Tomus 5. Fasciculus 3.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1981

Curat: Universitas Szegediensis de Attila József nominata

5. kötet 3. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK
KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1981. december
A Szegedi József Attila Tudományegyetem gondozásában

Subdirectly irreducible commutative automata

By Z. ÉSIK and B. IMREH

M. YOELI gave a characterization of finite subdirectly irreducible automata with a single input sign (cf. [9]). In [8] G. H. WENZEL generalized this result for the infinite case. In this paper we present another result along this line. Namely, we characterize all subdirectly irreducible commutative automata and hence all subdirectly irreducible commutative semigroups as well.

Notions and notations

An *automaton* is a system $\mathbf{A}=(A, X, \delta)$ where A is a nonempty set, the set of states, X is an arbitrary set, the set of input signs and, finally, $\delta\colon A\times X\to A$ is the transition function. As in general, we shall also use this transition function in the extended sense, i.e. as a mapping $\delta\colon A\times X^*\to A$. Here X^* denotes the free monoid generated by X. The identity of X^* is the empty word λ and $X^+=X^*\setminus\{\lambda\}$. We use the notation δ_p to denote the mapping induced by p: $\delta_p(a)=\delta(a, p)$ $(a\in A, p\in X^*)$. If a sign $x\in X$ induces a permutation of A then it is called a *permutation sign*. In this way we can divide X into two disjoint sets X_P and X_{NP}. X_P is the set of all permutation signs and $X_{NP}=X\setminus X_P$.

The mappings δ_p $(p\in X^*)$ form a monoid with respect to the composition of mappings. The identity of this monoid is the identity mapping on A, $\delta_\lambda=\mathrm{id}_A$. This monoid $S(\mathbf{A})$ is called the *characteristic semigroup* of $\mathbf{A}$. Sometimes another representation of the characteristic semigroup is useful in the literature. However, there is no essential difference among these definitions.

Each automaton $\mathbf{A}=(A, X, \delta)$ can be considered as a unoid, i.e. as a universal algebra equipped with unary operations only. Thus the notions such as subautomaton, homomorphism, congruence relation, quotient automaton, free automaton etc. can be introduced in a natural way. In connection with these notions we shall use the following notations: if $B\subseteq A$ then $[B]$ denotes the subautomaton generated by B, $C(\mathbf{A})$ denotes the lattice of all congruence relations of $\mathbf{A}$, if $\theta\in C(\mathbf{A})$ and $a\in A$ then $\theta(a)$ denotes the block containing a in the partition induced by θ, Δ_A is the equality relation of A, if $B\subseteq A$ then $\theta_{|B}=\theta\cap B\times B$, finally, if $\theta\in C(\mathbf{A})$ then the quotient automaton induced by θ is denoted by $\mathbf{A}/\theta=(A/\theta, X, \delta)$. Observe that we have used the same notation δ for the transition function of $\mathbf{A}/\theta$ as well. An automaton $\mathbf{A}$ is called *subdirectly irreducible* if either $\mathbf{A}$ has one state only, or $\Delta_A \neq \bigcap(\theta\colon \theta \in C(\mathbf{A}), \theta \neq \Delta_A)$.

Each subautomaton $\mathbf{B}=(B, X, \delta)$ of an automaton $\mathbf{A}=(A, X, \delta)$ can be viewed as a congruence relation $\sigma_{\mathbf{B}} \in C(\mathbf{A})$: $a\sigma_{\mathbf{B}}b$ if and only if $a, b \in B$ or $a=b$. And what is more, $C(\mathbf{B})$ can be embedded into $C(\mathbf{A})$ in a natural way, i.e. by the correspondence $\theta \to \theta'$ where $a\theta' b$ if and only if $a\theta b$ or $a=b$ for any $a, b \in A$. From this it follows that an automaton is subdirectly irreducible if and only if each of its subautomaton is subdirectly irreducible (cf. also [8]).

In the sequel we shall need a more general concept of subautomata, too. The automaton $\mathbf{B}=(B, Y, \delta')$ is an *X-subautomaton* of $\mathbf{A}=(A, X, \delta)$ if $B \subseteq A$, $Y \subseteq X$ and $\delta_{|B\times Y}=\delta'$. For the sake of simplicity we shall not make any distinction between δ and δ'. A special X-subautomaton of $\mathbf{A}$ is the X-subautomaton $\mathbf{B}=(A, X_P, \delta)$. It is called the *permutational subautomaton* of $\mathbf{A}$.

Various concepts of connectedness can be found in the literature. In what follows we shall use two of these concepts. An automaton $\mathbf{A}=(A, X, \delta)$ is called *strongly connected* if each state $a \in A$ is a generator of $\mathbf{A}$ and it is called *connected* if for arbitrary $a, b \in A$ $[a]\cap[b] \neq \emptyset$.

Our results pertain to commutative automata. An automaton $\mathbf{A}=(A, X, \delta)$ is said to be *commutative* if $\delta_{xy}=\delta_{yx}$ is satisfied for any $x, y \in X$, i.e. $xy=yx$ is an identity in $\mathbf{A}$. It is well-known that $\mathbf{A}$ is commutative if and only if δ_p is an endomorphism of $\mathbf{A}$ for every $p \in X^*$, and this is the reason why if $\mathbf{A}$ is generated by a state a then $\mathbf{A}$ is a free automaton with free generator a.

Thus a strongly connected commutative automaton is freely generated by any of its states. This implies that each input sign of a strongly connected commutative automaton $\mathbf{A}$ is a permutation sign, i.e. $S(\mathbf{A})$ is a commutative permutation group on A.

We have proved in [2] (cf. Theorem 1) that if a finite commutative automaton $\mathbf{A}$ has a generator state then $C(\mathbf{A}) \cong C(S(\mathbf{A}))$ and $|A|=|S(\mathbf{A})|$, where $C(S(\mathbf{A}))$ denotes the lattice of all congruences of $S(\mathbf{A})$. However, we have not used the finiteness of $\mathbf{A}$ in proving this statement thus this remains valid for arbitrary commutative automaton as well. Consequently, if $\mathbf{A}$ is a singly generated commutative automaton then $\mathbf{A}$ is subdirectly irreducible if and only if $S(\mathbf{A})$ is subdirectly irreducible. This was also discovered by I. PEÁK in [5].

Strongly connected commutative automata

The previously mentioned fact helped us to prove in [2] that a finite strongly connected commutative automaton is subdirectly irreducible if and only if it is a cyclic automaton of prime-power order. In this section we extend this result to the infinite case.

According to [3, 6] Abelian groups Z_{p^k} and Z_{p^∞} — where p is a prime — are called *cocyclic*. An automaton $\mathbf{A}=(A, X, \delta)$ is cocyclic, if its input-reduced subautomaton is (A, X)-isomorphic[1] to a strongly connected X-subautomaton of

[1] An automaton $\mathbf{A}=(A, X, \delta)$ is said to be (A, X)-isomorphic to an automaton $\mathbf{B}=(B, Y, \delta')$ if there exist bijections $\mu\colon A\to B$ and $\nu\colon X \to Y$ such that $\mu(\delta(a, x))=\delta'(\mu(a), \nu(x))$ for any $a \in A$ and $x \in X$.

an automaton obtained by viewing a cocyclic group as an automaton. (By the input-reduced subautomaton of an automaton $\mathbf{A}=(A, X, \delta)$ we mean an X-subautomaton $\mathbf{B}=(A, Y, \delta)$ where Y is a maximal subset of X with the property that $y_1 \neq y_2 (\in Y)$ implies $\delta_{y_1} \neq \delta_{y_2}$. $\mathbf{B}$ is unique up to isomorphism.) Observe that a strongly connected commutative automaton $\mathbf{A}$ is cocyclic if and only if $S(\mathbf{A})$ is a cocyclic group. It is known that an Abelian group is subdirectly irreducible if and only if it is a cocyclic group (cf. [3, 6]). Thus, by our previous remarks we obtain the following

Statement. A strongly connected commutative automaton is subdirectly irreducible if and only if it is a cocyclic automaton.

The general case

In this section we shall characterize all subdirectly irreducible commutative automata. First we need some definitions.

Let $\mathbf{A}=(A, X, \delta)$ be an arbitrary commutative automaton and define the binary relation $\leqq$ on A as follows: $a \leqq b$ if and only if there is a word $p \in X^*$ satisfying $\delta(a, p)=b$. It is not difficult to see that this relation is a preorder on A and it has the substitution property. Thus the relation $\leqq$ determines a congruence relation $\theta \in C(\mathbf{A})$: $a\theta b$ if and only if $a \leqq b$ and $b \leqq a$. Furthermore, the system $(A/\theta, \leqq)$ — where $\theta(a) \leqq \theta(b)$ if and only if $a \leqq b$ — becomes a partially ordered set. It is obvious that if $\mathbf{B}=(B, X, \delta)$ is a subautomaton of $\mathbf{A}$ then $B=\bigcup(\theta(b)\colon b \in B)$ and B/θ is an upper ideal in $(A/\theta, \leqq)$. Conversely, if B is an upper ideal in A/θ then $(\bigcup(\theta(b)\colon \theta(b) \in B), X, \delta)$ is a subautomaton of $\mathbf{A}$.

The automaton $\mathbf{A}$ is called *quasi-nilpotent* if the following three conditions are satisfied by $\mathbf{A}$:

i) $(A/\theta, \leqq)$ has a greatest element $\theta(a_0)$ and $\theta(a_0)=\{a_0\}$ where a_0 is called the absorbent state,

ii) $A/\theta \setminus \theta(a_0)$ has a greatest element which will always be denoted by $\theta(a_1)$,

iii) $\theta(a) < \delta(\theta(a), x)$ holds for any $a \in A \setminus \{a_0\}$ and $x \in X$ provided that $\delta_x \neq \mathrm{id}_{A/\theta}$ holds in the factor automaton $\mathbf{A}/\theta$.

Observe that for a quasi-nilpotent automaton $\mathbf{A}=(A, X, \delta)$ the condition $\delta_x = \mathrm{id}_{A/\theta}$ is equivalent to the condition that x is a permutation sign of $\mathbf{A}$. Furthermore, if $\mathbf{A}$ is quasi-nilpotent and finite then $(A/\theta, X_{NP}, \delta)$ is nilpotent.

Let $\mathbf{A}=(A, X, \delta)$ be again an arbitrary commutative automaton and let $P(A/\theta)$ denote the power set of A/θ. Define the mapping $f\colon P(A/\theta) \to P(A/\theta)$ by $f(C)=C \cup \max \bar{C}$ where $\max \bar{C}$ denotes the set of all maximal elements in the complement of C. It is easy to verify that f is a monoton mapping, i.e. $f(C) \subseteq f(C')$ provided $C \subseteq C'$. Thus, by Tarski's fixpoint theorem, (cf. [7]) f has a least fixpoint M'. M' is the smallest subset of A/θ such that $\max \overline{M'}=\emptyset$. Let $M(\mathbf{A})=\bigcup(\theta(a)\colon \theta(a) \in M')$.

On the other hand it is well-known that the least fixpoint of a monoton mapping on a complete lattice can be obtained as the least upper bound of a chain constructed from the least element of the lattice. Applying this construction to f we get $M'=\bigcup_{\alpha} M'_\alpha$ — or equivalently $M'=\bigcup_{\alpha<\beta} M'_\alpha$ — where for an arbitrary ordinal α the set M'_α is defined by transfinite induction as follows:

i) $M'_0=\max A/\theta$,

ii) $M'_\alpha=M'_{\alpha_1}\cup\max \overline{M'_{\alpha_1}}$ if $\alpha=\alpha_1+1$,

iii) $M'_\alpha=\bigcup_{\alpha_1<\alpha} M'_{\alpha_1}$ if $\alpha\neq 0$ is a limit ordinal.

It is obvious — by transfinite induction on α — that M'_α is an upper ideal in $(A/\theta, \leqq)$ and M'_α does not contain ω-chains. (By an ω-chain in a partially ordered set $(R, \leqq)$ we mean a subset $Q=\{q_0, q_1, \ldots\}\subseteq R$ such that $q_0<q_1<\ldots$. ω^{op}-chains are similarly defined just require $q_0>q_1>\ldots$ instead of the above condition.) As M'_α is always an upper ideal in $(A/\theta, \leqq)$ the system $M_\alpha(\mathbf{A})=(\bigcup(\theta(a)\colon \theta(a)\in M'_\alpha), X, \delta)$ is a subautomaton of $\mathbf{A}$. Observe that if $\mathbf{A}$ was a quasi-nilpotent automaton then $M_0(\mathbf{A})=\{a_0\}$ and $M_1(\mathbf{A})=\{a_0\}\cup\theta(a_1)$. If there is no danger of confusion we shall omit $\mathbf{A}$ in $M_\alpha(\mathbf{A})$ and $M(\mathbf{A})$.

A quasi-nilpotent automaton $\mathbf{A}=(A, X, \delta)$ will be called *separable* if for arbitrary states $a\neq b\in A$ such that $\{a, b\}\nsubseteq M_1$ there is a word $p\in X^+_{NP}$ satisfying both $\{\delta(a, p), \delta(b, p)\}\cap M\neq\emptyset$ and $\delta(a, p)\neq\delta(b, p)$.

We are now ready to state our main result.

Theorem. A commutative automaton $\mathbf{A}=(A, X, \delta)$ is subdirectly irreducible if and only if one of the following three conditions is satisfied by $\mathbf{A}$:

(a) $\mathbf{A}$ is a cocyclic automaton,

(b) $\mathbf{A}$ is a separable quasi-nilpotent automaton and the X-subautomaton $(A\setminus\{a_0\}, X_P, \delta)$, i.e. its permutational subautomaton without the absorbent state a_0, is the disjoint sum of pairwise isomorphic cocyclic automata,

(c) $\mathbf{A}$ is the disjoint sum of a cocyclic automaton and an automaton of one state.

