ holds, then we set $A = \bigcap_{a \in E_{ij}} E_{ij}$.

If $T=E$ (E is the set of all equality sets of R), then it is obvious that $\{a\}\xrightarrow[R]{f}A$. If $T\subset E$ then for E_{ij}: $E_{ij}\notin T$ we obtain $h_i(a)\neq h_j(a)$. Consequently, we have also $\{a\}\xrightarrow[R]{f}A$. Denote by A' the set with the following properties:

(i) $\{a\}\xrightarrow[R]{f}A'$,

(ii) $A'\subset A''$ implies $\{a\}\xrightarrow[R]{f}A''$, i.e. A'' does not functionally depend on $\{a\}$.

It can be seen that $A'=A$. According to the definition of F_{S_R}, we obtain $F_{S_R}(\{a\})=\bigcap_{a\in E_{ij}} E_{ij}$. Thus, if $S_R=Y$ then we have (1). Conversely, if F_Y satisfies (1), then according to the above considerations, for any $a\in\Omega$ we have $F_Y(\{a\})=F_{S_R}(\{a\})$. Because F_Y and F_{S_R} are strong operations over Ω, and by Theorem 2.5 we obtain $\forall A\subseteq\Omega$: $F_{S_R}(A)=F_{S_Y}(A)$. Consequently, $F_Y=F_{S_R}$ holds. The proof is complete.

Definition 2.11. Let R be a relation, and F a strong operation over Ω. We say that the relation R represents F iff $F_{S_R}=F$.

By Theorem 2.10 the next corollary is obvious.

Corollary 2.12. Let F be a strong operation and R a relation over Ω. Then R represents F iff for all $a\in\Omega$,

$$F(\{a\})=\begin{cases}\bigcap_{a\in E_{ij}} E_{ij} & \text{if } \exists E_{ij}\colon a\in E_{ij},\\ \Omega & \text{otherwise.}\end{cases}$$

Clearly, from a relation R we can construct the set of all equality sets of R. Consequently, the following corollary is also obvious.

Corollary 2.13. Let R be a relation and F a strong operation over Ω. Then there is an algorithm which decides whether R represents F or not. This algorithm requires time polynomial in the number of rows and columns of R.

Based on Theorem 2.10 the next proposition is straightforward and so its proof will be omitted.

Proposition 2.14. Let Y be a full family of strong dependencies over Ω. Denote N the minimal generator of s-semilattice I_Y.

Suppose that $N-\Omega=\{B_1,\ldots,B_l\}$. We set $R=\{h_0,h_1,\ldots,h_l\}$ as follows:

$$\text{for all } a\in\Omega\colon h_0(a)=0,$$

$$\text{for each } i\ (i=1,\ldots,l),\ h_i(a)=\begin{cases}0 & \text{if } a\in B_i,\\ i & \text{otherwise.}\end{cases}$$

Then $S_R=Y$ holds.

Clearly, Proposition 2.14 shows that from a given s-semilattice I we can construct a simple concrete relation R such that $I=I_{S_R}$. Because between $\cap$-semilattices and minimal generators there is a one-to-one correspondence, it can be seen that (based on Theorem 2.3, Corollary 2.7, and Proposition 2.14) from the minimal generators of s-semilattices we can construct suitable full families of strong dependencies, strong operations, and relations.

Acknowledgements

The author would like to take this opportunity to express deep gratitude to Professor J. Demetrovics for his help, valuable comments and suggestions.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
KENDE U. 13—17
BUDAPEST, HUNGARY
H—1502

References

[1] ARMSTRONG, W. W., Dependency structures of data base relationships, Information Processing 74, North-Holland (1974) 580—583.
[2] CZÉDLI, G., Függőségek relációs adatbázis modellben, Alkalmazott Matematikai Lapok 6 (1980) 131—143.
[3] DEMETROVICS, J., Relációs adatmodell logikai és strukturális vizsgálata, MTA—SZTAKI Tanulmányok, Budapest, 114 (1980) 1—97.
[4] DEMETROVICS J. and GY. GYEPESI, On the functional dependency and some generalizations of it, Acta Cybernetica, Tom. 5, Fasc. 3 (1981) 295—305.

(Received Okt. 27, 1986)

A finite axiomatization of flowchart schemes

M. BARTHA

1. Introduction

An equational axiomatization of flowchart schemes and their behaviours, being the syntax and semantics of flowchart algorithms, was given by Bloom and Ésik in [B—Es]. This paper is another approach toward the same goal, characterizing the algebra of schemes with a different set of operations. We use separated sum and a constant ε instead of the pairing operation, and replace iteration by an operation called the feedback. The advantage of using this operation is that vector iteration can be done simply by a repeated application of the feedback. This advantage comes out apparently in the form of the axioms that are much simpler than those listed in [B—Es].

Since the algebra of flowchart schemes is sorted by the infinite set $N \times N$ (N denotes the set of all nonnegative integers), to describe our system of axioms we use a scheme of axioms rather than a set of ordinary equational axioms where both sides of the equations are terms built up from constants and variable symbols of fixed sort with the given operations. In our sense such a scheme consists of equation patterns of the following form. The terms on the left and right side are built up from variables of variable sort and subterms denoting algebraic constants. These subterms, however, are allowed to depend on the sort of the variables so that they are uniquely determined by a fixed choice of the sorts of the variables occuring in the whole term. A scheme of axioms is called finite if the number of equation patterns is finite. In this sense the scheme of axioms developed in [B—Es] is infinite. It turns out, however, that a more careful treatment of algebraic constants yields a finite scheme.

As the scheme algebra operations of Bloom and Ésik are easy to derive from our ones (and vice versa), it is possible to approve our scheme of axioms by proving the equivalence of the two axiom systems remaining strictly within the framework of equational logic. This, however, would require a tremendous amount of computation. Instead, we follow the way of constructing suitable normal forms of terms (as it was done also in [B—Es]), which is easy to illustrate by schematic proof-diagrams.

2. The axiomatization of flowchart schemes

We shall consider three classes of $(N\times N)$-sorted algebras, called P, M and S-algebras, respectively. If A is such an algebra, then $A(p, q)$ denotes the underlying set of A corresponding to sort (p, q). The notation $f\colon p\to q$ is introduced with the meaning $f\in A(p, q)$ if A is understood.

A P-algebra is an $(N\times N)$-sorted algebra equipped with the following operations and constants.

Composition: a binary operation which maps $A(p, q)\times A(q, r)$ into $A(p, r)$ for each triple $p, q, r\in N$. Composition is usually denoted by juxtaposition or $\cdot$ if it is intended to be emphasized. Composition is in fact a collection of binary operations $\cdot_{(p,q,r)}$, but the subscript (p, q, r) is omitted for simplicity.

Separated sum: also a binary operation mapping $A(p_1, q_1)\times A(p_2, q_2)$ into $A(p_1+p_2, q_1+q_2)$ for each choice p_1, q_1, p_2, q_2 of nonnegative integers. Separated sum is denoted by $+$.

There are three constants: $1\in A(1, 1)$, $0\in A(0, 0)$ and $x\in A(2, 2)$.

Terms constructed from these constants with the above operations are called base P-terms. Clearly, every base P-term is of sort (p, p) for some $p\in N$. For each $n\in N$ let $t(n)$ denote the base P-term defined recursively as follows.

(i) if $n=0$ or $n=1$, then $t(n)=n$,

(ii) $t(n+1)=t(n)+1$ if $n\geqq 1$.

However, we shall write n instead of $t(n)$ if there is no danger of confusion.

Definition 1. A permutation algebra is a P-algebra satisfying the following equational axioms:

$P1$: $f\cdot(g\cdot h)=(f\cdot g)\cdot h$ for all $f\colon p\to q$, $g\colon q\to r$, $h\colon r\to s$;

$P2$: $f+(g+h)=(f+g)+h$ for all $f\colon p_1\to q_1$, $g\colon p_2\to q_2$, $h\colon p_3\to q_3$;

$P3$: $p\cdot f=f$ and $f\cdot q=f$ for all $f\colon p\to q$;

$P4$: $f+0=f$ and $0+f=f$ for all $f\colon p\to q$;

$P5$: $(f_1\cdot g_1)+(f_2\cdot g_2)=(f_1+f_2)\cdot(g_1+g_2)$ for all $f_i\colon p_i\to q_i$, $g_i\colon q_i\to r_i$, $i=1, 2$;

$P6^*$: $x\cdot x=2$;

$P7^*$: $(1+x)(x+1)(1+x)=(x+1)(1+x)(x+1)$.[1]

For each pair $(p, q)\in N\times N$ let $\Pi(p, q)$ denote the set of all p-ary permutations if $p=q$, else let $\Pi(p, q)=\varnothing$. Define composition and separated sum over the sets $\Pi(p, q)$ in the usual way, and let 1 and 0 be the unique elements of $\Pi(1, 1)$ and $\Pi(0, 0)$, respectively. Interpreting x as the transposition $2\to 2$ we get a P-algebra Π, which is clearly a permutation algebra.

[1] $P6^*$ and $P7^*$ will be replaced by a single axiom called the block permutation axiom. In fact $P6^*$ and $P7^*$ are the weakest special cases of this axiom that are enough to prove that the P-algebra of all permutations is the initial permutation algebra.