Proof. In order to prove the necessity of our Theorem assume that $\mathbf{A}$ is subdirectly irreducible. First we shall consider the case when $\mathbf{A}$ is connected and show that $(A/\theta, \leqq)$ has a greatest element.

As $\mathbf{A}$ is connected there is at most one maximal element in A/θ. Therefore, it is enough to show that each element of A/θ has an upper bound which is maximal. Assume to the contrary that there is no maximal element in the upper ideal B' generated by an element $\theta(a)\in A/\theta$. Let $B=\bigcup(\theta(b)\colon \theta(b)\in B')$. (B, X, δ) is exactly the subautomaton generated by a, i.e. $B=[a]$. Let $b\in B$ be arbitrary. There is a state $b'\in B$ such that $\theta(b)<\theta(b')$, thus $\sigma_{[b]}\neq\Delta_A$. We shall show that $\bigcap(\sigma_{[b]}\colon b\in B)=\Delta_A$.

Suppose that $c\neq d$ and $c\sigma_{[b]}d$ holds for any $b\in B$. Of course we have $c, d\in B$. There is a state $\bar{b}\in B$ such that $\{c, d\}\nsubseteq[\bar{b}]$. Indeed, if $\theta(c)=\theta(d)$ then we may choose $\bar{b}$ such that $\theta(c)<\theta(\bar{b})$ if $\theta(c)<\theta(d)$ or $\theta(c)$ and $\theta(d)$ are incomparable then let $\bar{b}=d$. We supposed that $c\sigma_{[\bar{b}]}d$. But this is possible only if $c=d$, a contradiction. Therefore, $\bigcap(\sigma_{[b]}\colon b\in B)=\Delta_A$.

Let $\theta(a_0)$ denote the greatest element of A/θ. Since $\theta(a_0)$ is maximal in A/θ $(\theta(a_0), X, \delta)$ is a subautomaton of $\mathbf{A}$, furthermore, by the definition of θ, it is strongly connected. On the other hand we know that $(\theta(a_0), X, \delta)$ has to be a subdirectly irreducible automaton, thus, by the previous statement, it is a cocyclic automaton.

Suppose that $|\theta(a_0)|>1$. We show that in this case $\theta(a_0)=A$, i.e. **A** satisfies condition (a) of our theorem.

Assume that $a\in A$ and $a\notin\theta(a_0)$. Because of $\theta(a)<\theta(a_0)$ there is a word $p\in X^*$ such that $\delta(a,p)=a_0$. Let $\varrho\in C(\mathbf{A})$ be the congruence relation induced by the endomorphism δ_p. As $\delta_{p|\theta(a_0)}$ is a permutation of $\theta(a_0)$ we have $\varrho|_{\theta(a_0)}=$ $=\Delta_{\theta(a_0)}$ and $\varrho\neq\Delta_A$. Thus $\varrho\cap\sigma_{\theta(a_0)}=\Delta_A$. This, by $|\theta(a_0)|>1$ yields that **A** is subdirectly reducible, which is a contradiction.

Now consider the case $\theta(a_0)=\{a_0\}$ and $A\neq\{a_0\}$. By the same order of ideas as we have shown that A/θ has a greatest element one can easily prove that every element of $A/\theta\setminus\theta(a_0)$ has a maximal upper bound in $A/\theta\setminus\theta(a_0)$. But $A/\theta\setminus\theta(a_0)$ can not have two distinct maximal elements, consequently, there exists a greatest element $\theta(a_1)$ in $A/\theta\setminus\theta(a_0)$. Indeed, if both $\theta(a)$ and $\theta(b)$ are maximal in $A/\theta\setminus\theta(a_0)$ then $\sigma_{[a]}\cap\sigma_{[b]}=\Delta_A$ and $\sigma_{[a]},\sigma_{[b]}\neq\Delta_A$ are satisfied, contrary to the subdirect irreducibility of **A**.

Let $\theta(a_1)$ be the greatest element of $A/\theta\setminus\theta(a_0)$. Let us divide X into two disjoint sets X_1 and X_2: $X_1=\{x\colon x\in X, \delta(a_1,x)\in\theta(a_1)\}$, $X_2=\{x\colon x\in X, \delta(a_1,x)=a_0\}$. Since θ is a congruence relation we have $\delta(\theta(a_1),x)\subseteq\theta(a_1)$ if $x\in X_1$ and $\delta(\theta(a_1),x)=\theta(a_0)$ if $x\in X_2$. Hence $\mathbf{A}_1=(\theta(a_1),X_1,\delta)$ is a strongly connected X-subautomaton of **A**. We now show that $\mathbf{A}_1$ is a cocyclic automaton.

Assume that $\mathbf{A}_1$ is subdirectly reducible, i.e. there exist congruence relations $\{\varrho_i\in C(\mathbf{A}_1)\colon i\in I\}$ with $\bigcap(\varrho_i\colon i\in I)=\Delta_{\theta(a_1)}$ and $\varrho_i\neq\Delta_{\theta(a_1)}$ $(i\in I)$. Define the congruence relations $\Psi_i\in C(\mathbf{A})$ $(i\in I)$ by the equivalence $a\Psi_i b$ if and only if $a\varrho_i b$ or $a=b$ $(a,b\in A)$. It can be immediately seen that $\bigcap(\Psi_i\colon i\in I)=\Delta_A$ and $\Psi_i\neq\Delta_A$ $(i\in I)$ are satisfied. This contradicts the subdirect irreducibility of **A**. Therefore, $\mathbf{A}_1$ is subdirectly irreducible and thus, by our Statement, it is a cocyclic automaton.

Next we show that δ_x is a permutation of A and $\delta(\theta(a),x)\subseteq\theta(a)$ holds for any $x\in X_1$ and $a\in A$. Indeed, δ_x is injective since otherwise we would have $\sigma_{\theta(a_0)\cup\theta(a_1)}\cap\varrho=\Delta_A$ and $\sigma_{\theta(a_0)\cup\theta(a_1)},\varrho\neq\Delta_A$ where $\varrho\in C(\mathbf{A})$ is the congruence relation induced by the endomorphism δ_x. Now let $a\in A$ be arbitrary and let r^k be the order of δ_x in $S(\mathbf{A}_1)$. Define $\varrho\subseteq A\times A$ by $c\varrho d$ if and only if there is a non-negative integer n such that either $\delta(c,x^{nr^k})=d$ or $\delta(d,x^{nr^k})=c$. It is obvious that ϱ is reflexive and simmetric and has the substitution property, i.e. it is an invariant tolerance relation of **A**. By the injectivity of δ_x, it can be seen that it is transitive as well. Thus $\varrho\in C(\mathbf{A})$. It is not difficult to see that $\varrho\cap\sigma_{\theta(a_0)\cup\theta(a_1)}=\Delta_A$ while $\sigma_{\theta(a_0)\cup\theta(a_1)}\neq\Delta_A$. On the other hand $\varrho\neq\Delta_A$ holds if $a\notin\{\delta_{x^m}(a)\colon m\geqq 1\}$. Therefore, for every $x\in X_1$ and $a\in A$ there is an integer $n\geqq 1$ such that $a=\delta(a,x^n)$. Consequently, $\delta(\theta(a),x)\subseteq\theta(a)$ and $x^n=\lambda$ is an identity in $[a]$ implying that δ_x is a permutation of A.

As $X_1\subseteq X_P$ and $X_2\subseteq X_{NP}$ we get $X_1=X_P$ and $X_2=X_{NP}$. We have shown that if $x\in X_P$ then $\delta(\theta(a),x)\subseteq\theta(a)$ holds for each $a\in A$. Conversely, if $\delta(\theta(a),x)\subseteq\theta(a)$ holds for some $a\in A\setminus\theta(a_0)$ then also $\delta(\theta(a_1),x)\subseteq\theta(a_1)$ i.e. $x\in X_P$. This can be seen immediately as follows. As $\delta(\theta(a),x)=\theta(a)$ holds in $\mathbf{A}/\theta$ we obtain that $x=\lambda$ is an identity in $[\theta(a)]$. But $\theta(a_1)\in[\theta(a)]$, thus, $\delta(\theta((a_1),x)=\theta(a_1)$ in $\mathbf{A}/\theta$, i.e. $\delta(\theta(a_1),x)\subseteq\theta(a_1)$ in **A**.

So far we have proved that if **A** is subdirectly irreducible, connected, moreover, $\theta(a_0)=\{a_0\}$ and $A\neq\theta(a_0)$ then it is a quasi-nilpotent automaton. Next we show that in this case $(\theta(a),X_P,\delta)\cong(\theta(a_1),X_P,\delta)$ for any $a\in A\setminus\theta(a_0)$, hence the permutational subautomaton of **A** without the absorbent state is the disjoint sum of pairwise isomorphic cocyclic automata.

Indeed, if $a\in A\setminus\theta(a_0)$ then there exists a word $p\in X^*$ such that $\delta(a,p)=a_1$. By commutativity, the mapping $\delta_{p|\theta(a)}\colon \theta(a)\to\theta(a_1)$ is a homomorphism of $(\theta(a), X_P, \delta)$ into $(\theta(a_1), X_P, \delta)$. As $(\theta(a_1), X_p, \delta)$ is strongly connected $\delta_{p|\theta(a)}$ is an epimorphism. Now we shall show that $\delta_{p|\theta(a)}$ is an isomorphism. Assume that $b, c\in\theta(a)$ satisfy the condition $\delta(b,p)=\delta(c,p)=d$. Since $(\theta(a), X_P, \delta)$ is strongly connected there is a word $q\in X_P^*$ such that $\delta(b,q)=c$. By commutativity, $\delta(d,q)=d$, thus, $q=\lambda$ is an identity in $(\theta(a_1), X_P, \delta)$. In other words $\delta_{q|\theta(a_1)}= =\mathrm{id}_{\theta(a_1)}$. Let us define the relation $\varrho\in C(\mathbf{A})$ by $u\varrho v$ if and only if there is an integer $n\geqq 0$ such that either $\delta(u,q^n)=v$ or $\delta(v,q^n)=u$. Obviously, $\varrho\cap\sigma_{\theta(a_0)\cup\theta(a_1)}=\Delta_A$, and hence, by the subdirect irreducibility of **A**, from this it follows that $\varrho=\Delta_A$. Thus $b=c$ and $\delta_{p|\theta(a)}$ is an isomorphism.

It remained to prove that **A** is separable. Consider the set Z of all pairs (a,b) $(a\neq b\in A)$ such that $\{a,b\}\nsubseteq\theta(a_0)\cup\theta(a_1)$ and for every word $p\in X_{NP}^+$ if $\delta(a,p)\in M$ then $\delta(a,p)=\delta(b,p)$. We shall show that if $(a,b)\in Z$ and $x\in X_P$ then also $(\delta(a,x),\delta(b,x))\in Z$. Assume to the contrary $(\delta(a,x),\delta(b,x))\notin Z$. There are two cases. Either there is a word $p\in X_{NP}^+$ with $\delta(\delta(a,x),p)\in M$ and $\delta(\delta(b,x),p)\notin M$ or $\delta(a,xp),\delta(b,xp)\in M$ and $\delta(a,xp)\neq\delta(b,xp)$. In the first case, by commutativity and the facts $\delta_x(M)\subseteq M$ and $\delta_x(\overline{M})\subseteq\overline{M}$ it follows that $\delta(a,p)\in M$ and $\delta(b,p)\notin M$. This contradicts $(a,b)\in Z$. One can get a similar contradiction in the other case, too.

Suppose now that **A** is not separable, i.e. $Z\neq\emptyset$. Let $(a,b)\in Z$ and denote by $\varrho\in C(\mathbf{A})$ the congruence relation generated by the pair (a,b). By Malcev's lemma (cf. Theorem 10.3 in [4]), ϱ is the transitive closure of the relation Ψ given by $c\Psi d$ if and only if there is a word $p\in X^*$ with $\{c,d\}\subseteq\{\delta(a,p),\delta(b,p)\}$ or $c=d$. As $(a,b)\in Z$ and $(\delta(a,p),\delta(b,p))\in Z$ holds for every $p\in X_P^*$ it is not difficult to see that if $\theta(u)>\theta(a)$ and $u\Psi v$ are valid for some states $u,v\in M$ then $u=v$. Consequently, $\varrho_{|([a]\setminus\theta(a))\cap M}=\Delta_{([a]\setminus\theta(a))\cap M}$. If $a\in\theta(a_0)\cup\theta(a_1)$ then $\delta(a,p)=a_0$ holds for each $p\in X_{NP}^+$. Thus $\delta(b,p)=a_0$ is also valid for each $p\in X_{NP}^+$. But this is possible only if $b\in\theta(a_0)\cup\theta(a_1)$ contradicting $\{a,b\}\nsubseteq\theta(a_0)\cup\theta(a_1)$. Therefore $\theta(a)<\theta(a_1)$ and hence $\varrho_{|\theta(a_0)\cup\theta(a_1)}=\Delta_{\theta(a_0)\cup\theta(a_1)}$. Thus $\varrho\cap\sigma_{\theta(a_0)\cup\theta(a_1)}=\Delta_A$, a contradiction.

We have already proved that if **A** is a subdirectly irreducible connected commutative automaton then **A** satisfies condition (a) or (b) of our Theorem. Assume now that **A** is not connected. Then **A** is the disjoint sum of its connected subautomata $\mathbf{B}_i=(B_i,X,\delta)$ $(i\in I, |I|\geqq 2)$. We have $\cap(\sigma_{A\setminus B_i}\colon i\in I)=\Delta_A$ while if $|I|\geqq 3$ or $|I|=2$ and $|B_i|\geqq 2$ $(i\in I)$ then $\sigma_{A\setminus B_i}\neq\Delta_A$ $(i\in I)$. Therefore, $|I|=2$ — say $I=\{1,2\}$ — and $|B_2|=1$. As $\mathbf{B}_1$ has to be a subdirectly irreducible automaton and it is connected, one can show that $\mathbf{B}_1$ is a cocyclic automaton, i.e. **A** satisfies condition (c) of our Theorem. This ends the proof of necessity.

Conversely, by our Statement, it is obvious that if **A** contents condition (a) or (c) of the Theorem then **A** is subdirectly irreducible. Hence assume that condition (b) is satisfied by **A**.

We shall show that $\varrho_{|\theta(a_0)\cup\theta(a_1)}\neq\Delta_{\theta(a_0)\cup\theta(a_1)}$ holds for each congruence relation $\varrho\in C(\mathbf{A})$ generated by two distinct states $a,b\in A$.

This is quite obvious if $a,b\in\theta(a_0)\cup\theta(a_1)$. Hence suppose that $\{a,b\}\nsubseteq \nsubseteq\theta(a_0)\cup\theta(a_1)$ and set $Z=\{\varrho(c)\colon c\in M, |\varrho(c)|>1\}$. Since **A** is separable there is a word $p \in X_{NP}^+$ such that — say — $\delta(a, p) \in M$ and $\delta(a, p) \neq \delta(b, p)$. Thus $Z \neq \emptyset$. Since M/θ does not contain ω-chains there is a state $c_0 \in M$ such that $\varrho(c_0) \in Z$ and $\theta(c_0) \nless \theta(c)$ holds for any $\varrho(c) \in Z$.

Let us distinguish three cases and let $d_0 \in \varrho(c_0)$, $d_0 \neq c_0$. First assume that $c_0 = a_0$. If $d_0 \in \theta(a_1)$ we are ready. If $d_0 \notin \theta(a_1)$ then there is a word $p \in X_{NP}^+$ with $\delta(d_0, p) \in \theta(a_1)$. At the same time $\delta(a_0, p) = a_0$ thus we get $a_0 \varrho \delta(d_0, p)$, i.e. $\varrho_{|\theta(a_0) \cup \theta(a_1)} \neq \Delta_{\theta(a_0) \cup \theta(a_1)}$. Secondly assume that $c_0 \in \theta(a_1)$. If $d_0 \in \theta(a_1)$ then we are again ready. If $\theta(d_0) < \theta(a_1)$ then there is word $p \in X_{NP}^+$ such that $\delta(d_0, p) \in \theta(a_1)$. But $\delta(c_0, p) = a_0$ thus, $a_0 \varrho \delta(d_0, p)$. Finally, let $c_0 \notin \theta(a_0) \cup \theta(a_1)$. By separability, there is a word $p \in X_{NP}^+$ with $\delta(c_0, p) \neq \delta(d_0, p)$. But $\delta(c_0, p) \in M$ because (M, X, δ) is a subautomaton of **A** and $\theta(\delta(c_0, p)) > \theta(c_0)$ since **A** is quasi-nilpotent. Consequently, $(\delta(c_0, p), \delta(d_0, p)) \in Z$ contradicting the maximality of $\theta(c_0)$.

We have proved that every congruence relation $\varrho \in C(\mathbf{A})$ generated by two distinct elements of **A** satisfies $\varrho_{|\theta(a_0) \cup \theta(a_1)} \neq \Delta_{\theta(a_0) \cup \theta(a_1)}$. Therefore, **A** is subdirectly irreducible if and only if $(\theta(a_0) \cup \theta(a_1), X, \delta)$ is subdirectly irreducible. On the other hand $(\theta(a_0) \cup \theta(a_1), X, \delta)$ is subdirectly irreducible. This ends the proof of the Theorem.

Commutative automata with a finite set of input signs

In this section we shall point out that there is a somewhat simpler characterization of subdirect irreducibility in case of commutative automata with a finite set of input signs. Actually, we prove

Corollary 1. Let $\mathbf{A} = (A, X, \delta)$ be a commutative automaton with finite X. Then **A** is subdirectly irreducible if and only if one of the following three conditions are satisfied by **A**:

(a) **A** is a cyclic automaton of prime-power order,

(b) **A** is a quasi-nilpotent automaton and its permutational subautomaton without the absorbent state is the disjoint sum of pairwise isomorphic cyclic automaton of prime-power order, furthermore, for any $a \neq b \in A$ such that $\{a, b\} \nsubseteq \theta(a_0) \cup \theta(a_1)$ there is a sign $x \in X_{NP}$ with $\delta(a, x) \neq \delta(b, x)$,

(c) **A** is the disjoint sum of a cyclic automaton of prime-power order and an automaton of one state.

Proof. The proof follows by our Theorem and the fact that if **A** is quasi-nilpotent then we have $A = M(\mathbf{A})$. This latter equality can be seen by showing that if **A** is quasi-nilpotent then $\mathbf{A}/\theta$ can not contain an ω-chain.

Assume to the contrary **A** is quasi-nilpotent and $\theta(b_0) < \theta(b_1) < \dots$ is an ω-chain in $(A/\theta, \leqq)$. Let $X = \{x_1, \dots, x_r\}$. As $\theta(a_1)$ is the greatest element of $A/\theta \setminus \theta(a_0)$ there is a word $q_n = x_1^{\alpha_1^{(n)}} \dots x_r^{\alpha_r^{(n)}}$ with $\delta(b_n, q_n) = a_1$ for any $n \geqq 0$. Let $\alpha^{(n)}$ denote the vector consisting of the exponents occuring in q_n, i.e. $\alpha^{(n)} = (\alpha_1^{(n)}, \dots, \alpha_r^{(n)})$ $(n \geqq 0)$. By induction on t $(t = 0, \dots, r)$ we show that there is an infinite sequence of indices $I_t \subseteq \{0, 1, \dots\}$ such that $\alpha_s^{(i)} \leqq \alpha_s^{(j)}$ holds if $s \leqq t$ and $i < j \in I_t$. If $t = 0$ then let $I_t = \{0, 1, \dots\}$. Assume that we have already constructed the set I_{t-1} $(t \geqq 1)$ and consider $\Gamma = \{(\alpha_t^{(i)}, \dots, \alpha_r^{(i)}) \colon i \in I_{t-1}\}$. Supposing Γ is finite we obtain integers $i < j$ $(i, j \in I_{t-1})$ with $(\alpha_t^{(i)}, \dots, \alpha_r^{(i)}) = (\alpha_t^{(j)}, \dots, \alpha_r^{(j)})$. Let $w = x_1^{\alpha_1^{(j)} - \alpha_1^{(i)}} \dots x_{t-1}^{\alpha_{t-1}^{(j)} - \alpha_{t-1}^{(i)}}$. By commutativity, $\delta(b_i, q_i) = \delta(b_j, q_j) = \delta(b_j, wq_i) = a_1$.

On the other hand, by $\theta(b_i)<\theta(b_j)$, there is a word $p\in X_{NP}X^*$ with $\delta(\theta(b_i), p)=\theta(b_j)$. Or even, we may choose p in such a way that $\delta(b_i, p)=b_j$. Thus $a_0=\delta(\delta(b_i, q_i), p)=\delta(\delta(b_i, p)q_i)=\delta(b_j, q_i)\equiv\delta(b_j, q_i w)=\delta(b_j, wq_i)=a_1$, i.e. $a_0\equiv a_1$ yielding a contradiction. We have shown that Γ is infinite from which the existence of I_t follows.

Now let $I=I_r$ and $i<j$ $(i, j\in I)$. Applying the same sequence of ideas for the corresponding states b_i and b_j one can get a similar contradiction. This ends the proof of Corollary 1.

It is interesting to note that if $\mathbf{A}=(A, X, \delta)$ is a subdirectly irreducible commutative automaton and X is finite then $A=M_\omega(\mathbf{A})$. This can be seen as follows. We have proved that $A=M$ and one can prove in a similar way that there is no commutative automaton $\mathbf{B}$ with a finite set of input signs which is generated by one state such that $(B/\theta, \leqq)$ contains an ω^{op}-chain. Now, to see that $A=M_\omega(\mathbf{A})$ assume to the contrary $\max \overline{M'_\omega}\neq\emptyset$ and let $\theta(a)\in\max \overline{M'_\omega}$. Set $Z=\{\theta(b)\colon \theta(a)<\theta(b)\}$ and let Z_0 consist of all minimal elements of Z (with respect to the ordering $\leqq$). Of course $Z\subseteq M'_\omega$. For every $\theta(b)\in Z_0$ there exists a sign $x\in X$ with $\delta(a, x)\in\theta(b)$. Thus Z_0 is finite, $Z_0=\{\theta(b_1), \ldots, \theta(b_n)\}$. On the other hand Z can not contain ω^{op}-chains since otherwise $[a]/\theta$ would contain ω^{op}-chains. Thus, together with the fact that M'_ω is an upper ideal, $Z=\{\theta(b)\colon (\exists i)(i\in\{1, \ldots, n\}, \theta(b_i)<\theta(b))\}$. As $M'_\omega=\bigcup_{k<\omega} M'_k$, there corresponds an integer k_i to each $i\in\{1, \ldots, n\}$ such that $\theta(b_i)\in M'_{k_i}$. Let $k=\max_{i=1,\ldots,n} k_i$. Obviously, $Z_0\subseteq M'_k$ and, since M'_k is an upper ideal as well, $Z\subseteq M'_k$. But in this case if $\theta(b)$ is such that $\theta(a)<\theta(b)$ then $\theta(b)\in M'_k$, therefore, $\theta(a)$ is maximal in $\overline{M'_k}$, too. This results that $\theta(a)\in M'_{k+1}\subseteq M'_\omega$ contradicting our assumption $\theta(a)\in\overline{M'_\omega}$.

Also observe that if $\mathbf{A}=(A, X, \delta)$ is a subdirectly irreducible commutative automaton with finite X and if $\mathbf{A}$ is generated by one state then A is finite, too. Indeed, we know that $A=M_\omega$ holds, thus, $a_0\in M_\omega$ where a_0 denotes an arbitrary generator of $\mathbf{A}$. But $M_\omega=\bigcup_{n<\omega} M_n$, therefore, there is an integer n such that $a_0\in M_n$ and hence, $A=M_n$. On the other hand the finiteness of X implies the finiteness of M_n.

The following simple example shows that the equality $A=M(\mathbf{A})$ does not hold in general for arbitrary subdirectly irreducible commutative automata. Indeed, let $A=\{a_i, b_i\colon i\geqq 0\}$, $X=\{x\}\cup\{y_i\colon i\geqq 0\}$ and let $\delta\colon A\times X\to A$ be defined by:

(a) $$\delta(a_i,x)=\begin{cases}a_{i-1}, & \text{if } i>0\\ a_0, & \text{if } i=0,\end{cases}$$

(b) $$\delta(a_i, y_j)=a_0 \quad (i, j\geqq 0),$$

(c) $$\delta(b_i, x)=b_{i+1},$$

(d) $$\delta(b_i, y_j)=\begin{cases}a_{j-i}, & \text{if } j\geqq i\\ a_0, & \text{if } j<i.\end{cases}$$

It can be seen by an easy computation that $\mathbf{A}=(A, X, \delta)$ is a subdirectly irreducible commutative automaton with $M(\mathbf{A})=\{a_0, a_1, \ldots\}$.

Subdirectly irreducible commutative semigroups

Our Theorem makes possible for us to describe all subdirectly irreducible commutative semigroups.

First we note that if S is a commutative semigroup which has no identity then S is subdirectly irreducible if and only if S^1 is subdirectly irreducible where S^1 is S equipped with a new element 1, the identity of S^1. The sufficiency of this statement is obvious and does not require the commutativity of S. Conversely, assume that S^1 is subdirectly reducible, i.e. there are congruence relations $\varrho_i \neq \Delta_{S^1}$ $(i \in I)$ of S^1 such that $\bigcap(\varrho_i;\ i\in I)=\Delta_{S^1}$. We shall show that $\varrho_i|_S \neq \Delta_S$ is satisfied for each $i \in I$. Suppose that $\varrho_i|_S=\Delta_S$. There is exactly one element $s \in S$ with $s\varrho_i 1$. Let $s' \in S$ be arbitrary. As ϱ_i is a congruence relation of $S^1 ss' \varrho_i s'$. As S is closed under composition and $\varrho_{i|S}=\Delta_S$ from this we obtain $ss'=s'$. This means that s is a left identity, and by commutativity, an identity. This contradicts our assumption on S.

In the next corollary we use the notations in accordance with [1]. Observe that the congruence relations θ of the previous section corresponds to the Green's congruence relations $\mathcal{J}$ of commutative semigroups.

Corollary 2. A commutative semigroup S is subdirectly irreducible if and only if one of the following conditions is satisfied by S:

(i) S is a cocyclic group,

(ii) S is a commutative monoid with zero element and

(a) there is a least 0-minimal ideal R in S,

(b) J_1 is a cocyclic group, $(J_s, J_1|J_s) \cong J_1$ under the correspondence $\alpha \to \alpha|J_s$ $(\alpha \in J_1)$ if $s \neq 0$ furthermore, $J_s \neq J_s J_{s'}$, for arbitrary $s \in S \setminus \{0\}$ and $s' \in S \setminus J_1$,

(c) for any $\{s_1, s_2\} \subsetneqq R$ $(s_1 \neq s_2)$ there is an element $s \in S \setminus J_1$ with $\{s_1 s, s_2 s\} \cap \cap M \neq \emptyset$ and $s_1 s \neq s_2 s$ where M denotes the least ideal in S such that $\overline{M}/\mathcal{J}$ does not contain maximal elements with respect to the ordering $J_s \leqq J_{s'}$ if and only if $s|s'$ $(s, s' \in S)$,

(iii) S does not contain identity element and S^1 satisfies condition (ii) with $J_1=\{1\}$.

Every finitely generated subdirectly irreducible commutative semigroup is finite.

Proof. By our Theorem, Theorem 1 in [2], the representation theorem of semigroups and our previous remarks.

This Corollary implies Corollaries IV.7.4. and IV.7.5. in [6].