A base P-term is called simple if it is equal to k for some $k\in N$, or it is of the form $(i-1)+x+(n-i)$ for some $n\geqq 1$, $i\in[n]=\{1, 2, \ldots, n\}$. (In the latter case if $(i-1)$ or $(n-i)$ is 0, then it is omitted according to $P4$.) The term $(i-1)+x+(n-i)$ will be denoted by $x_n(i)$. Let P denote the collection of axioms $P1, \ldots, P7$. The following remark can be easily proved using only the "magmoid identities" (cf. [AD]) $P1, \ldots, P5$.

Remark. For every base P-term t there exists a base P-term t' which is the composite of a number of simple base P-terms (called the factors of t') and $P\vdash t=t'$, i.e. the identity $t=t'$ is provable from P. t' is said to be in split normal form (s.n.f. for short).

Lemma 1. Let $n\geqq 1$ and $i\in[n-1]$ ($[0]=\emptyset$). Then

$$x_n(1)x_n(2)\cdot\ldots\cdot x_n(n)x_n(i) = x_n(i+1)x_n(1)x_n(2)\cdot\ldots\cdot x_n(n)$$

is provable from P.

Proof. (See also Fig. 1 for the case $n=3$, $i=1$.)

$$x_n(1)x_n(2)\cdot\ldots\cdot x_n(n)x_n(i) = x_n(1)\cdot\ldots\cdot x_n(i)x_n(i+1)x_n(i)x_n(i+2)\cdot\ldots\cdot x_n(n) =$$
$$= (\text{by } P7) = x_n(1)\cdot\ldots\cdot x_n(i-1)x_n(i+1)x_n(i)x_n(i+1)\cdot\ldots\cdot x_n(n) =$$
$$= x_n(i+1)x_n(1)\cdot\ldots\cdot x_n(n).$$

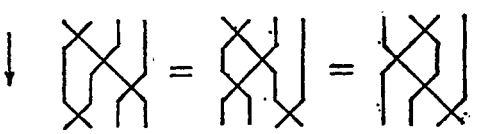


Fig. 1. Proof of Lemma 1 on an example

In two steps of the above derivation we used the obvious identity $x_n(k)x_n(l)=x_n(l)x_n(k)$, where $1\leqq k<l-1<n$.

For a base P-term t let $|t|$ denote the value of t in Π. (In other words $|\ |$ is the unique homomorphism of the initial P-algebra into Π.) Since every permutation is expressible as a composite of permutations of the form $|x_n(i)|$, the following proposition says that the initial permutation algebra is Π.

Proposition 1. Let t and t' be base P-terms. If $|t|=|t'|$, then $P\vdash t=t'$.

Proof. By the Remark we can assume that t and t' are both in s.n.f. Listing the factors of t in reverse order we get a term t^{-1} such that $P\vdash t^{-1}t=n$ for appropriate $n\in N$. If we can prove $t't^{-1}=n$, then we get the required proof: $t'=t'(t^{-1}t)=(t't^{-1})t=t$. Hence it is enough to show that if $|t|=|n|$ for some base P-term t in s.n.f., then $P\vdash t=n$. We follow an induction on n. If $n\leqq 1$, then the statement is trivial. Let $n\geqq 2$. If none of the factors of t is equal to $x_{n-1}(1)$, then $t=1+t'$, and the induction hypothesis works for t'. If $x_{n-1}(1)$ occurs in t, then assume indirectly that $P\not\vdash t=n$, and the length of t (i.e. the number of the factors of t) is minimal. Split $t=\alpha x_{n-1}(1)\beta$ so that $x_{n-1}(1)$ should not occur in α. By

Lemma 1 and the assumption that the length of t is minimal we get that $P\vdash x_{n-1}(1)\beta=\gamma x_{n-1}(1)x_{n-1}(2)\cdot\ldots\cdot x_{n-1}(j)$ for some $j\in[n-1]$, where $x_{n-1}(1)$ does not occur even in γ. We conclude that $P\vdash t=\alpha\gamma x_{n-1}(1)\ldots x_{n-1}(j)$ which is a contradiction, since in this case $|t|(1)=j+1>1$.

Corollary 1. The initial permutation algebra is Π.

In the light of Proposition 1, when working in permutation algebras we identify a base P-term t with the permutation $|t|$.

The following definition is adopted from [B—Es]. Let s be a finite sequence $(n_1, \ldots, n_r)$ of nonnegative integers and suppose that $\alpha\colon r\to r$ is a permutation. Let n be the sum of the numbers n_i.

Definition 2. $\alpha\#s\colon n\to n$ is the permutation which takes a number in $[n]$ of the form

$$n_1+\ldots+n_k+j,$$

where $j\in[n_{k+1}]$, to the number $y+j$, where y is the sum of all numbers n_i such that $\alpha(i)<\alpha(k+1)$.

From now on we drop the axioms $P6^*$ and $P7^*$, replacing them by the stronger block permutation axiom of [ES]:

$$P6;\ f_1+f_2 = x\ \#\ (p_1,p_2)\cdot(f_2+f_1)\cdot x\ \#\ (q_2,q_1) \quad \text{for all}$$

$$f_i\colon p_i\to q_i,\quad i=1,2.$$

Assume that $x\#(p_1,p_2)$ and $x\#(q_2,q_1)$ are represented in $P6$ by base P-terms in a minimal length s.n.f. Then we see that $P6^*$ and $P7^*$ are indeed consequences of $P6$. (Take $f_1=f_2=1$ in the case of $P6$, and $f_1=x$, $f_2=1$ for $P7$.) Since $P6$ is also valid in Π, Π remains initial. Now the following lemma is true in every permutation algebra.

Lemma 2. Let $\alpha\colon r\to r$ be a permutation and $f_i\colon p_i\to q_i$ for each $i\in[r]$. Then

$$\sum_{i=1}^{r} f_i = (\alpha^{-1}\#s_1)\cdot\Big(\sum_{i=1}^{r}\cdot f_{\alpha(i)}\Big)\cdot(\alpha\#s_2),$$

where $s_1=(p_1,\ldots,p_r)$ and $s_2=(q_{\alpha(1)},\ldots,q_{\alpha(r)})$.

Proof. Easy induction on the length of a base P-term in s.n.f. representing α.

An M-algebra is an $(N\times N)$-sorted algebra having all the operations and constants of P-algebras and two further constats: ε of sort $(2,1)$ and 0_1 of sort $(0,1)$. As in the case of P-algebras, base M-terms are those built up from the constants using the given operations. Define base M-terms ε_n and 0_n for each $n\in N$ as follows.

(i) $\varepsilon_0=0_1,\quad \varepsilon_1=1,\quad \varepsilon_2=\varepsilon,\quad 0_0=0,\quad 0_2=0_1+0_1;$

(ii) if $n\geqq 2$ then $\varepsilon_{n+1}=(\varepsilon_n+1)\cdot\varepsilon,\quad 0_{n+1}=0_n+0_1.$

Definition 3. A mapping algebra is an M-algebra satisfying the identities belonging to P and the following ones.

$M1$: $(\varepsilon+1)\cdot\varepsilon = (1+\varepsilon)\cdot\varepsilon$;

$M2$: $x\cdot\varepsilon = \varepsilon$;

$M3$: $(1+0_1)\cdot\varepsilon = 1$.

For $(p, q)\in N\times N$ let $\theta(p, q)$ be the set of all mappings of $[p]$ into $[q]$. Let 0_1 and ε be the unique elements of $\theta(0, 1)$ and $\theta(2, 1)$, respectively, and interpret the P-algebra operations and constants over the sets $\theta(p, q)$ as an obvious extension of their interpretation in Π. In this way we get the M-algebra θ, which is clearly a mapping algebra as well.

A base M-term is called simple if it is a simple base P-term, or it is one of the forms

(i) $(i-1)+0_1+(n-i)$, or

(ii) $(i-1)+\varepsilon+(n-i)$

for some $n\geqq 1$, $i\in[n]$. Let M denote the collection of axioms $M1$, $M2$, $M3$. As in the case of base P-terms, for every base M-term t these exists a base M-term t' such that t' is the composite of simple base M-terms and $P\vdash t=t'$. Moreover, by $P6$ it is possible to rearrange the factors of t' in such a way that $P\cup M\vdash t'=\alpha\beta$, where α is a base P-term in s.n.f., but none of the factors of β is a simple base P-term (exept when $\beta=k$ for some $k\in N$). But then $P\cup M\vdash\beta=\varepsilon_{j_1}+\ldots+\varepsilon_{j_m}$ for some non-negative integers m, $j_1, \ldots, j_m$.

For a base M-term t let $|t|$ denote the value of t in θ. The above reasoning together with Proposition 1 yields the following result.

Proposition 2. Let t and t' be base M-terms. If $|t|=|t'|$, then $P\cup M\vdash t=t'$. Equivalently, the initial mapping algebra is θ.

As in the case of permutation algebras, when working in mapping algebras we identify a base M-term t with the mapping $|t|$.

Let $\alpha\colon p\to q$ be a mapping and $B\subseteqq[q]$. We say that α is onto B if $\alpha^{-1}(j)\neq\emptyset$ for any $j\in B$. If $\alpha_i\colon p_i\to q$, $i=1, 2$ are mappings, then define their pairing $\langle\alpha_1, \alpha_2\rangle\colon p_1+p_2\to q$ as

$$\langle\alpha_1, \alpha_2\rangle(i) = \begin{cases}\alpha_1(i) & \text{if} \quad i\in[p_1]\\ \alpha_2(i-p_1) & \text{if} \quad i\in[p_1+p_2]-[p_1].\end{cases}$$

An S-algebra has one further unary operation beyond the M-algebra operations and constants. This operation will be called the feedback and denoted by $\dagger$. In an S-algebra the feedback maps $A(1+p, 1+q)$ into $A(p, q)$ for each pair $(p, q)\in N\times N$.