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

References

[1] CLIFFORD, A. H. and B. B. PRESTON, The algebraic theory of semigroups, Amer. Math. Soc. Surveys 7, Providence, Rhode Island, 1961.

[2] ÉSIK, Z. and B. IMREH, Remarks on finite commutative automata, *Acta Cybernet.*, v. 5. 1981, pp, 143—146.

[3] FUCHS, L., *Infinite abelian groups*, Academic Press, New York, London, 1970.
[4] GRATZER, G., *Universal algebra*, Van Nostrand, Princeton, N. J., 1968.
[5] PEAK, I., *Автоматы и полугруппы*, II., *Acta Sci. Math.* (Szeged), v. 26, 1965, pp. 49—54.
[6] PETRICH, M., *Lectures in semigroups*, Akademie Verlag, Berlin, 1977.
[7] TARSKI, A., A lattice-theoretic fixpoint theorem and its applications, *Pacific J. Math.*, v. 5, 1955, pp. 285—309.
[8] WENZEL, G. H., Subdirect irreducibility and equational compactness in unary algebras $(A; f)$, *Arch. Math.* (Basel), v. 21, 1970, pp. 256—263.
[9] YOELI, M., Subdirectly irreducible unary algebras, *Amer. Math. Monthly*, v. 74, 1967, pp. 957—960.

(Received June 19, 1980)

On attributed tree transducers

By Z. FÜLÖP

Introduction

The concept of attribute grammar was introduced by Knuth in [1] as a formal tool for defining the meaning of sentences generated by a context free grammar. Taking trees over some ranked alphabet instead of derivation trees of a context free grammar, and allowing the values of attributes to be only trees over another ranked alphabet, finally, restricting the semantic functions to tree-concatenation we obtain the notion of attributed tree translators.

In this paper we study some basic properties of attributed tree transformations. Namely, we point out that each completely defined top-down tree transformation can be induced by an attributed tree translator while the class of all completely defined bottom-up tree transformations and the class of all attributed tree transformations are incomparable. Finally, we prove some results concerning the composition of attributed tree transformations.

I. Notions and notations

Before turning to the discussion of attributed tree transducers we recall some fundamental notions and notations.

By a type, or ranked alphabet, we mean a finite nonempty set F of the form $F=F_0\cup F_1\cup\ldots\cup F_{\nu(F)}$, where the sets F_n ($n=0, \ldots, \nu(F)$) are pairwise disjoint. The elements of F_n are called n-ary operator symbols.

For arbitrary ranked alphabet F and set S the set of trees over S of type F is the smallest set $T_F(S)$ satisfying

(i) $F_0\cup S\subseteq T_F(S)$ and

(ii) if $f\in F_n$ ($n\geqq 0$); $p_1, \ldots, p_n\in T_F(S)$ then $f(p_1, \ldots, p_n)\in T_F(S)$.

We can define the height $(\mathrm{ht}(p))$, rank $(\mathrm{rn}(p))$, root $(\mathrm{root}(p))$ and the set of subtrees $(\mathrm{sub}(p))$ of a tree $p(\in T_F(S))$ as follows: if $p\in F_0\cup S$ then $\mathrm{ht}(p)=0$, $\mathrm{rn}(p)=1$, $\mathrm{root}(p)=p$ and $\mathrm{sub}(p)=\{p\}$ else, if p is of form $f(p_1, \ldots, p_n)$ for some $n(\geqq 1)$ and $f(\in F_n)$, then $\mathrm{ht}(p)=\max\{\mathrm{ht}(p_j)\,|\,1\leqq j\leqq n\}+1$, $\mathrm{rn}(p)=1+ \sum_{i=1}^{n}\mathrm{rn}(p_i)$, $\mathrm{root}(p)=f$ and $\mathrm{sub}(p)=\left(\bigcup_{i=1}^{n}\mathrm{sub}(p_i)\right)\cup\{p\}$.

Next we define the set path (p) of paths being in p as a subset of N^* (where N^* is the free monoid generated by the natural numbers, with identity λ) in the following way:

$$\text{path}(p) = \begin{cases} \{\lambda\} & \text{if} \quad p \in F_0 \cup S \\ \left(\bigcup_{j=1}^{n} \{jw \mid w \in \text{path}(p_j)\} \right) \cup \{\lambda\} & \text{if} \quad p = f(p_1, \ldots, p_n). \end{cases}$$

There is a corresponding label $\text{lb}_p(w)$ and a subtree $\text{str}_p(w)$ for each path w in a tree $p(\in T_F(S))$. They are defined as follows:

$$\text{lb}_p(w) = \begin{cases} \text{root}(p) & \text{if} \quad w = \lambda \\ \text{lb}_{p_j}(v) & \text{if} \quad w = jv, p = f(p_1, \ldots, p_n), \quad 1 \leqq j \leqq n, \end{cases}$$

and

$$\text{str}_p(w) = \begin{cases} p & \text{if} \quad w = \lambda \\ \text{str}_{p_j}(v) & \text{if} \quad w = jv, p = f(p_1, \ldots, p_n), \quad 1 \leqq j \leqq n. \end{cases}$$

In the rest of the paper the pairwise disjoint sets of variables $X = \{x_1, x_2, \ldots\}$, $Y = \{y_1, y_2, \ldots\}$, $U = \{u_1, u_2, \ldots\}$ and $Z = \{z_0, z_1, \ldots\}$ are kept fix. The variables, $z_0, z_1, \ldots$ are used as auxiliary variables. For an arbitrary integer $n(\geqq 0)$ the notations X_n, Y_n, U_n, Z_n are used to denote the sets $\{x_1, \ldots, x_n\}$, $\{y_1, \ldots, y_n\}$, $\{u_1, \ldots, u_n\}$, $\{z_1, \ldots, z_n\}$, respectively.

If at most the auxiliary variables $z_1, \ldots, z_l$ $(l \geqq 0)$ appear in a tree p, then p is also denoted by $p(z_1, \ldots, z_l)$. Substituting the elements $s_1, \ldots, s_l$ of a set S for the auxiliary variables $z_1, \ldots, z_l$ in a tree $p(z_1, \ldots, z_l)$, respectively, we obtain another tree which is denoted by $p(s_1, \ldots, s_l)$.

Sets of form $T(\subseteq T_F(X_n) \times T_G(Y_m))$ $(n, m \geqq 0)$ are called tree transformations and if $(p, q) \in T$ then q is called an image of p.

By a bottom-up tree transducer we mean a system $\mathbf{A} = (T_F(X_n), A, T_G(Y_m), A', P)$ where $n, m \geqq 0$ are integers, A is a nonempty finite set, $A' \subseteq A$, finally, P is a finite set of rules (or rewriting rules) having one of the following two forms:

(a) $f(a_1 z_1, \ldots, a_k z_k) \to a\bar{q}(z_{i_1}, \ldots, z_{i_l})$ where $k, l \geqq 0$; $f \in F_k$; $a, a_1, \ldots, a_k \in A$; $\bar{q} \in T_G(Y_m \cup Z_l)$; $1 \leqq i_1, \ldots, i_l \leqq k$, and

(b) $x_j \to a\bar{q}$ where $1 \leqq j \leqq n$, $a \in A$, $\bar{q} \in T_G(Y_m)$.

If there is a rule of form (a) in P for each $k(\geqq 0)$, $f(\in F_k)$; $a_1, \ldots, a_k$ $(\in A)$ as well as a rule of form (b) for each j $(1 \leqq j \leqq n)$, then $\mathbf{A}$ is said to be completely defined. Furthermore, if different rules have different left sides, then $\mathbf{A}$ is called deterministic. Let $p, q \in T_F(X_n \cup (A \times T_G(Y_m)))$. We say that q is directly derived from p — written $p \underset{\mathbf{A}}{\Rightarrow} q$ — if q appears from p in one of the following two ways:

(i) the tree $a\bar{q}(p_{i_1}, \ldots, p_{i_l})$ is substituted for a subtree $f(a_1 p_1, \ldots, a_k p_k)$ of p and the rule (a) is in P;

(ii) the tree $a\bar{q}$ is substituted for a subtree x_j of p and the rule (b) is in P.

Let us denote by $\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}$ the reflexive, transitive closure of the relation $\underset{\mathbf{A}}{\Rightarrow}$. Then the transformation $T(\mathbf{A})$ induced by $\mathbf{A}$ is:

$$T(\mathbf{A}) = \{(p, q) \mid p \in T_F(X_n),\ q \in T_G(Y_m) \text{ and } p \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} aq \text{ for some } a(\in A')\}.$$

Another type of tree-transducers is the top-down tree transducer. The system $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A', P))$ is called a top-down tree transducer if A is a finite nonempty set, $A' \subseteq A$ and finally P is a finite set of rules of the following two forms:

(c) $af(z_1, \ldots, z_k) \to \bar{q}(a_1 z_{i_1}, \ldots, a_l z_{i_l})$ where $k, l \geqq 0$; $a, a_1, \ldots, a_l \in A$; $f \in F_k$; $1 \leqq i_1, \ldots, i_l \leqq k$; $\bar{q} \in T_G(Y_m \cup Z_l)$;

(d) $ax_j \to \bar{q}$ where $a \in A$, $1 \leqq j \leqq n$, $\bar{q} \in T_G(Y_m)$.

Consider the trees $p, q(\in T_G(Y_m \cup (A \times T_F(X_n))))$. The relation $\underset{\mathbf{A}}{\Rightarrow}$ is now defined as follows: $p \underset{\mathbf{A}}{\Rightarrow} q$ if q appears from p

(i) by substituting the tree $\bar{q}(a_1 p_{i_1}, \ldots, a_l p_{i_l})$ for a subtree $af(p_1, \ldots, p_k)$ of p if the rule (c) is in P, or

(ii) by substituting the tree $\bar{q}$ for a subtree ax_j of p if the rule (d) is in P.

Again, $\overset{*}{\underset{\mathbf{A}}{\Rightarrow}}$ denotes the reflexive, transitive closure of $\underset{\mathbf{A}}{\Rightarrow}$ and the transformation $T(\mathbf{A})$ induced by $\mathbf{A}$ is given by

$$T(\mathbf{A}) = \{(p, q) | p \in T_F(X_n),\ q \in T_G(Y_m) \quad \text{and} \quad ap \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} q \quad \text{for some } a(\in A')\}.$$

If for all $a(\in A)$, $k(\geqq 0)$, $f(\in F_k)$ there is a rule of form (c) in P, moreover, for all $a(\in A)$, $j(=1, \ldots, n)$ there is a rule of form (d) in P then $\mathbf{A}$ is called completely defined. Finally, if different rules have different left sides and A' is a singleton set then $\mathbf{A}$ is called deterministic.

The cardinality of a set S is denoted by $|S|$ and we write s instead of the singleton $\{s\}$.

II. Attributed tree transducers

We now introduce the concept of attributed tree transducers.

Definition 2.1. The system $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A'_s, P, rt))$ where $n, m \geqq 0$ is called an attributed tree transducer — shortly, AT transduder — provided

(a) F and G are ranked alphabets;

(b) A is a finite set, the set of attributes which can be written in the form $A = A_s \cup A_i$ where A_s is the set of synthesized, A_i is the set of inherited attributes with $A_s \cap A_i = \emptyset$;

(c) $A'_s \subseteq A_s$;

(d) rt is a mapping of A_i into nonempty, finite subsets of $T_G(Y_m)$ (if $A_i = \emptyset$ then rt is not specified);

(e) the set of rules $P = (\bigcup_{f \in F} P_f) \cup \left(\bigcup_{j=1}^{n} P_{x_j}\right)$ is a finite subset of the set $(A \times \times (T_F(Z) \cup X_n)) \times T_G(Y_m \cup (A \times Z))$. For the sets P_f, for all $k(\geqq 0)$ and $f(\in F_k)$, it holds:

(i) for each $a(\in A_s)$ at least one rule of the form $af(z_1, \ldots, z_k) \leftarrow q(a_1 z_{i_1}, \ldots, a_l z_{i_l})$ $(l \geqq 0;\ 0 \leqq i_1, \ldots, i_l \leqq k;\ a_1, \ldots, a_l \in A;\ q \in T_G(Y_m \cup Z_l))$ is in P_f,

(ii) for each $a(\in A_i)$ and $1 \leqq j \leqq k$ at least one rule of the form $az_j \leftarrow q(a_1 z_{i_1}, \ldots, a_l z_{i_l})$ $(l \geqq 0;\ 0 \leqq i_1, \ldots, i_l \leqq k;\ a_1, \ldots, a_l \in A;\ q \in T_G(Y_m \cup Z_l))$ is in P_f,

(iii) P_f contains only rules of type (i) and (ii). For P_{x_j} (for each $j\ (=1, \dots, n)$) it holds that for any $a(\in A_s)$ at least one rule of form $ax_j \leftarrow q(a_1 z_0, \dots, a_l z_0)$ is in P_{x_j} and there is no other rule in P_{x_j}. (Observe that here, as well as in the rest of this paper, the elements (x, y) of P are written $x \leftarrow y$.)

If we write "one and only one" instead of "at least one" in (e) moreover require A_s' and $rt(a)$ (for each $a \in A_t$) to be a singleton then we obtain the concept of deterministic AT transducer.

Now let **A** be an AT transducer defined in 2.1 and take the trees $p(\in T_F(X_n \cup Z))$, $q, r(\in T_G(Y_m \cup (A \times \text{path}(p)) \cup (A \times Z)))$. We say that r is directly derived from q in p — and write $q \underset{p,\mathbf{A}}{\Leftarrow} r$ — if r appears from q by one of the following manners:

(a) substituting the tree $\bar{q}((a_1, v_1), \dots, (a_l, v_l))$ for some leaf $(a, w)(\in A \times \text{path}(p))$ of q if the following conditions hold:

(i) $a \in A_s$,

(ii) $\text{lb}_p(w) = f(\in F_k \text{ for some } k \geqq 0)$,

(iii) $af(z_1, \dots, z_k) \leftarrow \bar{q}(a_1 z_{i_1}, \dots, a_l z_{i_l}) \in P_f$,

(iv) $v_j = \begin{cases} w & \text{if } i_j = 0 \\ wi_j & \text{if } 1 \leqq i_j \leqq k \quad (j = 1, \dots, l); \end{cases}$

(b) substituting the tree $\bar{q}((a_1, w), \dots, (a_l, w))$ for some leaf (a, w) of q if

(i) $a \in A_s$,

(ii) $\text{lb}_p(w) = x_j\ (\in X_n)$,

(iii) $ax_j \leftarrow \bar{q}(a_1 z_0, \dots, a_l z_0) \in P_{x_j}$ hold;

(c) substituting the tree $\bar{q}((a_1, v_1), \dots, (a_l, v_l))$ for some leaf (a, w) of q if the following conditions hold:

(i) $a \in A_i$,

(ii) $w = vj$ ($v \in \mathbf{N}^*$, $j \in \mathbf{N}$ where $\mathbf{N}$ is the set of natural numbers),

(iii) $\text{lb}_p(v) = f\ \ (\in F_k \text{ for some } k \geqq 1)$,

(iv) $1 \leqq j \leqq k$,

(v) $az_j \leftarrow \bar{q}(a_1 z_{i_1}, \dots, a_l z_{i_l}) \in P_f$,

(vi) $v_t = \begin{cases} v & \text{if } i_t = 0 \\ vi_t & \text{if } 1 \leqq i_t \leqq k \quad (t = 1, \dots, l); \end{cases}$

(d) substituting a tree in $rt(a)$ for some leaf (a, w) of q if

(i) $w = \lambda$,

(ii) $a \in A_t$ hold;

(e) substituting the tree (a, z_j) for some leaf (a, w) if $\text{lb}_p(w) = z_j(\in Z)$ holds.

Let $\overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}}$ denote the reflexive and transitive closure of the relation $\underset{p,\mathbf{A}}{\Leftarrow}$. (Sometimes, if **A** is clear, instead of the notations $\underset{p,\mathbf{A}}{\Leftarrow}, \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}}$ we simply write $\underset{p}{\Leftarrow}, \overset{*}{\underset{p}{\Leftarrow}}$, respectively.)

Definition 2.2. Let **A** be the AT transducer defined in 2.1. By the transformation induced by **A** we mean the set

$$T(\mathbf{A}) = \{(p, q) \mid p \in T_F(X_n),\ q \in T_G(Y_m),\ (s_0, \lambda) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q \text{ for some } s_0(\in A_s')\}.$$

Observe that, in order to define the transformation induced by an AT transducer, it would have been enough to introduce the concept of derivation in a simpler way. Namely, it would have been enough to take p from $T_F(X_n)$ and the trees q, r from $T_G(Y_m(\cup A\times\text{path}(p)))$ — hence, (e) would have disappeared. The previously given more general notion of derivation will be needed only in Section IV.

Definition 2.3. Let **A** be an AT transducer. We say that **A** is circular if there exist $p(\in T_F(X_n))$, $q(\in T_G(Y_m\cup A\times\text{path}(p)))$ and $(a, w)(\in A\times\text{path}(p))$ such that $(a, w)\overset{+}{\underset{p,\mathbf{A}}{\Leftarrow}} q$ holds and (a, w) occurs in q as a leaf (where $\overset{+}{\underset{p,\mathbf{A}}{\Leftarrow}}$ is the transitive closure of $\underset{p,\mathbf{A}}{\Leftarrow}$). D. E. Knuth has pointed out in [1] that the circularity problem of attribute grammars is decidable. The algorithm presented by Knuth, with a small modification, is suitable to decide whether an AT transducer is circular or not. In the rest of this paper we shall always confine ourselves to noncircular AT transducers.

Therefore, it is clear that for an arbitrary AT transducer $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A_s', P, rt))$, and for each $p(\in T_F(X_n))$ and $(a, w)(\in A\times\text{path}(p))$ there exists a tree $q(\in T_G(Y_m))$ (if **A** is deterministic then only one) for which $(a, w)\overset{*}{\underset{p}{\Leftarrow}}q$ holds. Thus we may say that **A** is completely defined and this way of speaking is in accordance with the discussion of bottom-up and top-down tree transducers. Since **A** is completely defined it is clear that the domain of $T(\mathbf{A})$ is the set $T_F(X_n)$. Furthermore, if **A** is deterministic then $T(\mathbf{A})$ is a mapping of $T_F(X_n)$ into $T_G(Y_m)$.

Definition 2.4. The AT transducer (defined in 2.1) is called reduced if the following two conditions are satisfied by any leaf $(a, z)\in(A\times Z)$ appearing on the right side of a rule in P:

(i) if $z=z_0$ then $a\in A_i$,

(ii) if $z\in Z-\{z_0\}$ then $a\in A_s$.

Concerning attribute grammars the property being reduced means that no semantic rule may depend on a synthesyzed attribute of the left side or an inherited attribute of a nonterminal appearing in the right side of the corresponding context-free rewritting rule.

It is easy to show that for every AT transducer $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A_s', P, rt))$ there exists an AT transducer $\mathbf{A}'(=(T_F(X_n), A, T_G(Y_m), A_s', P', rt))$ which is reduced and equivalent to **A** in the sense that $T(\mathbf{A})=T(\mathbf{A}')$. P' can be obtained from P by a suitable substituting of rules in P in each other, and this process will terminate because **A** is noncircular.

Similary to the concept of dependency graph introduced by D. E. Knuth in [1], for every AT transducer **A**, each derivation $(a, w)\overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}}q$ can be represented by a directed graph. The nodes of this graph are the elements of $A\times\text{path}(p)$, moreover, if, in the derivation mentioned above some leaf (b, v) is substituted by some tree $\bar{q}((b_1, v_1), \ldots, (b_l, v_l))$, then there are directed arcs from nodes $(b_1, v_1), \ldots, (b_l, v_l)$ to the node (b, v). This representation of derivation makes the notions and proofs clearer. E.g. the notion of circularity means that the dependency directed graph corresponding to some derivation contains a directed circle.

We shall name the elements of $A\times\text{path}(p)$ attribute occurences in accordance with the above representation.

Further on we shall not always study the properties of AT transducers on the whole input set $T_F(X_n)$. The restriction of $T(\mathbf{A})$ to some $R(\subseteq T_F(X_n))$ will be denoted by $T(\mathbf{A})|_R$.

EXAMPLE 2.1. Let $n=m=3$ and $\mathbf{A}=(T_F(X_3), A, T_F(X_3), s_0, P, rt)$ where

(i) $F=F_1\cup F_2,\quad F_1=\{g\},\qquad F_2=\{f\};$

(ii) $A=A_s\cup A_i,\quad A_s=\{s_0, s_1\},\quad A_i=\{i\};$

(iii) $P=P_g\cup P_f\cup\left(\bigcup_{j=1}^{3} P_{x_j}\right),$

$$P_g=\{s_0 g(z_1)\leftarrow g(s_0 z_1),\ s_1 g(z_1)\leftarrow \text{arbitrary tree},\ iz_1\leftarrow s_1 z_1\},$$

$$P_f=\{s_0 f(z_1, z_2)\leftarrow s_0 z_2,\ s_1 f(z_1, z_2)\leftarrow s_1 z_1,\ iz_1\leftarrow \text{arbitrary tree},\ iz_2\leftarrow f(iz_0, s_1 z_2)\},$$

$$P_{x_j}=\{s_0 x_j\leftarrow iz_0,\ s_1 x_j\leftarrow x_j\},\quad j=1,2,3;$$

(iv) rt is an arbitrary mapping.

It is obvious that $\mathbf{A}$ is a deterministic and reduced AT transducer. Take the tree $p=g(f(x_2, f(x_1, x_3)))(\in T_F(X_3))$. The derivation $(s_0, \lambda)\underset{p}{\Leftarrow} g((s_0, 1))\underset{p}{\Leftarrow} g((s_0, 12))\underset{p}{\Leftarrow} g((s_0, 122))\underset{p}{\Leftarrow} g((i, 122))\underset{p}{\Leftarrow} g(f((i, 12), (s_1, 122)))\underset{p}{\Leftarrow} g(f(f((i, 1), (s_1, 12)), (s_1, 122)))\underset{p}{\Leftarrow} g(f(f((s_1, 1), (s_1, 12)), (s_1, 122)))\overset{*}{\underset{p}{\Leftarrow}} g(f(f(x_2, x_1), x_3))=q$ holds, consequently $(p, q)\in T(\mathbf{A})$ (see Figure 1).

We can see the directed graph corresponding to this derivation in Figure 2. The path components of the elements of $A\times\text{path}(p)$ are left for the sake of clarity.

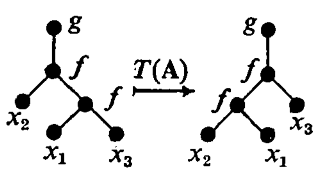


Fig. 1

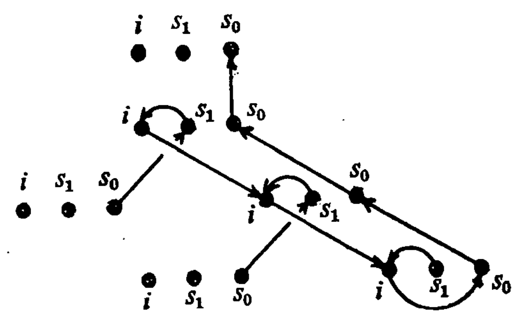


Fig. 2

Let us introduce the notation

$$R=\{g(f(x_{i_1}, f(x_{i_2}, \ldots, f(x_{i_{n-1}}, x_{i_n})\ldots)))|n\geqq 2,\ 1\leqq i_1, \ldots, i_n\leqq 3\}.$$

One can show that

$$T(\mathbf{A})|_R=\{(g(f(x_{i_1}, \ldots, f(x_{i_{n-1}}, x_{i_n})\ldots)), g(f(f\ldots f(x_{i_1}, x_{i_2}), \ldots, x_{i_{n-1}}), x_{i_n}))|n\geqq 2\},$$

hence $\mathbf{A}$ does not change the frontier of trees of R.

III. Comparing between AT transducers and classical tree transducers

Let us denote the class of all tree transformations induced by
(i) AT transducers,
(ii) AT transducers having only synthesized attributes,
(iii) deterministic AT transducers,
(iv) deterministic AT transducers having only synthesized attributes,
(v) completely defined top-down tree transducers,
(vi) completely defined deterministic top-down tree transducers,
(vii) completely defined bottom-up tree transducers,
(viii) completely defined deterministic bottom-up tree transducers

by
(i) $\mathcal{TA}$,
(ii) $\mathcal{TA}_s$,
(iii) $\mathcal{TDA}$,
(iv) $\mathcal{TDA}_s$,
(v) $\mathcal{TT}$,
(vi) $\mathcal{TDT}$,
(vii) $\mathcal{TB}$,
(viii) $\mathcal{TDB}$.

Before we shall go further let us introduce the concept of length of a derivation. Let $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A_s', P, rt))$ be an AT transducer and let $p(\in T_F(X_n))$, $(a, w)(\in A \times \text{path}(p))$ and $q(\in T_G(Y_m))$ satisfy the derivation $d=(a, w) \overset{*}{\underset{p}{\Leftarrow}} q$. The length $\text{lt}(d)$ of the derivation d is the least integer $n(\geqq 1)$ such that $(a, w) \overset{n}{\underset{p}{\Leftarrow}} q$, where $\overset{n}{\underset{p}{\Leftarrow}}$ denotes the n-th power of the relation $\underset{p}{\Leftarrow}$.

By induction on the length of derivation it is easy to prove:

Lemma 3.1. Let $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A_s', P))$ be a reduced AT transducer satisfying $A_i=\emptyset$. Then the following equivalence holds for each $p(\in T_F(X_n))$, $(a, w)(\in A \times \text{path}(p))$, $q(\in T_G(Y_m))$ and partition $w=uv$

$$(a, w) \overset{*}{\underset{p}{\Leftarrow}} q \quad \text{if and only if} \quad (a, v) \overset{*}{\underset{\text{str}_p(u)}{\Longleftarrow}} q. \quad \square$$

The next theorem has essentially appeared in [2] but we mention it for the sake of completness.

Theorem 3.1. $\mathcal{TT}=\mathcal{TA}_s$

Proof. First we are going to show that $\mathcal{TT} \subseteq \mathcal{TA}_s$. Indeed, let $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A', P))$ be a completely defined top-down tree transducer. Consider the AT transducer $\mathbf{B}(=(T_F(X_n), B, T_G(Y_m), B_s', P'))$ where
(i) $B=B_s=A$,
(ii) $B_s'=A'$,
(iii) $P'=P$.

It is easy to show by induction on $\text{ht}(p)$, and making use of Lemma 3.1, that for any $p(\in T_F(X_n))$, $a(\in A)$ and $q(\in T_G(Y_m))$

$$ap \overset{*}{\underset{\mathbf{A}}{\Rightarrow}} q \quad \text{if and only if} \quad (a, \lambda) \overset{*}{\underset{p,\mathbf{B}}{\Leftarrow}} q,$$

consequently, $T(\mathbf{A})=T(\mathbf{B})$. Conversely, take an arbitrary completely defined AT transducer $\mathbf{B}(=(T_F(X_n), B, T_G(Y_m), B'_s, P'))$ with $B_i=\emptyset$. We may assume without loss of the generality that $\mathbf{B}$ is reduced. Define $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A', P))$ by the equalities from (i) to (iii). Then, as earlier, we have $T(\mathbf{A})=T(\mathbf{B})$. This proves $\mathcal{T}\mathcal{A}_s\subseteq\mathcal{T}\mathcal{T}$. □

It is obvious that if $\mathbf{A}$ was deterministic in Theorem 3.1 then $\mathbf{B}$ would have been deterministic and conversely, hence we have

Corollary 3.1. $\mathcal{T}\mathcal{D}\mathcal{T}=\mathcal{T}\mathcal{D}\mathcal{A}_s$. □

However, it is easy to see that the tree transformation given in Example 2.1. can not be induced by a (deterministic) top-down tree transducer. Therefore, it is valid

Corollary 3.2. $\mathcal{T}\mathcal{T}\subset\mathcal{T}\mathcal{A}$ and $\mathcal{T}\mathcal{D}\mathcal{T}\subset\mathcal{T}\mathcal{D}\mathcal{A}$. □

Now we are going to see that these inclusions are not true in the bottom-up case.