Let Σ be a doubly ranked set. (Recall from [B—Es] that

$$\Sigma = \{\Sigma(p, q)|(p, q)\in N\times N\},$$

where the sets $\Sigma(p, q)$ are pairwise disjoint.) A Σ-flowchart scheme with p begins and q exist consists of:

(i) A finite nonempty set V of labelled vertices, where the labels belong to the union of four, pairwise disjoint sets:

$$(\cup\Sigma)\cup\{b_i|i\in[p]\}\cup\{\mathrm{ex}_j|j\in[q]\}\cup\{\perp\}.$$

For each $i\in[p]$ and $j\in[q]$ there exists exactly one vertex labelled by b_i, called the i-th begin vertex, and exactly one vertex labelled by ex_j, the j-th exit vertex. Moreover, exactly one vertex, the loop vertex is labelled by $\perp$. For each $v\in V$ denote v_{in} and v_{out} the following sets of so called "signed vertices". Let α be the label of v. If $\alpha\in\Sigma(r, s)$, then

$$v_{\mathrm{in}} = \{(v, i)|i\in[r]\} \quad \text{and} \quad v_{\mathrm{out}} = \{(v, j)|j\in[s]\}.$$

If $\alpha=\mathrm{ex}_j$ or $\alpha=\perp$, then $v_{\mathrm{in}}=\{(v, 1)\}$ and $v_{\mathrm{out}}=\emptyset$, else (i.e. if $\alpha=b_i$) $v_{\mathrm{in}}=\emptyset$ and $v_{\mathrm{out}}=\{(v, 1)\}$. Signed vertices belonging to begin, exit and loop vertices will be identified with their label.

(ii) A mapping E of $V_{\mathrm{out}}=\cup(v_{\mathrm{out}}|v\in V)$ into $V_{\mathrm{in}}=\cup(v_{\mathrm{in}}|v\in V)$. E represents the edges of the scheme, and in this sense we consider Σ-flowchart schemes as directed graphs.

Define the S-algebra operations on Σ-schemes as follows.

— The composition of schemes $F: p\to q$ and $G: q\to r$ is constructed in three steps.

1) Take the disjoint union of the graphs of F and G.

2) Direct each edge of F ending in any exit vertex of F, say ex_j to the signed vertex pointed by $E(b_j)$ in G.

3) Identify the lopp vertices of F and G, and delete the exists of F as well as the begins of G (together with all the incoming and outgoing edges, of course).

— The sum of schemes $F_1: p_1\to q_1$ and $F_2: p_2\to q_2$ is taken as follows.

1) Take the disjoint union of F_1 and F_2.

2) Relabel each begin vertex of F_2 from b_i to b_{p_1+i} and each exit vertex of F_2 from ex_j to ex_{q_1+j} $(i\in[p_2], j\in[q_2])$.

3) Identify their loop vertices.

— If F is a scheme $1+p\to 1+q$, then $\dagger F$ is constructed as follows.

1) Direct each edge of F ending in ex_1 to $E(b_1)$ if $E(b_1)\neq\mathrm{ex}_1$, else to $\perp$.

2) Delete vertices b_1 and ex_1, and relabel b_{i+1} and ex_{j+1} as b_i (resp. ex_j) for $i\in[p]$, $j\in[q]$.

— The interpretation of the constants is shown by Fig. 2.

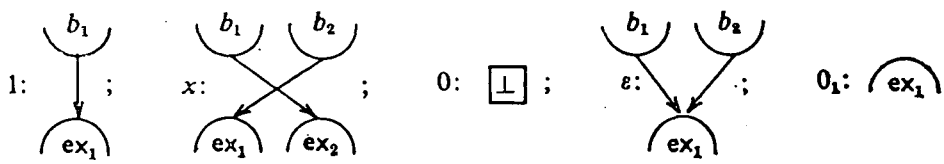


Fig. 2. Interpretation of the constants as schemes

The loop vertex is omitted on the figure (exept for 0).

Definition 4. A scheme algebra is an S-algebra satisfying the identities $P \cup M$ and the following ones (the collection of these identities will be denoted by PMS).

$S1$: $\dagger(f_1+f_2) = \dagger f_1 + f_2$ for $f_1\colon 1+p_1 \to 1+q_1$, $f_2\colon p_2 \to q_2$;

$S2$: $\dagger\dagger((x+p)f) = \dagger\dagger(f(x+q))$ for $f\colon 2+p \to 2+q$;

$S3$: $\dagger(f(1+g)) = (\dagger f)g$ for $f\colon 1+p \to 1+q$, $g\colon q \to r$;

$S4$: $\dagger((1+g)f) = g \cdot \dagger f$ for $f\colon 1+q \to 1+r$, $g\colon p \to q$;

$S5$: $\dagger 1 = 0$ and $\varepsilon \cdot \perp = \perp + \perp$, where $\perp = \dagger\varepsilon$;

$S6$: $\dagger x = 1$.

It is easy to see that Σ-schemes together with the operations and constants defined above form a scheme algebra, which will be denoted by $\mathrm{Sch}(\Sigma)$.

Lemma 3. If $\alpha\colon 1+p \to 1+q$ is a mapping with $\alpha(1) \neq 1$, then there exists a mapping $\beta\colon p \to q$ such that $PMS \vdash \dagger\alpha = \beta$.

Proof. Split α into the form

$$(1+\beta_1)(x+r)(1+\beta_2),$$

where $\beta_1\colon p \to 1+r$ and $\beta_2\colon 1+r \to q$ are appropriate mappings. Then

$$\dagger\alpha = (\text{by } S3 \text{ and } S4) = \beta_1 \cdot \dagger(x+r)\beta_2 = (\text{by } S1 \text{ and } S6) = \beta_1\beta_2.$$

Claim. The following identities are valid in every scheme algebra.

$S1^*$: $\dagger^l(f_1+f_2) = \dagger^l f_1 + f_2$ for $f_1\colon l+p_1 \to l+q_1$, $f_2\colon p_2 \to q_2$.

($\dagger^l$ denotes the l-fold application of $\dagger$.)

$S3^*$: $\dagger^l(f(l+g)) = (\dagger^l f) \cdot g$ for $f\colon l+p \to l+q$, $g\colon q \to r$.

$S4^*$: $\dagger^l((l+g)f) = g \cdot \dagger^l f$ for $f\colon l+q \to l+r$, $g\colon p \to q$.

Proof. Trivial.

$X1$: $\dagger^l((\alpha+p)f(\alpha^{-1}+q)) = \dagger^l f$ for $f\colon l+p \to l+q$ and permutation α.

Proof. Put α into s.n.f., and apply $S2$ with $S3^*$ and $S4^*$ repeatedly.

$X2$: $\dagger^{l_1+l_2}((l_1+x\,\#(l_2,p_1)+p_2)(f_1+f_2)(l_1+x\,\#(q_1,l_2)+q_2)) =$

$$= \dagger^{l_1} f_1 + \dagger^{l_2} f_2 \quad \text{for } f_i\colon l_i+p_i \to l_i+q_i,\ i = 1,2.$$

Proof. A schematic derivation is shown by Fig. 3.

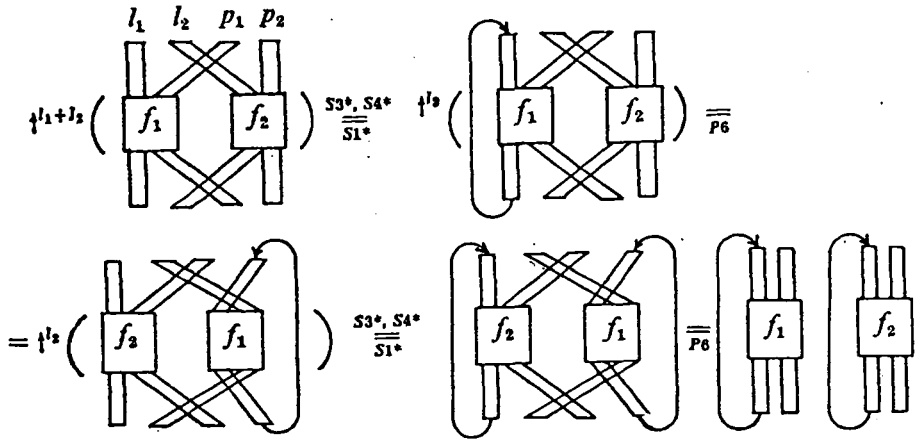


Fig. 3. Schematic proof of X2

The same proof formally:

$$\uparrow^{l_1+l_2}((l_1+x\,\#(l_2,p_1)+p_2)(f_1+f_2)(l_1+x\,\#(q_1,l_2)+q_2)) =$$
$$= \uparrow^{l_2}(\uparrow^{l_1}((l_1+x\,\#(l_2,p_1)+p_2)((f_1+f_2)(l_1+x\,\#(q_1,l_2)+q_2)))) = (S4^*)$$
$$= \uparrow^{l_2}((x\,\#(l_2,p_1)+p_2)\cdot\uparrow^{l_1}((f_1+f_2)(l_1+x\,\#(q_1,l_2)+q_2))) = (S3^*)$$
$$= \uparrow^{l_2}((x\,\#(l_2,p_1)+p_2)\cdot\uparrow^{l_1}(f_1+f_2)(x\,\#(q_1,l_2)+q_2)) = (S1^*)$$
$$= \uparrow^{l_2}((x\,\#(l_2,p_1)+p_2)(\uparrow^{l_1}f_1+f_2)(x\,\#(q_1,l_2)+q_2)) = (P6)$$
$$= \uparrow^{l_2}((l_2+x\,\#(p_1,p_2))(f_2+\uparrow^{l_1}f_1)(l_2+x\,\#(q_2,q_1))) = (S4^*, S3^*, S1^*)$$
$$= (x\,\#(p_1,p_2))(\uparrow^{l_2}f_2+\uparrow^{l_1}f_1)(x\,\#(q_2,q_1)) = (P6) = \uparrow^{l_1}f_1+\uparrow^{l_2}f_2.$$

In the sequel we shall omit the tedious formal proofs restricting ourselves to the corresponding schematic ones.