Theorem 3.2. The class $\mathcal{T}\mathcal{D}\mathcal{B}$ and $\mathcal{T}\mathcal{D}\mathcal{A}$ are incomparable.

Proof. The tree transformation given in Example 2.1. is in $\mathcal{T}\mathcal{D}\mathcal{A}$ but it can not be induced by deterministic bottom-up tree transducers.

On the other hand the following deterministic bottom-up tree transformation will not be in $\mathcal{T}\mathcal{D}\mathcal{A}$.

Let $\mathbf{A}(=T_F(X_2), A, T_{F'}(X_2), A', P))$ be the bottom-up tree transducer where
(i) $F=F_1=\{f, g\}$, $F'=F'_1=\{f_1, f_2, g\}$;
(ii) $A=A'=\{a_1, a_2\}$;
(iii) P consists of the following rules:

$$x_1 \to a_1x_1, \qquad x_2 \to a_2x_2,$$
$$g(a_1z_1) \to a_1g(z_1), \quad g(a_2z_1) \to a_2g(z_1),$$
$$f(a_1z_1) \to a_1f_1(z_1), \quad f(a_2z_1) \to a_2f_2(z_1).$$

It is obvious that $\mathbf{A}$ is completely defined and deterministic. Consider $T(\mathbf{A})|_R$ where

$$R = \{f^n g^m(x_1)|n, m \geqq 0\}\cup\{f^n g^m(x_2)|n, m \geqq 0\}.$$

It is easy to see that

$$T(\mathbf{A})|_R = \{(f^n g^m(x_1), f_1^n g^m(x_1))|n, m \geqq 0\}\cup\{(f^n g^m(x_2), f_2^n g^m(x_2))|n, m \geqq 0\}.$$

Suppose that $T(\mathbf{A})$ is in $\mathcal{T}\mathcal{D}\mathcal{A}$ i.e. $T(\mathbf{A})$ can be induced by a deterministic AT transducer $\mathbf{B}(=(T_F(X_2), B, T_{F'}(X_2), b_0, P', rt))$ and suppose that $\mathbf{B}$ is reduced. Then $T(\mathbf{A})|_R=T(\mathbf{B})|_R$, necessarily. Let

$$K = |B_s|, \quad L = |B_i| \quad (\text{where } B = B_s\cup B_i),$$

$N=\max\{\mathrm{ht}(q)|q$ is the right side of some rule of $P'\}$, let $n>2NL(K+L)$ be fixed, and consider the trees $p_j^{(1)}=f^n g^j(x_1)$, $q_j^{(1)}=f_1^n g^j(x_1)$, $p_j^{(2)}=f^n g^j(x_2)$, $q_j^{(2)}= =f_2^n g^j(x_2)$ for all j $(=0, 1, \dots, L)$. (In the special case when the operator symbols appearing in some tree p are of arity 0 or 1, p is called unary. If p is unary then the elements of path (p) are of the form 1^l, further on simply written l.)

Now let us fix an arbitrary index $j(=0, 1, \dots, L)$ and denote $p_j^{(1)}, q_j^{(1)}, p_j^{(2)}, q_j^{(2)}$ by $p^{(1)}, q^{(1)}, p^{(2)}, q^{(2)}$, respectively. Then $(p^{(1)}, q^{(1)}) \in T(\mathbf{B})$, i.e. $(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}$. This derivation can be written as

$$(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q\big((s, n+j)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)} \tag{1}$$

for some $q\big(\in T_{F'}(Z_1)\big)$ and $s(\in B_s)$, since otherwise the derivation $(b_0, 0) \overset{*}{\underset{p^{(2)}}{\Leftarrow}} q^{(1)}$ would be true, and it is obviously a contradiction. (1) means that the derivation is to depend on some synthesized attribute of x_1. Furthermore, as $\mathbf{B}$ was reduced, we may suppose that for any tree $\bar{q}\big((b, w)\big)$ $\big(\bar{q} \in T_{F'}(Z_1), (b, w) \in B \times \text{path}\,(p^{(1)})\big)$ if

$$(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} \bar{q}\,\big((b, w)\big) \overset{+}{\underset{p^{(1)}}{\Leftarrow}} q\big((s, n+j)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}$$

then $w < n+j$ is true. On the other hand we must have $q = z_1$ i.e.

$$(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} (s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}. \tag{2}$$

If (2) would not be true then, by $(b_0, 0) \overset{*}{\underset{p^{(2)}}{\Leftarrow}} q\big((s, n+j)\big)$, we would have trees $\bar{q}^{(1)}, \bar{q}^{(2)} (\in T_{F'}(X_2))$ with $q(\bar{q}^{(1)}) = q^{(1)}$ and $q(\bar{q}^{(2)}) = q^{(2)}$, yielding a contradiction.

In Fig. 3, a heavy line views of the directed graph corresponding to the derivation $(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} (s, n+j)$. Take into account that in case of unary input and output trees the directed graph corresponding to any derivation is a directed "line".

Now we are going to study the derivation $(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}$. Since there are n operator symbols f_1 in $q^{(1)}$ and $n > 2NL(K+L)$, for some $c(\in B_i)$ and $r\big(\in T_{F'}(Z_1)\big)$ we have

$$(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r\big((c, n-L)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)} \tag{3}$$

and

(i) $r = z_1$ or r contains operator symbols f_1 only,

(ii) if for some tree $\bar{q}\big((b, w)\big)$ the derivation

$$(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} \bar{q}\big((b, w)\big) \overset{+}{\underset{p^{(1)}}{\Leftarrow}} r\big((c, n-L)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}$$

holds then $w > n-L$.

This follows easily by taking into account that each attribute occurence may appear at most once in a derivation and at most $2L(K+L)$ attribute occurences are in the top $n-L$-th level, moreover, that $\mathbf{B}$ is reduced.

As $\mathbf{B}$ is reduced there exist trees $r_l\big(\in T_{F'}(Z_1)\big)$ and attributes $i_l(\in B_i)$ $l(=0, 1, \dots, L)$ such that

$$\begin{aligned}(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r_0\big((i_0, n)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r_1\big((i_1, n-1)\big) \dots \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r_L\big((i_L, n-L)\big) = \\ = r\big((c, n-L)\big) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)},\end{aligned} \tag{4}$$

furthermore, if for some tree $\bar{q}\big((b, w)\big)$ $(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} \bar{q}\big((b, w)\big) \overset{+}{\underset{p^{(1)}}{\Leftarrow}} r_0\big((i_0, n)\big)$ then $w > n$.

Consider the attributes $i_0, i_1, \dots, i_L (\in B_i)$. Since $L = |B_i|$ there exist indices k, l $(0 \leqq) k < l (\leqq L)$ with $i_k = i_l$. Let $i = i_k (= i_l)$, then (4) can be written as

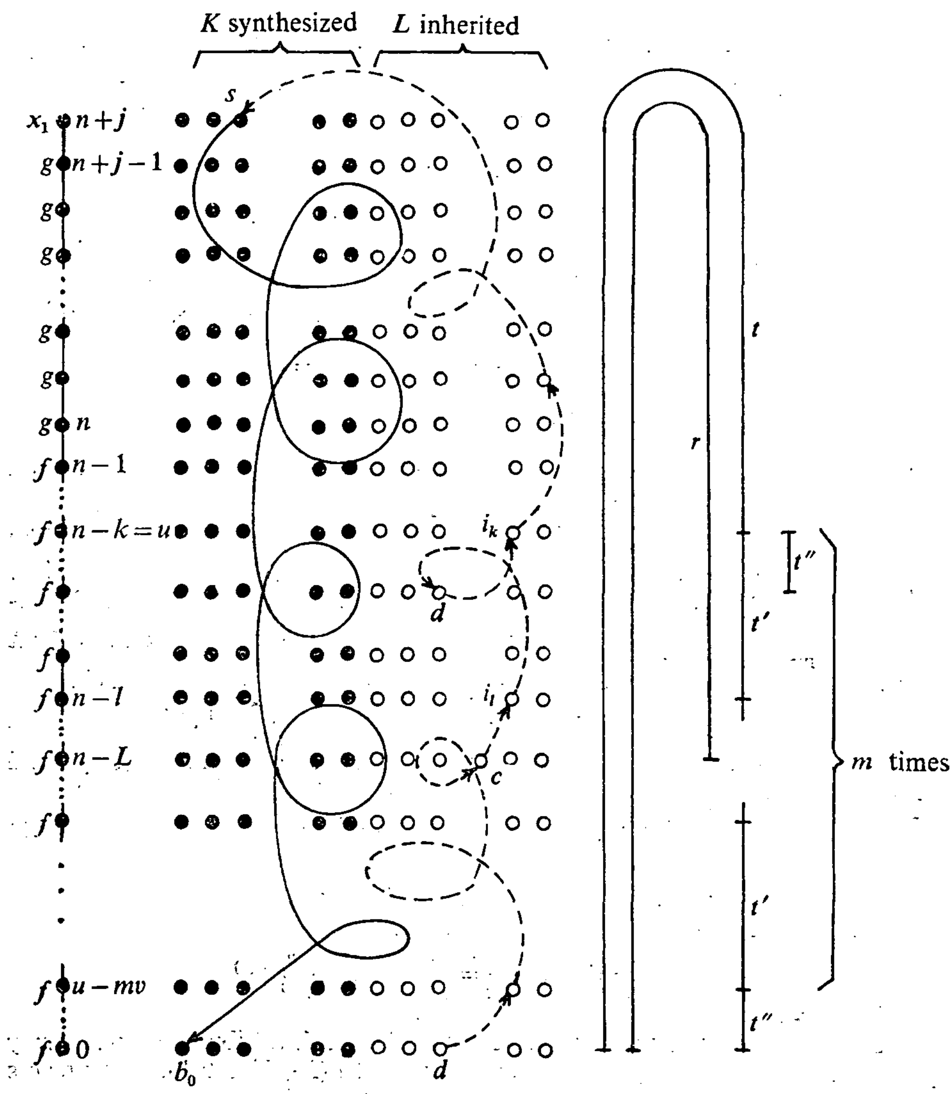


Fig. 3

$$(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r_k((i, n-k)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r_l((i, n-l)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r((c, n-L)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}. \tag{5}$$

Let us introduce the notations $u=n-k, v=l-k$ and $t=r_k$. Then there exists a tree $t'(\in T_{F'}(Z_1))$ such that $r_l=tt'$.[1] Thus (5) can be written as

$$(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} t((i, u)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} tt'((i, u-v)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} r((c, n-L)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)}. \tag{6}$$

[1] If t and t' are trees we often write tt' instead of $t(t')$.

Observe, that both t and t' may contain operator symbols f_1 only. Let m be the greatest integer number satisfying $u-mv\geqq 0$. It follows from (6) that

$$(s, n+j) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} t(t')^m((i, u-mv)) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} q^{(1)} \qquad (7)$$

[2]

(see Fig. 3). Finally, introduce the notation $y=u-mv$. Consider the tree t'' for which $(i, u) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} t''((d, u-y))$ holds for some $d(\in B_i)$ and which satisfies that if $(i, u) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} \bar{q}((b, w)) \overset{+}{\underset{p^{(1)}}{\Leftarrow}} t''((d, u-y))$ is valid for a tree $\bar{q}(\in T_{F'}(Z_1))$ and $(b, w)(\in B\times \text{path}(p^{(1)}))$ then $w>u-y$ holds. It follows from the definition of t'' that

$$(i, u-mv) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} t''((d, 0)) \qquad (8)$$

and $t''=z_1$ or t'' contains operator symbols f_1 only, as t' does.

We have from (2), (7) and (8) that

$$(b_0, 0) \overset{*}{\underset{p^{(1)}}{\Leftarrow}} t(t')^m t''((d, 0)) \underset{p^{(1)}}{\Leftarrow} q^{(1)}. \qquad (9)$$

Do not we forget that we have fixed j, therefore, t, t', t'', m and d depend on j.

But from (9), we can read that for each $j(=0, 1, \ldots, L)$ there exists a tree t_j as well as an inherited attribute d_j such that

$$(b_0, 0) \overset{*}{\underset{p_j^{(1)}}{\Leftarrow}} t_j((d_j, 0)) \underset{p_j^{(1)}}{\Leftarrow} q_j^{(1)}$$

moreover, t_j contains operator symbols f_1 only. Consider the inherited attributes $d_0, \ldots, d_L$. Then there exist indices k', l' such that $k'\neq l'$ and $d_{k'}=d_{l'}$. This implies that the trees $q_{k'}^{(1)}$ and $q_{l'}^{(1)}$ are of form $t_{k'}(s)$ and $t_{l'}(s)$, respectively, where $s=rt(d_{k'})=rt(d_{l'})$. But it is a contradiction which arises from $T(\mathbf{A})=T(\mathbf{B})$. Therefore $T(\mathbf{A})$ is not in $\mathcal{TDA}$. □

By a slight modification of the preceeding proof we get that $T(\mathbf{A})$ can not be induced by nondeterministic AT transducers. It is clear, besides, that the tree transformation given in Example 2.1 can not be induced by (nondeterministic) bottom-up tree transducers. Thus we obtain

Corollary 3.3. The classes $\mathcal{TB}$ and $\mathcal{TA}$ are incomparable. □

IV. Compositions of attributed tree transformations

First of all we are going to enter some notions. For any tree transformations $T_1(\subseteq T_F(X_n)\times T_G(Y_m))$, $T_2(\subseteq T_G(Y_m)\times T_H(U_r))$, the composition of T_1 and T_2 is the following transformation:

$$T_1\circ T_2 = \{(p, q)|(p, r)\in T_1 \text{ and } (r, q)\in T_2 \text{ for some } r\}.$$

[2] If t is a tree then $(t)^n$ means $\overbrace{tt\ldots t}^{n \text{ times}}$.

Let $\mathscr{C}_1$ and $\mathscr{C}_2$ be classes of tree transformations. The composition of $\mathscr{C}_1$ and $\mathscr{C}_2$ is the class

$$\mathscr{C}_1 \circ \mathscr{C}_2 = \{T_1 \circ T_2 | T_1 \in \mathscr{C}_1 \text{ and } T_2 \in \mathscr{C}_2\},$$

and for any class $\mathscr{C}$ and nonnegative integer n $\mathscr{C}^n$ is defined by induction

$$\mathscr{C}^1 = \mathscr{C},$$

$$\mathscr{C}^{n+1} = \mathscr{C}^n \circ \mathscr{C} \quad \text{if} \quad n \geqq 1.$$

We shall need the next Lemma.

Lemma 4.1. Let $n, m \geqq 0$ and let $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A_s', P, rt))$ be an AT transducer. Then there exists a constant N such that $\mathrm{rn}(q) \leqq N^{\mathrm{rn}(p)}$ holds for all $(p, q)(\in T(\mathbf{A}))$.

Proof. Let us enter the notations:

$$K = |A_s|,$$

$$L = |A_i| \quad \text{where} \quad A = A_s \cup A_i,$$

$$M = \max \{\mathrm{ht}(q) | q \text{ is the right side of some rule of } P\}.$$

Let $(p, q) \in T(\mathbf{A})$, i.e. assume that $d = (s_0, \lambda) \overset{*}{\underset{p, \mathbf{A}}{\Longleftarrow}} q$ for some $s_0 (\in A_s')$. Since $\mathbf{A}$ is noncircular $\mathrm{ht}(q) \leqq (K+L) M \mathrm{rn}(p)$ follows. It is obvious that there exists a constant R such that $\mathrm{rn}(q) \leqq R^{\mathrm{ht}(q)}$ for all $q(\in T_G(Y_m))$. It follows from the two latter inequality that the choice $N = R^{(K+L)M}$ will be right for our purposes. □

Theorem 4.1. $\mathscr{T}\mathscr{D}\mathscr{A}^n \subset \mathscr{T}\mathscr{D}\mathscr{A}_s \circ \mathscr{T}\mathscr{D}\mathscr{A}^n$

Proof. The inclusion $\mathscr{T}\mathscr{D}\mathscr{A}^n \subseteq \mathscr{T}\mathscr{D}\mathscr{A}_s \circ \mathscr{T}\mathscr{D}\mathscr{A}^n$ is obvious. In order to show that the inclusion is proper consider the transformation T in the class $\mathscr{T}\mathscr{D}\mathscr{A}_s \circ \mathscr{T}\mathscr{D}\mathscr{A}^n$ defined in the following way.

Let $\mathbf{A}(=(T_F(X_1), A, T_G(X_1), s, P))$ be a deterministic AT transducer with

(i) $F = F_1 = \{f\}, \quad G = G_2 = \{g\};$

(ii) $A = A_s = \{s\};$

(iii) $P = P_f \cup P_{x_1}$, where

$$P_f = \{s f(z_1) \leftarrow g(s z_1, s z_1)\},$$

$$P_{x_1} = \{s z_1 \leftarrow x_1\}.$$

If we denote by q_m the balanced tree over X_1 of type G, the height of which is m, then it is obvious that

$$T(\mathbf{A}) = \{(f^m(x_1), q_m) | m \geqq 0\}. \tag{10}$$

Moreover, let $\mathbf{B}(=(T_G(X_1), B, T_G(X_1), b, P', rt))$ be the deterministic AT transducer, where

(i) $G = G_2 = \{g\}$;

(ii) $B = B_s \cup B_i$ and $B_s = \{b\}$, $B_i = \{i\}$;

(iii) $P' = P'_g \cup P'_{x_1}$, where

$P'_g = \{bg(z_1, z_2) \leftarrow g(bz_1, bz_1),\ iz_1 \leftarrow bz_2,\ iz_2 \leftarrow iz_0\}$,

$P'_{x_1} = \{bx_1 \leftarrow g(iz_0, iz_0)\}$;

(iv) $rt(i) = x_1$.

Figure 4 shows the effect of **B** on balanced trees of type G. Let $R = \{q_m \mid m \geqq 0\}$. If we take into account, that for each $m(\geqq 0)$ the rank of q_m is $2^{m+1}-1$ then we can easily prove that

$$T(\mathbf{B})|_R =$$

$$= \{(q_m, q_{m'}) \mid m \geqq 0,\ m' = 2^{m+1}-1\}.$$

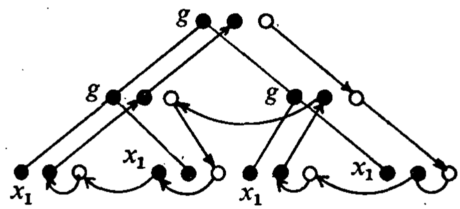


Fig. 4

Now let $T = T(\mathbf{A}) \circ \underbrace{T(\mathbf{B}) \circ \ldots \circ T(\mathbf{B})}_{n \text{ times}}$, hence $T \in \mathcal{TDA}_s \circ \mathcal{TDA}^n$. It follows from (10) and (11) that

$$T = \{(f^m(x_1), q_{m'}) \mid m \geqq 0,\ m' = \left.2^{2^{\cdot^{\cdot^{\cdot^{2^{m+1}}}}}}\right\}_{n \text{ times}} - 1\} \tag{11}$$

and this means that the rank of the image of the tree $f^m(x_1)$ at T is

$$\left.2^{2^{\cdot^{\cdot^{\cdot^{2^{m+1}}}}}}\right\}_{n+1 \text{ times}} - 1 \tag{12}$$

for each $m(\geqq 0)$.

We show that $T \notin \mathcal{TDA}^n$. Indeed, in the opposite case we would have a decomposition $T = T'_1 \circ \ldots \circ T'_n$ where $T'_j = T(\mathbf{A}'_j)$ for some deterministic AT transducers $\mathbf{A}'_j$ $(j = 1, \ldots, n)$. Thus, for each $j(=1, \ldots, n)$, Lemma 4.1 would give a constant N_j belonging to $\mathbf{A}'_j$ such that $\mathrm{rn}(q) \leqq N_j^{\mathrm{rn}(p)}$ if $(p, q) \in T'_j$ holds. From it would follow that the rank of the image of $f^m(x_1)$ at T would be at least

$$N_n^{N_{n-1}^{\cdot^{\cdot^{\cdot^{N_1^{m+1}}}}}}$$

for each $m(\geqq 0)$. This contradicts to (12). □

Taking into consideration the inclusion $\mathcal{TDA}_s \subset \mathcal{TDA}$ and the fact that Lemma 4.1 is true for nondeterministic AT transducers, we have

Corollary 4.1. $\mathcal{TDA}^n \subset \mathcal{TDA}^{n+1}$ and $\mathcal{TA}^n \subset \mathcal{TA}^{n+1}$. □

As we have seen, the proof of Theorem 4.1 depended on the fact that the AT transducers can "greatly" augment the rank of the trees. The question arises, whether the above inclusions will be true if we study a smaller class of AT transducers which can not do it. For this purpose we introduce the concept of linear AT transducer.

We say that an AT transducer $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), A'_s, P, rt))$ is linear (where $n, m \geqq 0$) if there exists a constant K such that from $(p, q) \in T(\mathbf{A})$ it follows that $\mathrm{rn}\,(q) \leqq K \mathrm{rn}\,(p)$. Let us denote the class of tree transformations induced by (deterministic) linear AT transducers by ($\mathscr{T}\mathscr{D}\mathscr{L}\mathscr{A}$) $\mathscr{T}\mathscr{L}\mathscr{A}$.

Theorem 4.2. The classes $\mathscr{T}\mathscr{D}\mathscr{A}$ and $\mathscr{T}\mathscr{D}\mathscr{L}\mathscr{A} \circ \mathscr{T}\mathscr{D}\mathscr{L}\mathscr{A}$ are incomparable.

Proof. It is obvious that there are tree transformations which are in $\mathscr{T}\mathscr{D}\mathscr{A}$ but not in $\mathscr{T}\mathscr{D}\mathscr{L}\mathscr{A} \circ \mathscr{T}\mathscr{D}\mathscr{L}\mathscr{A}$.

As we have seen the tree transformation $T(\mathbf{A})$ defined in Theorem 3.2 is not in $\mathscr{T}\mathscr{D}\mathscr{A}$. On the other hand $T(\mathbf{A})$ can be decomposed in the following way. Let $\mathbf{B}(=(T_F(X_2), B, T_F(X_2), P', b, rt'))$ be the AT transducer where

(i) $F = F_1 = \{f, g\}$,

(ii) $B = B_s \cup B_i, \quad B_s = \{b\}, \quad B_i = \{b_1, b_2\}$,

(iii) $P' = P'_f \cup P'_g \cup \left(\bigcup_{j=1}^{2} P'_{x_j} \right)$ with

$$P'_f = \{bf(z_1) \leftarrow bz_1,\ b_1 z_1 \leftarrow f(b_1 z_0),\ b_2 z_1 \leftarrow f(b_2 z_0)\},$$

$$P'_g = \{bg(z_1) \leftarrow bz_1,\ b_1 z_1 \leftarrow g(b_1 z_0),\ b_2 z_1 \leftarrow g(b_2 z_0)\},$$

$$P'_{x_j} = \{bx_j \leftarrow b_j z_0\} \quad (j = 1, 2),$$

(iv) $rt'(b_j) = x_j, \quad (j = 1, 2)$;

and $\mathbf{C}(=(T_F(X_2), C, T_{F'}(X_2), P'', c, rt''))$ be the AT transducer where

(i) $F' = F'_1 = \{f_1, f_2, g\}$,

(ii) $C = C_s \cup C_i, \quad C_s = \{c\}, \quad C_i = \{c_1, c_2\}$,

(iii) $P'' = P''_f \cup P''_g \cup \left(\bigcup_{j=1}^{2} P''_{x_j} \right)$ with

$$P''_f = \{cf(z_1) \leftarrow cz_1,\ c_1 z_1 \leftarrow f_1(c_1 z_0),\ c_2 z_1 \leftarrow f_2(c_2 z_0)\},$$

$$P''_g = \{cg(z_1) \leftarrow cz_1,\ c_1 z_1 \leftarrow g(c_1 z_0),\ c_2 z_1 \leftarrow g(c_2 z_0)\},$$

$$P''_{x_j} = \{cx_j \leftarrow c_j z_0\} \quad (j = 1, 2),$$

(iv) $rt''(c_j) = x_j \quad (j = 1, 2)$.

It is easy to see that both $\mathbf{B}$ and $\mathbf{C}$ is deterministic and linear, moreover, that $T(\mathbf{A}) = T(\mathbf{B}) \circ T(\mathbf{C})$ holds. This ends the proof. □

In case of $n=1$ Theorem 4.1 says that $\mathscr{T}\mathscr{D}\mathscr{A} \subset \mathscr{T}\mathscr{D}\mathscr{A}_s \circ \mathscr{T}\mathscr{D}\mathscr{A}$. If we exchange the factors of the composition then we have

Theorem 4.3. $\mathcal{TDA}\circ\mathcal{TDA}_s=\mathcal{TDA}$.

Proof. First we prove the inclusion $\mathcal{TDA}\circ\mathcal{TDA}_s\subseteq\mathcal{TDA}$.

For this purpose let $\mathbf{A}(=(T_F(X_n), A, T_G(Y_m), a_0, P, rt))$ and $\mathbf{B}(=(T_G(Y_m), B, T_H(U_r), b_0, P'))$ be deterministic AT transducers and suppose that $\mathbf{B}$ has only synthesized attributes.

Consider the AT transducer $\mathbf{C}(=(T_F(X_n), C, T_H(U_r), c_0, P'', rt''))$ defined as follows:

(a) $C=C_s\cup C_i$ where $C_s=B\times A_s$, $C_i=B\times A_i$ $(A=A_s\cup A_i)$;

(b) $c_0=(b_0, a_0)$;

(c) P'' is built in the way:

(i) for each $a(\in A_s)$, $b(\in B)$, $k(\geqq 0)$ and $f(\in F_k)$, if $af\leftarrow\bar q(a_1z_{i_1},\dots,a_lz_{i_l})\in P_f$ $(l\geqq 0;\ 0\leqq i_1,\dots,i_l\leqq k;\ \bar q\in T_G(Y_m\cup Z_l))$ and $(b,\lambda)\overset{*}{\underset{\bar q,\mathbf{B}}{\Longleftarrow}}q((b_1,z_{j_1}),\dots,(b_t,z_{j_t}))$ $(t\geqq 0;\ 1\leqq j_1,\dots,j_t\leqq l;\ q\in T_H(U_r\cup Z_t))$ then take into P''_f the rule $(b,a)f\leftarrow q((b_1,a_{j_1})z_{i_{j_1}},\dots,(b_t,a_{j_t})z_{i_{j_t}})$;

(ii) for each $a(\in A_s)$, $b(\in B)$, $x_j(\in X_n)$ if $ax_j\leftarrow\bar q(a_1z_0,\dots,a_lz_0)\in P_{x_j}$ $(l\geqq 0$, $\bar q\in T_G(Y_m\cup Z_l))$ and $(b,\lambda)\overset{*}{\underset{\bar q,\mathbf{A}}{\Longleftarrow}}q((b_1,z_{j_1}),\dots,(b_t,z_{j_t}))$ $(t\geqq 0;\ 1\leqq j_1,\dots,j_t\leqq l;\ q\in T_H(U_r\cup Z_t))$ then take into P''_{x_j} the rule $(b,a)x_j\to q((b_1,a_{j_1})z_0,\dots,(b_t,a_{j_t})z_0)$;

(iii) for each $i(\in A_i)$, $b(\in B)$, $k(\geqq 1)$, $f(\in F_k)$ and $1\leqq j\leqq k$ if $iz_j\leftarrow\bar q(a_1z_{i_1},\dots,a_lz_{i_l})\in P_f$ $(l\geqq 0;\ 0\leqq i_1,\dots,i_l\leqq k;\ \bar q\in T_G(Y_m\cup Z_l))$ and $(b,\lambda)\overset{*}{\underset{\bar q,\mathbf{B}}{\Longleftarrow}}q((b_1,z_{j_1}),\dots,(b_t,z_{j_t}))$ $(t\geqq 0;\ 1\leqq j_1,\dots,j_t\leqq l;\ q\in T_H(U_r\cup Z_t))$ then take into P''_f the rule $(b,i)z_j\leftarrow q((b_1,a_{j_1})z_{i_{j_1}},\dots,(b_t,a_{j_t})z_{i_{j_t}})$;

(d) for each $(b,i)(\in C_i)$ let $rt''((b,i))=q(\in T_H(U_r))$ if $rt(i)=\bar q(\in T_G(Y_m))$ and $(b,\lambda)\overset{*}{\underset{\bar q,\mathbf{B}}{\Longleftarrow}}q$ hold.