X3a: $\uparrow^q(x\,\#(p,q)+l)(f+g)) = (f+l)g$ for

$f\colon p \to q, \quad g\colon q+l \to r,$

X3b: $\uparrow^q((g+f)(x\,\#(r,q)+l)) = f(g+l)$ for

$f\colon p \to q+l, \quad g\colon q \to r.$

Proof. Both cases *a* and *b* can be proved in a similar way, so we only prove *X3a*. (Dotted lines indicate composition in Fig. 4.)

X4: $\uparrow(f(\varepsilon+q)) = \uparrow^2((\varepsilon+p)f)$ for $f\colon 1+p \to 2+q.$

Proof. See Fig. 5.

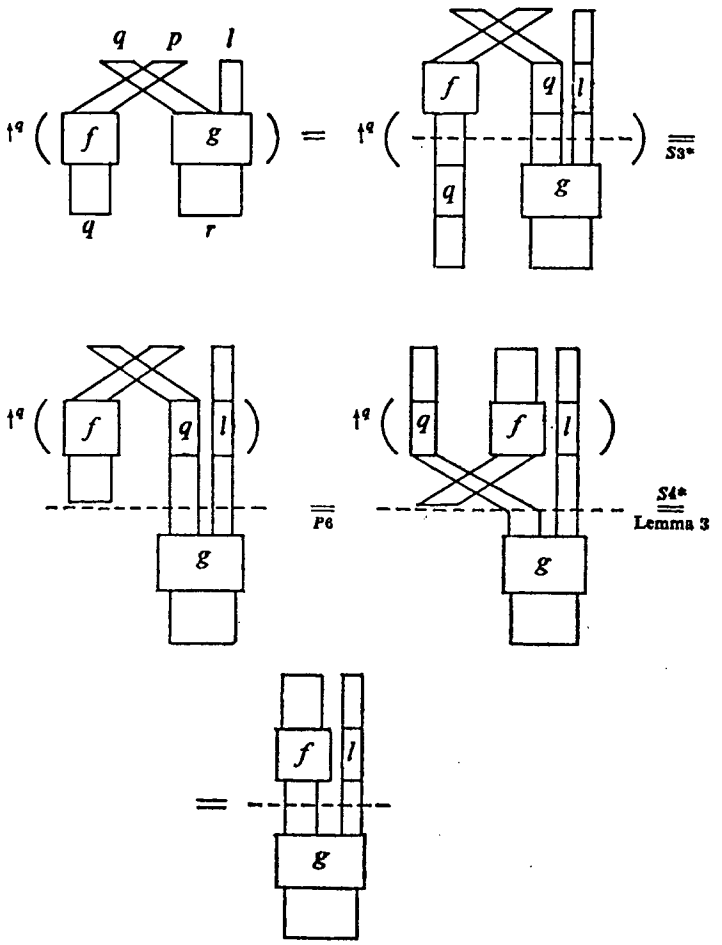


Fig. 4. Proof of X3a

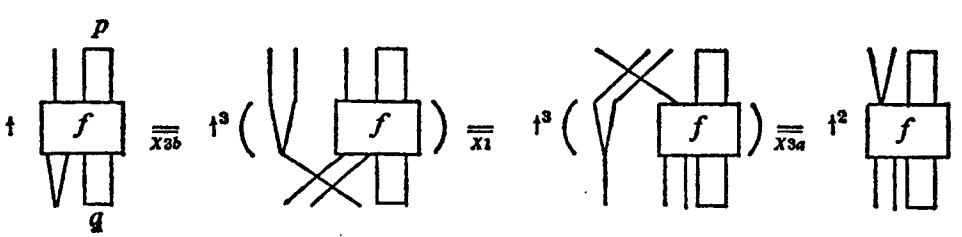


Fig. 5. Proof of X4

Lemma 4. Let $\alpha: r \to 1+p$ and $\beta: 1+q \to s$ be mappings, $f: p \to q$. In every scheme algebra we have

a) if $\alpha(1)=1$ and $\beta(1)\neq 1$, then

$$\dagger(\alpha(1+f)\beta) = \alpha'(1+f)\beta';$$

b) if $\beta(1)=1$ and $\alpha(1)\neq 1$, then

$$\dagger(\alpha(1+f)\beta) = \dagger(\alpha'' f \beta'');$$

c) if $\alpha(1)=\beta(1)=1$, then

$$\dagger(\alpha(1+f)\beta) = \dagger(\alpha'''(\perp + f)\beta''')$$

for appropriate mappings $\alpha', \alpha'', \alpha''', \beta', \beta'', \beta'''$; moreover $\alpha'''(1)=1$.

Proof.

Case a: By assumption α can be written in the form $\alpha=(1+\alpha')(\varepsilon+p)$. Then

$$\alpha(1+f)\beta = (1+\alpha')(\varepsilon+p)(1+f)\beta = (1+\alpha')(2+f)(\varepsilon+q)\beta =$$
$$= (1+\alpha'(1+f))(\varepsilon+q)\beta.$$

Hence by $S4$:

$$\dagger(\alpha(1+f)\beta) = \alpha'(1+f)\cdot\dagger((\varepsilon+q)\beta).$$

Since $\beta(1)\neq 1$, Lemma 3 says that $\dagger((\varepsilon+q)\beta)=\beta'$ for some mapping β'.

Case b: In this case β can be written in the form

$$\beta = (1+\beta'')(\varepsilon+s-1),$$

so

$$\dagger(\alpha(1+f)\beta) = \dagger((\alpha(1+f)\beta''(\varepsilon+s-1)) = (\text{by } X4)$$
$$= \dagger^2((\varepsilon+r-1)\alpha(1+f)(1+\beta'')) = \dagger(\dagger((\varepsilon+r-1)\alpha(1+f\beta''))) = (\text{by } S3)$$
$$= \dagger(\dagger((\varepsilon+r-1)\alpha)f\beta'') = (\text{by Lemma 3}) = \dagger(\alpha'' f\beta'').$$

Case c: As in the previous case we have

$$\dagger(\alpha(1+f)\beta) = \dagger(\dagger((\varepsilon+r-1)\alpha)f\beta'')$$

but now

$$(\varepsilon+r-1)\alpha = (\varepsilon+r-1)(1+\alpha')(\varepsilon+p) = (1+r)(2+\alpha')(\varepsilon_3+p);$$
$$\dagger((\varepsilon+r-1)\alpha) = (\text{by } S4) = (1+\alpha')\dagger(\varepsilon_3+p) = (1+\alpha')(\varepsilon\cdot\perp+p) =$$
$$= (1+\alpha')(\varepsilon+p)(\perp+p).$$

Putting $\alpha'''=(1+\alpha')(\varepsilon+p)$ and $\beta'''=\beta''$ we get:

$$\dagger(\alpha(1+f)\beta) = \dagger(\alpha'''(\perp+p)f\beta''') = \dagger(\alpha'''(\perp+f)\beta''').$$

We call Σ-terms those S-terms that are built up from the elements of Σ considered as atomic terms (recall that Σ is a doubly ranked alphabet) and from the constants, using the given operations. Since the S-algebra of Σ-terms is freely generated by Σ, each homomorphism of it into the algebra $\mathrm{Sch}(\Sigma)$ is uniquely determined by its restriction to Σ. Let $|\ |$ be the homomorphism determined by the mapping shown by Fig. 6.

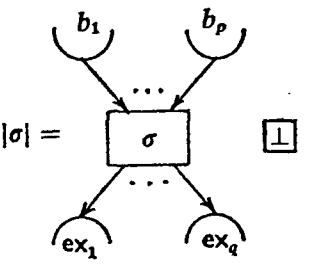


Fig. 6. $\sigma \in \Sigma(p, q)$ as an atomic scheme

A Σ-term t is said to be in weak normal form (w.n.f.) if

$$t = \uparrow^l(\alpha(a_1 + \ldots + a_n)\beta),$$

where

(i) $a_i \in \Sigma$ or $a_i = 1$ or $a_i = \perp$ for each $i \in [n]$,

and there exists at least one $j \in [n]$ such that $a_j = \perp$.

(ii) α and β are mappings of appropriate sort.

Lemma 5. For each Σ-term t there exists a Σ-term t' in w.n.f. such that $t = t'$ is provable from *PMS*.

Proof. Induction on the structure of t. If t is a constant, then its simplest w.n.f. is one of the following: $1 = (1 + 0_1)(1 + \perp)$, $0 = 0_1 \cdot \perp$, $x = (x + 0_1)(2 + \perp)$, $0_1 = 0_1 \cdot \perp \cdot 0_1$, $\varepsilon = (\varepsilon + 0_1)(1 + \perp)$. These identities are easy to prove. If $t = \sigma \in \Sigma(p, q)$, then its w.n.f. is $(p + 0_1)(\sigma + \perp)$. Let $t = t_1 \ \mathrm{op} \ t_2$, where $\mathrm{op} = +$ or $\cdot$, and let t_1' and t_2' be some w.n.f.-s of t_1 and t_2, respectively. If $\mathrm{op} = +$, then we get a w.n.f. of t by applying $X2$ for $f_1 = t_1'$ and $f_2 = t_2'$. If $\mathrm{op} = \cdot$, then apply $X3$ with $l = 0$ (both cases a and b are appropriate), and then $X2$ together with $S3^*$ or $S4^*$ to get the required w.n.f. of t. For $t = \uparrow t'$ the induction step is trivial.

Definition 5. A Σ-term $t\colon p \to q$ is said to be in normal form (n.f.) if

$$t = \uparrow^l(\langle \alpha_1 + 0_{k+1}, \alpha_2 \rangle \cdot (\sum_{i=1}^{n} \sigma_i + \perp + k) \cdot \langle \beta_2, 0_l + \beta_1 \rangle),$$

where

(i) $n \geqq 0, \sigma_i \in \Sigma(r_i, s_i)$ for each $i \in [n]$,

$$\sum_{i=1}^{n} r_i = r, \quad \sum_{i=1}^{n} s_i = s;$$

(ii) $\alpha_1, \alpha_2, \beta_1$ and β_2 are mappings such that

$\alpha_1\colon l \to r + 1$ and $\beta_1\colon k \to q$ are injective and monotonic,

$\alpha_2\colon p \to r + 1 + k$ is onto $[r + 1 + k] - [r + 1]$;

$\beta_2\colon s \to l + q$ is onto $[l]$.

Lemma 6. For each Σ-term $t\colon p \to q$ there exists a Σ-term t' in n.f. such that $t = t'$ is provable from *PMS*.

Proof. By Lemma 5 we can assume that t is in w.n.f., i.e. $t=\dagger^l(\alpha(a_1+\ldots+a_m)\beta)$. Moreover, by $P6$ and $S5$ we can assume that for some $n<m$ we have: $a_i\in\Sigma$ if $i\in[n]$, $a_{n+1}=\perp$ and $a_j=1$ for each $n+1\leqq j\leqq m$. Let $k=m-n-1$. We prove by induction on the number $k+l$. If $k+l=0$, then we have nothing to prove. For $k+l>0$ we distinguish two cases. By assumption, α: $l+p\to r+1+k$ and β:$s+k\to$ $\to l+q$ for some $r,s\in N$.

Case a: $l>0$ and one of conditions $(*)$, $(**)$ or $(***)$ below is satisfied: Suppose that for some $j\in[k]$

$$(\alpha^{-1}(r+1+j)\cup\beta(s+j))\cap[l]\neq\emptyset. \qquad (*)$$

By $P6$ and $X1$

$$t=\dagger^l(\alpha'(1+a_1+\ldots+a_{j-1}+a_{j+1}+\ldots+a_m)\beta')$$

is provable for some α' and β' such that $\alpha'(1)=1$ or $\beta'(1)=1$. Using Lemma 4 we obtain a simpler w.n.f. of t by decreasing the number k or l, which makes the induction hypothesis applicable. If this type of reduction cannot be applied, i.e. condition $(*)$ does not holds for any $n+1<j\leqq m$, then α and β can be written in the form

$$\alpha=\langle\alpha_1+0_{k+1},\alpha_2\rangle \quad\text{and}\quad \beta=\langle\beta_2,0_l+\beta_1\rangle,$$

where α_1: $l\to r+1$, α_2: $p\to r+1+k$, β_1: $k\to q$ and β_2: $s\to l+q$ are appropriate mappings. Now suppose that there are distinct integers

$$1\leqq i_1<i_2\leqq l \quad\text{such that}\quad \alpha_1(i_1)=\alpha_1(i_2). \qquad (**)$$

By $X1$ we can assume that $i_1=1$ and $i_2=2$. Using $X4$ we can decrease l making the induction hypothesis applicable. In this way α_1 can be made injective. It is also easy to see that if

$$\beta_2 \text{ is not onto } [l], \qquad (***)$$

then the feedback counter l can be decreased again.

Case b: $l=0$ or none of $(*)$, $(**)$ and $(***)$ is satisfied.

In this case if α_2 were not onto $[r+1+k]-[r+1]$, then k could be decreased trivially, moreover any duplication of β_1 could be "lifted" to α_2 causing again the number k to be decreased.

Thus, we have seen that in any case when the induction hypothesis cannot be applied we have all the conditions of the n.f. satisfied, except monotonicity of α_1 and β_1. However, this can also be adjusted by the application of $X1$ and $P6$, so the proof is complete.

Theorem 1. Let t and t' be Σ-terms. If $|t|=|t'|$, then $t=t'$ is provable from *PMS*.

Proof. By Lemma 6 we can assume that t and t' are in n.f. Normal form of Σ-terms was defined in such a way that if t and t' were not identical, then the only difference between them should appear in the order of the atomic terms occuring in the sum $\sum_{i=1}^{n}\sigma_i$. In this case, however, an application of Lemma 2 with an appropriate permutation α will make them identical.

This theorem and the following corollary are the main results of the paper.

Corollary 2. $\mathrm{Sch}(\Sigma)$ is the free scheme algebra generated by Σ.

3. Connections to the same result of [B—Es]

In [B—Es] schemes are axiomatized as algebras equipped with the following operations and constants.

Composition, $\cdot$: $A(n,p)\times A(p,q)\to A(n,q)$;

Pairing, $\langle,\rangle$: $A(n,q)\times A(p,q)\to A(n+p,q)$;

Iteration, †: $A(n,n+p)\to A(n,p)$;

$\pi_p^i\in A(1,p)$ for each $p\in N$, $i\in[p]$;

$0_p\in A(0,p)$ for each $p\in N$.

Let us call algebras of hist type BS-algebras, and to make a temporary distinction call the BS-type scheme algebras of [B—Es] B-scheme algebras.

In an arbitrary scheme algebra A we can introduce the BS-algebra operations as derived ones in the following way.

Composition: adopted as a basic operation;

Pairing: for $f\colon n\to q$ and $g\colon p\to q$ let

$$\langle f,g\rangle=(f+g)\cdot w_2(q),$$

where $w_k(q)\colon kq\to q$ is the mapping which takes $(j-1)q+i$ to i for each $j\in[k]$, $i\in[q]$;

Iteration: for $f\colon n\to n+p$ let

$$f^\dagger=\dagger^n(w_2(n)f);$$

$$\pi_p^i=0_{i-1}+1+0_{p-i} \quad \text{and} \quad 0_p \text{ is adopted.}$$

It is straightforward to check that if A is a free scheme algebra, then the above derived interpretation of the BS-algebra operations coincides with their original interpretation considering A as a free B-scheme algebra. From well-known results of universal algebra it follows that every scheme algebra equipped with these BS-algebra operations becomes a B-scheme algebra.

Now let A be a B-scheme algebra. We can derive the S-algebra operations in A as follows.

$1=\pi_1^1$; $x=\langle\pi_2^2,\pi_2^1\rangle$; $\varepsilon=\langle\pi_1^1,\pi_1^1\rangle$; 0_1: adopted;

Composition: adopted;

Sum: for $f_1\colon p_1\to q_1$, $f_2\colon p_2\to q_2$ let

$$f_1+f_2=\langle f_1\langle\pi^1_{q_1+q_2},\ldots,\pi^{q_1}_{q_1+q_2}\rangle,f_2\langle\pi^{q_1+1}_{q_1+q_2},\ldots,\pi^{q_1+q_2}_{q_1+q_2}\rangle\rangle;$$

Feedback: for $f\colon 1+p\to 1+q$ let

$$\dagger f=(0_1+p)(f(1+0_p+q))^\dagger.$$

Repeating the previous argument for free algebras we can see that A equipped with the above defined S-algebra operations becomes a scheme algebra. Thus, we can state

Theorem 2. The equational class of all scheme algebras is equivalent to that of all B-scheme algebras.

4. Algebraic and iteration theories

Roughly speaking an algebraic theory (theory for short) is a BS-algebra without iteration satisfying the following equational axioms.

$TH1$: $f\cdot(g\cdot h)=(f\cdot g)\cdot h$ for all $f\colon n\to p$, $g\colon p\to q$, $h\colon q\to r$,

$TH2$: $p\cdot f=f=f\cdot q$ for all $f\colon p\to q$ (p denotes the term $\langle\pi_p^1,\ldots,\pi_p^p\rangle$),

$TH3$: $\langle\langle f,g\rangle,h\rangle=\langle f,\langle g,h\rangle\rangle$ for all $f\colon m\to q$, $g\colon n\to q$, $h\colon p\to q$,

$TH4$: $\langle f,0_q\rangle=f=\langle 0_q,f\rangle$ for all $f\colon p\to q$,

$TH5$: $\pi_p^i\langle f_1,\ldots,f_p\rangle=f_i$ for all $f_1,\ldots,f_p\colon 1\to q$, $i\in[p]$,

$TH6$: $\langle\pi_p^1 f,\ldots,\pi_p^p f\rangle=f$ for all $f\colon p\to q$.

We would like to extend our system of axioms PMS in such a way that in the corresponding smaller equational class each algebra should derive a B-scheme algebra which is a theory. We claim that the following two axioms are sufficient.

$Th1$: $0_1\cdot f=0_q$ for all $f\colon 1\to q$,

$Th2$: $w_p(p)\cdot f=\left(\sum_{i=1}^{p} f\right)\cdot w_p(q)$ for all $f\colon p\to q$

(recall that $w_p(q)\colon pq\to q$ takes $(j-1)q+i$ to i for each $j\in[p]$, $i\in[q]$). Indeed, the derived correspondents of $TH1$—$TH4$ follows already from $P\cup M$, so we only have to prove $TH5$ and $TH6$. The derived form of $TH5$ in S-algebras in the following:

$$(0_{i-1}+1+0_{p-i})(f_1+\ldots+f_p)w_p(q)=f_i.$$

By $Th1$ the left-hand side reduces to

$$(0_{(i-1)q}+f_i+0_{(p-i)q})w_p(q),$$

which is clearly equal to f_i.

Concerning $TH6$ we have to prove that

$$((1+0_{p-1})f+\ldots+(0_{p-1}+1)f)w_p(q)=f.$$

Manipulating the left-hand side using *Th*2 we obtain:

$$(\sum_{i=1}^{p}((0_{i-1}+1+0_{p-i})f))w_p(q) = (\sum_{i=1}^{p}(0_{i-1}+1+0_{p-i}))(\sum_{i=1}^{p}f)w_p(q) =$$

$$= (\sum_{i=1}^{p}(0_{i-1}+1+0_{p-i}))w_p(p)f = f.$$

Unfortunately I did not succeed in an essential simplification of Ésik's "commutativity" axioms (cf. [Es]) for iteration theories:

$$IT\colon \langle \pi_m^1 \varrho f(\varrho_1+p), \ldots, \pi_n^n \varrho f(\varrho_n+p)\rangle^\dagger = \varrho(f(\varrho+p))^\dagger,$$

where $f\colon n\to m+p$, $\varrho\colon m\to n$ is a surjective mapping and $\varrho_i\colon m\to m$ are also mappings satisfying $\varrho_i\varrho=\varrho$. In our sense *IT* is an infinite scheme of axioms, moreover, Ésik has proved that it cannot be replaced by any finite scheme, see [Es1].

BOLYAI INSTITUTE
A. JÓZSEF UNIVERSITY
ARADI VÉRTANÚK TERE 1
SZEGED, HUNGARY
H—6720

References

[B—Es] BLOOM, S. L., Z. ÉSIK, Axiomatizing schemes and their behaviours, J. Comp. System Sci., 31/3 (1985), 375—393.

[ES] ELGOT, C. C., J. SHEPHERDSON, An equational axiomatization of reducible flowchart schemes, in Calvin C. Elgot, Selected papers, S. L. Bloom, ed., Springer-Verlag, 1982.

[Es] ÉSIK, Z. Identities in iterative and rational theories, Comput. Languages 14 (1980), 183—207.

[Es1] ÉSIK, Z., Independence of the equational axioms of iteration theories, to appear.

[AD] ARNOLD, A., M. DAUCHET, Theorie des magmoïdes, RAIRO Inform. Theor. 12 (1978) 235—257 and 13 (1979) 135—154.

(Received Oct. 8, 1986)

О ВЕРОЯТНОСТНЫХ МЕТОДАХ ОПТИМИЗАЦИИ НА ДИСКРЕТНЫХ МНОЖЕСТВАХ

С. В. Яковлев

Пусть X дискретное множество, на котором определен функционал $\varkappa$: $X \to R^1$. Требуется найти

$$x^* = \arg\min_{x \in X} \varkappa(x) \tag{1}$$

Рандомизируем задачу (1). Отождествим пространство элементарных событий с X. Пусть $\mathscr{B}_X$ — σ-алгебра подмножеств X (в частности возможно $\mathscr{B}_X = 2^X$), а P_X — вероятностная мера на $\mathscr{B}_X$. Тогда $\varkappa(x)$ есть случайная величина на вероятностном пространстве $\langle X, \mathscr{B}_X, P_X \rangle$.

В данной статье предлагаются методы статистической оптимизации, основанные на использовании свойств случайной величины $\varkappa(x)$ для исследования поведения минимизируемого функционала на множестве X и его подмножествах. В дальнейшем, чтобы отличать, идет речь о функционале или о случайной величине, будем использовать соответственно обозначения $\varkappa(x)$ и $\varkappa(x)$: $x \in Y$, указывая тем самым множество, на котором задана вероятностная мера.

Пусть $\tau = \{Y_j^i\}$, $i = \overline{1, n-1}$, $j = \overline{1, n_i}$ такая система множеств, что

$$Y_j^i \subset \bigcup_{l=1}^{n_{ij}} Y_{\pi_l^{ij}}^{i+1}, \quad Y_1^1 = X, \quad n_1 = 1, \quad \operatorname{card} Y_j^i > \operatorname{card} Y_k^{i+1}, \quad \forall j, k,$$

где $\Pi_{ij} = \{\pi_1^{ij}, \ldots, \pi_{n_{ij}}^{ij}\}$ — такие подмножества множества $N_i = \{1, 2, \ldots, \sum_{j=1}^{n_i} n_{ij}\}$, что

$$\bigcup_{j=1}^{n_i} \Pi_{ij} = N_i, \quad \Pi_{ij} \cap \Pi_{ik} = \emptyset \quad \forall j, k = \overline{1, n_i}.$$

Для определенности занумеруем, например, множества системы τ так, чтобы

$$Y_j^i \subset \bigcup_{l=\alpha_{j-1}^i+1}^{\alpha_j^i} Y_l^{i+1},$$

где

$$\alpha_0^i = 0, \quad \alpha_k^i = \sum_{j=1}^{k} n_{ij}, \quad k = \overline{1, n_i}.$$

Ясно, что для построения τ достаточно, например, осуществить разбиение исходного множества X на подмножества, затем каждое из подмножеств снова разбнть на подмножества и т. д. Однако заметим, что условие попарного непересечения множеств с одинаковым верхним индексом не является обязательным.

На подмножествах системы τ будем задавать вероятностную меру. Тем самым получим совокупность случайных величин $\varkappa(x)$: $x \in Y \in \tau$. Одной из важнейших характеристик случайной величины является функция ее распределения. Идеи использования функции распределения случайной величины для оптимизации детерминированного функционала рассматривались, например, в работах [1, 2]. Заметим, что в общем случае вид функции распределения $F(v)$ случайной величины $\varkappa(x)$: $x \in Y$ не известен. Однако можно утверждать следующее.

Во-первых, $F(\varkappa(x^*)) = 0$, а для любого $\varepsilon > 0$: $F(\varkappa(x^*) + \varepsilon) > 0$.

Во-вторых, если множество X конечно, а $N(\varepsilon)$ есть множество тех значений $x \in X$, для которых $\varkappa(x) < \varkappa(x^*) + \varepsilon$, то

$$F(\varkappa(x^*) + \varepsilon) = \frac{\operatorname{card} N(\varepsilon)}{\operatorname{card} X}.$$

В-третьих, функция $F(v)$ ступенчатая. Вместе с тем ее можно аппроксимировать непрерывной функцией, воспользовавшись, в частности, разложением $F(v)$ в ряд по нормальным распределениям. При разложении необходимо знать моменты случайной величины. Для этого можно предложить их выборочные оценки. Однако оценки моментов высоких порядков порождают большие погрешности. Поэтому такой подход целесообразен, когда $F(v)$ хорошо приближается 2—3 членами ряда в разложении.

Для некоторых классов функционалов дискретный характер множества X позволяет получать точные значения моментов. В частности, для задачи назначения в работе [3] получены точные среднее и дисперсия, как определенные функции от элементов матрицы стоимости (эти результаты можно перенести на моменты более высоких порядков). Кроме того, сам вид минимизируемого функционала $\varkappa(x)$ иногда позволяет говорить о функции распределения $\varkappa(x)$: $x \in Y$. Так, если $\varkappa(x) = \sum_{i=1}^{m} c_i x_i$, то при выполнении достаточно общих условий случайная величина $\varkappa(x)$: $x \in Y$ асимптотически ($m \to \infty$) нормальна. Функцию распределения можно также оценивать с помощью критериев согласия и т. д.

В дальнейшем будем предполагать, что вид функции распределения $F(v)$ известен с точностью до параметров θ и пользоваться обозначением $F(v, \theta)$.

Назовем испытанием генерацию точки x из множества $Y \in \tau$. Для оценки вектора параметров $\theta(Y)$ функции распределения $F(v, \theta(Y))$ случайной величины $\varkappa(x)$: $x \in Y$ произведем серию испытаний (т. е. сформируем выборку $\{\varkappa(x_1), \ldots, \varkappa(x_s)\}$, $x_i \in Y$, $i = \overline{1, s}$. Наименьшее выборочное значение при этом

можно рассматривать как приближение к оптимуму функционала $\varkappa(x)$. Зная функцию распределения и экстремальное выборочное значение, можно оценивать вероятность генерации точек x, которым соответствуют значения функционала меньшие, чем получены ранее. Тем самым можно осуществлять такой направленный перебор точек множества X, при котором вероятность уменьшения значений функционала растет.

Следующий алгоритм основан на указанных выше соображениях. Полагается, что для оценки параметров функции распределения и экстремального выборочного значения производится серия из s испытаний, а суммарное число испытаний ограничено и равно S.

Алгоритм 1

Шаг 1. Полагаем $i=j=1$ $S=S-s$, $l_q^p=\alpha_{q-1}^p$, $p=\overline{1, n-1}$, $q=\overline{1, n_p}$. Генерируем точки $x_1, \ldots, x_s$ из множества X. По выборке $\varkappa(x_1), \ldots, \varkappa(x_s)$ определяем $\varkappa(x^0)=\min\{\varkappa(x_1), \ldots, \varkappa(x_s)\}$ и оцениваем параметры $\theta(Y_1^1)$ функции распределения случайной величины $\varkappa(x)$: $x\in Y_1^1$. Переходим к шагу 2.

Шаг 2. Если $i\geqq n$, переходим к шагу 6; иначе — к шагу 3.

Шаг 3. Полагаем $l_j^i=l_j^i+1$, $k=l_j^i$. Если $k>\alpha_j^i$, переходим к шагу 6; иначе — к шагу 4.

Шаг 4. Генерируем точки $x_1, \ldots, x_s$ из множества Y_k^{i+1}. Вычисляем $\varkappa(x^0)=$ $=\min\{\varkappa(x^0), \varkappa(x_1), \ldots, \varkappa(x_s)\}$ и оцениваем параметры $\theta(Y_k^{i+1})$ функции распределения случайной величины $\varkappa(x)$: $x\in Y_k^{i+1}$.

Шаг 5. Определяем такие p и q, для которых значение $F(\varkappa(x^0), \theta(Y_q^p))$ наибольшее из всех рассмотренных множеств Y_k^i. Полагаем $i=p$, $j=q$, $S=S-s$ и переходим к шагу 2.

Шаг 6. Генерируем точки $x_1, \ldots, x_S$ из множества Y_j^i. Точку x^0 и величину $\varkappa(x^0)=\min\{\varkappa(x^0), \varkappa(x_1), \ldots, \varkappa(x_S)\}$ принимаем за приближение к оптимуму.

Сделаем некоторые замечания. Алгоритм допускает несложные преобразования в случае если число испытаний s (объем выборки) на каждом шаге различное. Если суммарное число испытаний S не фиксировано, то в качестве S можно взять достаточно большое число. В этом случае, если будет получено уменьшение значения функционала $\varkappa(x^0)$ (шаг 6), следует вернуться к шагу 5 и алгоритм продолжит работу.

Согласно алгоритма после каждой серии испытаний необходимо пересчитывать величины $F(\varkappa(x^0), \theta(Y_j^i))$ для всех рассмотренных ранее множеств. Но если значение $\varkappa(x^0)$ не изменилось, указанные величины также останутся прежними, т. е. вычислять $F(\varkappa(x^0), \theta(Y_j^i))$ требуется только для нового сформированного множества.

Обобщением предложенного алгоритма можно считать алгоритмы, в которых на каждом шаге вероятностная мера задается не на Y_j^i, а на всем X. При этом структура вероятностной меры такова, что вероятность генерации точек, „близких" к рекордной, увеличивается.

При реализации алгоритма 1 вид функции распределения случайных величин $\varkappa(x)$: $x \in Y_j^i$ задается априорно или оценивается. Кроме того необходимо получать выборочные оценки параметров функций распределения. Все это порождает определенные погрешности вероятностной модели.

Следующий алгоритм основан на непараметрическом подходе к оценке вероятности улучшений и построен с использованием схемы независимых испытаний (схема Бернулли).

Предположим, что к началу очередной серии испытаний рекордное значение функционала равно $\varkappa(x^0)$. Пусть A-случайное событие, состоящее в генерации точки $x \in Y_j^i$, такой что $\varkappa(x) < \varkappa(x^0)$. Положим, что вероятность события A равна p. Рассмотрим сложное испытание, состоящее в m-кратном повторении простого испытания (генерации точки из Y_j^i). Число λ появлений события A при m-кратном повторении независимых простых испытаний подчиняется биномиальному закону распределения вероятностей

$$P\{\lambda = m\} = \binom{m}{\lambda} p^m (1-p)^{\lambda - m}. \tag{2}$$

Итак, величина p — это вероятность улучшений. Если в серии из s испытаний получено l улучшений, то p состоятельно оценивается отношением l/s. В частности, если улучшение (событие A) произошло на k-м испытании, то $p = 1/k$. Зная оценку p, а также значения λ и $P\{\lambda = m\}$, из выражения (2) несложно оценить длину серии испытаний до получения λ улучшений (успехов).

Алгоритм 2

Шаг 1. Полагаем $i = j = l = 1$, $v_0 = v^* = 10^{16}$. Задаем (или оцениваем) длину серии испытаний s и величину ожидаемых успешных испытаний λ.

Шаг 2. Формируем множество Y_l^i и задаем вероятностное распределение на нем. Полагаем $k = 0$, $m = 0$. Переходим к шагу 3.

Шаг 3. Полагаем $k = k + 1$. Если $k > s$, переходим к шагу 8; иначе — к шагу 4.

Шаг 4. Генерируем точку $x \in Y_l^i$ и вычисляем $\varkappa(x)$. Если $\varkappa(x) < v_0$, полагаем $x^0 = x$, $v_0 = \varkappa(x^0)$. Переходим к шагу 5.

Шаг 5. Если $\varkappa(x) < v^*$, полагаем $m = m + 1$. Если $m < \lambda$, переходим к шагу 3; иначе- полагаем $v^* = v_0$ и переходим к шагу 6.

Шаг 6. Полагаем $j = l$, $l = \alpha_{j+1}^i + 1$ и переходим к шагу 7.

Шаг 7. Полагаем $i = i + 1$. Если $i < n$, переходим к шагу 2; иначе — к шагу 9.

Шаг 8. Если $l < \alpha_j^i$, полагаем $l = l + 1$ и переходим к шагу 2; иначе — переходим к шагу 6.

Шаг 9. Точку x^0 и величину $\varkappa(x^0)$ принимаем за приближение к оптимуму.

В рамках приведенного алгоритма можно варьировать числом ожидаемых успешных испытаний, уменьшая λ при уменьшении card Y_l^i. Тот факт, что на начальных этапах число λ целесообразно брать большим, объясняется необходимостью более тщательного „просмотра" множества Y_j^i для определения перспективного направления дальнейшего поиска.

Заметим, что как в первом, так и во втором алгоритмах важное значение имеет правило выбора нового множества Y_l^i. Естественно это множество выбирать так, чтобы рекордная точка x^0 ему принадлежала. Если такое множество уже рассмотрено, выбирается множество, которому принадлежит ближайшая (по значению функционала) точка и т. д. В приведенных алгоритмах указанный шаг опущен в целях простоты изложения, но такая процедура легко реализуется путем соответствующей перенумерации множеств системы τ.

В задачах оптимизации важное значение имеет поведение функционала $\varkappa(x)$ в окрестности глобального экстремума. В терминах вероятностной модели это соответствует поведению функции распределения $F(v)$ случайной величины $\varkappa(x)$: $x \in Y$ на „хвостах". Поскольку функционал $\varkappa(x)$ ограничен на Y снизу, то положив

$$\eta_Y = \inf_{x \in Y} \varkappa(x),$$

можно утверждать, что

$$F(\eta_Y) = 0, \quad F(\eta_Y + \delta) \neq 0 \quad \forall \delta > 0. \tag{3}$$

Зададимся числом $\varepsilon > 0$. Поставим функции распределения $F(v)$, удовлетворяющей условию (3), в соответствие функцию распределения $F^*(v)$ такую, что для некоторого числа η_Y^* имеет место:

$$F^*(\eta_Y^*) = 0, \quad F^*(\eta_Y^* + \delta) \neq 0 \quad \forall \delta > 0, \quad |\eta_Y - \eta_Y^*| < \varepsilon.$$

Тогда параметр η_Y^* является статистической оценкой оптимума функционала $\varkappa(x)$ на Y. В качестве $F^*(v)$ могут выступать предельное распределение случайной величины $\varkappa(x)$: $x \in Y$, а также предельное распределение ее экстремальных значений. В последнем случае точный вид исходного распределения $F(v)$ нас не интересует. Важно, чтобы оно удовлетворяло постулату устойчивости [4].

При статистической оценке оптимума можно использовать и непараметрический подход, в основу которого положены элементы теории порядковых статистик. Рассмотрим независимые реализации $\varkappa(x_1), \ldots, \varkappa(x_s)$ случайной величины $\varkappa(x)$: $x \in Y$. Обозначим через $\eta_{(1)} \leqq \ldots \leqq \eta_{(s)}$ порядковые статистики, соответствующие данной выборке. Выберем к крайних порядковых статистик и рассмотрим величину

$$\hat{\eta}(Y) = \sum_{i=1}^{k} a_i \eta_{(i)}.$$

Правило выбора k и значения коэффициентов a_i, $i = \overline{1, k}$ приведены в [2, 5]. Заметим, что величина $\hat{\eta}(Y)$ состоятельно оценивает η_Y^*. Возможность статистического оценивания оптимума функционала $\varkappa(x)$ на множествах Y_j^i позволяет предложить группу алгоритмов, построенных аналогично методу ветвей и границ.

Алгоритм 3.

Шаг 1. Полагаем $i=j=1$, $v_0=10^{16}$, $l_p=1$, $p=\overline{0,n}$. Задаемся допустимой погрешностью оценки $\tilde{\varepsilon}>0$.

Шаг 2. Генерируем точки $x_1, \ldots, x_s$ из множества Y_j^i. Полагаем $v_0=\min\{v_0, \varkappa(x_1), \ldots, \varkappa(x_s)\}$. Вычисляем статистическую оценку оптимума $\hat{\eta}(Y_j^i)$. Если $\hat{\eta}(Y_j^i)>\varkappa(x^0)-\tilde{\varepsilon}$, переходим к шагу 4; иначе — к шагу 3.

Шаг 3. Полагаем $j=\alpha_{l_i-1}^i$, $i=i+1$. Если $i>n$, переходим к шагу 5; иначе — к шагу 4.

Шаг 4. Полагаем $j=j+1$, $l_i=j$. Если $j\leqq\alpha_{l_{i-1}}^{i-1}$, переходим к шагу 2; иначе — к шагу 5.

Шаг 5. Полагаем $i=i-1$, $j=l_i$. Если $i\leqq 1$, переходим к шагу 7; иначе — к шагу 6.

Шаг 6. Если $\hat{\eta}(Y_j^i)>\varkappa(x^0)-\tilde{\varepsilon}$, переходим к шагу 5; иначе — к шагу 4.

Шаг 7. Точку x^0 и величину $\varkappa(x^0)$ принимаем за проближение к оптимуму.

Как видно из приведенного алгоритма, он состоит из прямого хода, когда вычисляются статистические оценки, и обратного хода, когда эти оценки используются для отсечений. Прямой ход определяется шагами 2—4, а обратный — шагами 5, 6. Погрешность $\tilde{\varepsilon}$ оценки может задаваться априорно, что будет соответствовать некоторому уровню вероятности, либо вычисляться с учетом функции распределения случайной величины $\hat{\eta}(Y_j^i)$.

Заметим, что необходимость многократной статистической оценки оптимума на подмножествах Y_j^i увеличивает вероятность утери множества Y_j^n, которому принадлежит глобальный экстремум. Однако можно использовать дополнительные средства в процессе поиска.

Имеется в виду следующее. Если $Y_2\subset Y_1$, то

$$\min_{x\in Y_1}\varkappa(x)\leqq\min_{x\in Y_2}\varkappa(x).$$

Таким образом вместо непосредственной оценки $\hat{\eta}(Y_1)$ и $\hat{\eta}(Y_2)$ можно проверить гипотезу H_0: $\hat{\eta}(Y_2)-\hat{\eta}(Y_1)\leqq\tilde{\varepsilon}$ при альтернативе H_1: $\hat{\eta}(Y_2)-\hat{\eta}(Y_1)>\varepsilon$. Другими словами, проверяется гипотеза, принадлежит ли искомое приближение к оптимуму подмножеству множества, содержащего это приближение.

Алгоритм 4.

Шаг 1. Полагаем $i=j=1$, $v_0=10^{16}$. Задаемся погрешностью $\tilde{\varepsilon}>0$.

Шаг 2. Генерируем точки $x_1, \ldots, x_s\in Y_1^1$. Вычисляем $\eta^0=\hat{\eta}(Y_1^1)$, $v_0=\min\{v_0, \varkappa(x_1), \ldots, \varkappa(x_s)\}$.

Шаг 3. Полагаем $j=\alpha_{l_{i-1}}^i$, $i=i+1$. Если $i>n$, переходим к шагу 8; иначе — к шагу 4.

Шаг 4. Генерируем точки $x_1, \ldots, x_s \in Y_j^i$. Вычисляем статистическую оценку оптимума $\hat{\eta}(Y_j^i)$ и $v_0 = \min\{v_0, \varkappa(x_1), \ldots, \varkappa(x_s)\}$.

Шаг 5. Проверяем гипотезу H_0: $\hat{\eta}(Y_j^i) - \eta^0 \leqq \tilde{\varepsilon}$ при альтернативе H_1: $\hat{\eta}(Y_j^i) - \eta^0 > \tilde{\varepsilon}$. Если гипотеза принимается, переходим к шагу 3; иначе — к шагу 6.

Шаг 6. Полагаем $j = j+1$, $l_i = j$. Если $j \leqq \alpha_{l_{i-1}}^{i-1}$, переходим к шагу 4; иначе — к шагу 7.

Шаг 7. Полагаем $i = i-1$, $j = l_i$. Если $i \geqq 1$, переходим к шагу 4; иначе — к шагу 8.

Шаг 8. Точку x^0 и величину $\varkappa(x^0)$ принимаем за начальное приближение к оптимуму. Если $i > n$, дальнейший поиск осуществляется на множестве $Y_{l_{n-1}}^n$.

Заметим, что при отсутствии погрешностей вычисления, если $Y \subset \bigcup_{i=1}^{k} Y_i$, то существует такое $j = \overline{1,k}$, что $\eta_Y = \eta_{Y_i}$. Однако вероятностный характер оценок $\hat{\eta}(Y_i)$ может привести к тому, что для всех $j = \overline{1,k}$: $\hat{\eta}(Y_i) - \hat{\eta}(Y) > \tilde{\varepsilon}$. Поэтому на шаге 7 алгоритма по-существу осуществляется возврат к уже рассмотренному ранее множеству и пересчет оценки $\hat{\eta}(Y_j^i)$.

Сравнение статистических оценок оптимумов может осуществляться не только относительно исходного множества X, а и относительно множества, полученного на предыдущем этапе. В этом случае проверяется гипотеза H_0: $\hat{\eta}(Y_j^i) - \hat{\eta}(Y_{l_{i-1}}^{i-1}) > \tilde{\varepsilon}$.

Во всех приведенных в статье алгоритмах общим является использование статистических свойств функционала $\varkappa(x)$ на подмножествах множества X для определения направления поиска оптимума. Эти алгоритмы могут быть так или иначе модифицированы применительно к конкретно решаемому классу задач. В частности, особенности применения алгоритмов в задачах размещения, квадратичного назначения, компоновки, балансировки и т. д. рассмотрены в [6—10]. Обширный вычислительный эксперимент описан в [2].

В заключение отметим, что класс предложенных алгоритмов объединен общим названием — метод последовательной статистической оптимизации [2]. Начало разработкам положено членом-корреспондентом АН УССР, профессором Ю. Г. Стояном [11]. Теоретические аспекты обоснования метода нашли отражение в [2, 12]. Более подробную библиографию также можно найти в [2].

Автор выражает благодарность Ю. Г. Стояну за постоянное внимание и помощь в работе.

ХАРЬКОВСКИЙ ИНСТИТУТ РАДИОЭЛЕКТРОНИКИ
КАФЕДРА ПРИКЛАДНОЙ МАТЕМАТИКИ
Г. ХАРЬКОВ, ПРОСП. ЛЕНИНА, 14.
СССР, 310141.

Литература

[1] Моцкус, И. Б., Многоэкстремальные задачи в проектировании (Статистические решения. Усиление локальных методов. Эвристические способности человека). — М.: Наука, 1967. — 215 с.

[2] Стоян, Ю. Г., Яковлев, С. В., Математические модели и оптимизационные методы геометрического проектирования. — Киев: Наукова думка, 1986. — 276 с.
[3] Ходзинский, А. Н., Статистические свойства критерия и их использование в алгоритмах решения задач комбинаторной оптимизации. — В кн.: Применения случайного поиска, Кемерово, 1984, с. 41—42.
[4] Гумбель Э., Статистика экстремальных значений. — М.: Мир, 1965. — 450 с.
[5] Математическая теория планирования эксперимента (Под ред. С. М. Ермакова). — М.: Наука, 1983. — 392 с.
[6] Стоян, Ю. Г., Гиль Н. И., Методы и алгоритмы размещения плоских геометрических объектов. — Киев: Наукова думка, 1976. — 247 с.
[7] Стоян, Ю. Г., Туранов Н. Т., Алгоритм размещения плоских фигур с наименьшей длиной связывающей сети. — Управляющие системы, 1970, вып. 4/5, с. 97—107.
[8] Стоян, Ю. Г., Соколовский, В. З., Решение некоторых многоэкстремальных задач методом сужающихся окрестностей. — Киев: Наукова думка, 1981. — 205 с.
[9] Кухаренок, М. А., Совместное решение задач размещения элементов и внешних контактных площадок при проектировании ЦВА. — В кн.: Прикладные методы кибернетики. — Киев: ИК АН УССР, 1984, с. 72—78.
[10] Стоян, Ю. Г., Соколовский, В. З., Яковлев, С. В., Метод уравновешивания вращающихся дискретно распределенных масс. — Энергомашиностроение, 1982, № 2, с. 4—5.
[11] Стоян, Ю. Г., Об одном способе поиска наилучшего решения для одного класса многоэкстремальных задач. — Управляемые системы, 1969, вып. 3.
[12] Стоян, Ю. Г., Яковлев, С. В., Исследование сходимости и эффективности метода сужающихся окрестностей. — Харьков, 1981. — 43 с. (Препринт АН УССР.) Ин-т пробл. машиностроения; № 168.)

(Received March 21, 1986)

A SZERKESZTŐ BIZOTTSÁG CÍME:

6720 SZEGED
SOMOGYI U. 7.

EDITORIAL OFFICE:

6720 SZEGED
SOMOGYI U. 7.
HUNGARY

Information for authors

Acta Cybernetica publishes only original papers in the field of computer sciences mainly in English, but also in French, German or Russian. Authors should submit two copies of manuscripts to the Editorial Board. The manuscript must be typed double-spaced on one side of the paper only. Footnotes should be avoided and the number of figures should be as small as possible. For the form of references, see one of the articles previously published in the journal. A list of special symbols used in the manuscript should be supplied by the authors.

A galley proof will be sent to the authors. The first-named author will receive 50 reprints free of charge.

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1987.január
Példányszám: 400. Terjedelem: 9,45 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 6601 és az MSZ 5602—55 szabvány szerint
87-320—Szegedi Nyomda — Felelős vezető: Surányi Tibor igazgató