We can prove the following: for each $p(\in T_F(X_n))$, $q(\in T_H(U_r))$, $a(\in A)$, $b(\in B)$ and $w(\in\text{path}(p))$: $(\exists q'(\in T_G(Y_m)))\ ((a,w)\overset{*}{\underset{p,\mathbf{A}}{\Longleftarrow}}q'$ and $(b,\lambda)\overset{*}{\underset{q',\mathbf{B}}{\Longleftarrow}}q)$ if and only if $((b,a),w)\overset{*}{\underset{p,\mathbf{C}}{\Longleftarrow}}q$.

The proof of the only if part is performed by an induction on the length of the derivation $(a,w)\overset{*}{\underset{p,\mathbf{A}}{\Longleftarrow}}q'$.

If $\mathrm{lt}((a,w)\overset{*}{\underset{p,\mathbf{A}}{\Longleftarrow}}q')=1$ then one of the following cases is valid:

$$a\in A_s,\quad \mathrm{lb}_p(w)=f,\quad af\leftarrow q'\in P_f; \tag{13}$$

$$a\in A_s,\quad \mathrm{lb}_p(w)=x_j,\quad ax_j\leftarrow q'\in P_{x_j}; \tag{14}$$

$$a\in A_i,\quad w=vj\quad (v\in\mathbf{N}^*, j\in\mathbf{N}),\quad \mathrm{lb}_p(v)=f(\in F_k,\ k\geqq 1),\quad 1\leqq j\leqq k \text{ and } az_j\leftarrow q'\in P_f; \tag{15}$$

$$a\in A_i,\quad w=\lambda,\quad q'=rt(a). \tag{16}$$

Thus, what we wanted to prove it holds by definition in all of the four cases.

Now let $\mathrm{lt}\left((a, w) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q'\right) > 1$. Then the derivation $(a, w) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q'$ can be written as

$$(a, w) \underset{p,\mathbf{A}}{\Leftarrow} q_0'\left((a_1, w_1), \ldots, (a_l, w_l)\right) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_0'(q_1', \ldots, q_l') = q' \tag{17}$$
$$(l \geqq 1;\ a_1, \ldots, a_l \in A;\ q_0' \in T_G(Y_m \cup Z_l)).$$

Let $q_0(\in T_H(Z_t))$ be the tree for which

$$(b, \lambda) \overset{*}{\underset{q_0'\mathbf{B}}{\Leftarrow}} q_0\left((b_1, z_{j_1}), \ldots, (b_t, z_{j_t})\right) \quad (t \geqq 0;\ 1 \leqq j_1, \ldots, j_t \leqq l) \tag{18}$$

is valid. Then the derivation $(b, \lambda) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q$ can be specified in the following form:

$$(b, \lambda) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q_0\left((b_1, v_1), \ldots, (b_t, v_t)\right) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q_0(q_1, \ldots, q_t) = q \tag{19}$$

where $\mathrm{str}_{q'}(v_s) = q_{j_s}'$ for all $s(=1, \ldots, t)$. Taking into account this latter fact as well as the derivation $(b_s, v_s) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q_s$ $(s=1, \ldots, t)$ by Lemma 3.1 we get

$$(b_s, \lambda) \overset{*}{\underset{q_{j_s}',\mathbf{B}}{\Leftarrow}} q_s \quad (s = 1, \ldots, t). \tag{20}$$

Studying (17) we can say that three cases are possible, namely

$$a \in A_s,\quad \mathrm{lb}_p(w) = f (\in F_k \text{ for some } k \geqq 0); \tag{21}$$

$$a \in A_s,\quad \mathrm{lb}_p(w) = x_j (\in X_n); \tag{22}$$

$$a \in A_i,\quad w = vj,\quad \mathrm{lb}_p(v) = f\ (\in F_k \text{ for some } k (\geqq 1) \text{ and } 1 \leqq j \leqq k). \tag{23}$$

We only detail case (21) because the others can be done similarly. Then, from (21) and (17) we obtain

$$af(z_1, \ldots, z_k) \leftarrow q_0'(a_1 z_{i_1}, \ldots, a_l z_{i_l}) \in P_f \quad (0 \leqq i_1, \ldots, i_l \leqq k) \tag{24}$$

and

$$(a_s, w_s) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_s' \quad \text{for all} \quad s(= 1, \ldots, l). \tag{25}$$

Taking into account the relations (18) and (24), by the definition of P'' it follows that

$$(b, a)f(z_1, \ldots, z_k) \leftarrow q_0\left((b_1, a_{j_1}) z_{i_{j_1}}, \ldots, (b_t, a_{j_t}) z_{i_{j_t}}\right) \in P_F''. \tag{26}$$

Since $1 \leqq j_1, \ldots, j_t \leqq l$ thus, from (25),

$$(a_{j_s}, w_{j_s}) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_{j_s}' \tag{27}$$

follows for all $s(=1, \ldots, t)$, moreover, from this and by (20) and the induction hypothesis, we obtain $((b_s, a_{j_s}), w_{j_s}) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} q_{j_s}'$ $(s=1, \ldots, t)$. Finally, from these latters and the derivation $((b, a), w) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} q_0((b_1, a_{j_1}) w_{j_1}, \ldots, (b_t, a_{j_t}) w_{j_t})$ (flowing from (26)) we have what we wanted to prove.

In order to prove the if part of our Theorem let us suppose that the derivation $d=((b, a), w) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} q$ holds and let $\mathrm{lt}(d)=1$. There are four possible cases. Three of them can be specified as (21), (22) and (23) and the fourth is the case $a \in A_i$, $w=\lambda$. Because of similarity, we deal with case (21) only. Since $\mathrm{lt}(d)=1$ thus $(b, a) f \leftarrow q \in P_f''$ follows by the definition of the length of a derivation. From this we get $af(z_1, \ldots, z_k) \leftarrow q_0'(a_1 z_{i_1}, \ldots, a_l z_{i_l}) \in P_f (l \geqq 0;\ 0 \leqq i_1, \ldots, i_l \leqq k;\ q_0' \in T_G(Y_m \cup Z_l))$ and $(b, \lambda) \overset{*}{\underset{q_0', \mathbf{B}}{\Leftarrow}} q$. Consider the derivation $(a, w) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_0'((a_1, w_1), \ldots, (a_l, w_l))$ $\overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_0'(q_1', \ldots, q_l')$, which exists because of $\mathbf{A}$ is completely defiend. If we take the tree $q'=q_0'(q_1', \ldots, q_l')$ then $(b, \lambda) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q$ holds obviously.

Suppose now that $\mathrm{lt}(d)>1$. Then d can be written in the following form:

$$((b, a), w) \underset{p,\mathbf{C}}{\Leftarrow} q_0((b_1, a_1) w_1, \ldots, (b_t, a_t) w_t) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} q_0(q_1, \ldots, q_t) = q. \tag{28}$$

Then three different cases exist, namely (21), (22) and (23). Again, since these cases are similar we deal with case (21) only. In this case, (28) means that

$$(b, a) f(z_1, \ldots, z_k) \leftarrow q_0((b_1, a_1) z_{i_1}, \ldots, (b_t, a_t) z_{i_t}) \in P_f'', \tag{29}$$

moreover,

$$((b_s, a_s), w_s) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} q_s \quad \text{for all} \quad s(=1, \ldots, t). \tag{30}$$

By the definition of P_f'', this implies that

$$af(z_1, \ldots, z_k) \leftarrow q_0'(a_1' z_1', \ldots, a_l' z_l') \in P_f \tag{31}$$

for some $l (\geqq 0)$ and $q_0' (\in T_G(Z_l))$, furthermore,

$$(b, \lambda) \overset{*}{\underset{q_0', \mathbf{B}}{\Leftarrow}} q_0(b_1 z_{i_1}, \ldots, b_t z_{i_t}) \tag{32}$$

and $a_s = a'_{i_s}$, $z_{i_s} = z'_{i_s}$ $(s=1, \ldots, t)$. Then from (31), it follows that

$$(a, w) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_0'((a_1', w_1'), \ldots, (a_l', w_l')), \tag{33}$$

furthermore, $w_s = w'_{i_s}$ for all $s(=1, \ldots, t)$. Then, by the induction hypotesis and (30), we obtain

$$(\exists q_s'')((a_s, w_s) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q_s'' \text{ and } (b, \lambda) \overset{*}{\underset{q_s'', \mathbf{B}}{\Leftarrow}} q_s) \quad \text{for all} \quad s(=1, \ldots, t). \tag{34}$$

Define the trees $q_r' (\in T_G(Y_m))$ $(r=1, \ldots, l)$ as follows

$$q_r' = \begin{cases} q_s'' & \text{if} \quad r = i_s \quad \text{for some} \quad s(=1, \ldots, t) \\ \text{the tree which can be derived from} \\ (a_r', w_r') \text{ in } p \text{ with } \mathbf{A}, \text{ otherwise.} \end{cases}$$

(Note that if in the above definition both $r=i_s$ and $r=i_{s'}$ hold for some $r(=1, \ldots, l)$ and $s \neq s'$ then $w_s = w_{s'}$ holds because of $w'_{i_s} = w'_{i_{s'}}$, and $a_s = a_{s'}$

holds because of $a'_{i_s}=a'_{i_{s'}}$. Thus, also $q''_s=q''_{s'}$. On the other hand if $r=i_s$ does not hold for any s then q'_r exists because **A** is completely defined.)

Finally, we are going to show that the tree $q'=q'_0(q'_1, \dots, q'_t)$ $(\in T_G(Y_m))$ is suitable. Indeed, it follows from (33) and the definition of q'_r $(r=1, \dots, t)$ that $(a, w) \overset{*}{\underset{p,\mathbf{A}}{\Leftarrow}} q'$. Moreover, it follows from (32) that $(b, \lambda) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q_0(b_1v_1, \dots, b_tv_t)$ and $\mathrm{str}_{q'}(v_s)=q'_{i_s}$ for all $s(=1, \dots, t)$. Taking into account that $q'_{i_s}=q''_s$, from (33) and by lemma 3.1, we have $(b, \lambda) \overset{*}{\underset{q',\mathbf{B}}{\Leftarrow}} q$. This ends the proof of the if part.

If we choose $a=a_0, w=\lambda$ then we have $\mathcal{TDA} \circ \mathcal{TDA}_s \subseteq \mathcal{TDA}$.

The equality follows from the fact that every tree transformation being in $\mathcal{TDA}$ can be decomposed by itself and the identity and this latter is in $\mathcal{TDA}_s$. □

If in the former theorem $A_i=\emptyset$ then $C_i=\emptyset$, hence, we obtain

Corollary 4.2. $\mathcal{TDA}_s \circ \mathcal{TDA}_s = \mathcal{TDA}_s$. □

Finally, we want to show that if we apart from the condition that **A** is deterministic in Theorem 4.3 then this equality does not remain valid. Namely, we have the stronger

Theorem 4.4. The classes $\mathcal{TA}_s \circ \mathcal{TDA}_s$ and $\mathcal{TA}$ are incomparable.

Proof. It is easy to show that the tree transformation given in Example 2.1 is not in $\mathcal{TA}_s \circ \mathcal{TDA}_s$.

On the other hand consider the AT transducer $\mathbf{A}(=(T_F(X_1), A, T_G(X_1), a, P))$ where

(i) $F=F_1=\{f\}, \quad G=G_1=\{g_1, g_2\};$

(ii) $A=A_s=\{a\};$

(iii) $P \;=P_f \cup P_{x_1}$, where

$P_f \;=\{af(z_1) \leftarrow g_1(az_1),\ af(z_1) \leftarrow g_2(az_1)\},$

$P_{x_1}=\{ax_1 \leftarrow x_1\};$

and the AT transducer $\mathbf{B}(=(T_G(X_1), B, T_H(X_1), b, P'))$ where

(i) $H=H_1 \cup H_2, \quad H_1=\{h_1\}, \quad H_2=\{h_2\};$

(ii) $B=B_s=\{b\};$

(iii) $P' \;=P'_{g_1} \cup P'_{g_2} \cup P'_{x_1}$ where

$P'_{g_1}=\{bg_1(z_1) \leftarrow h_1(bz_1)\},\ P'_{g_2}=\{bg_2(z_1) \leftarrow h_2(bz_1, bz_1)\},$

$P'_{x_1}=\{bx_1 \rightarrow x_1\}.$

Let $T_1=T(\mathbf{A}), T_2=T(\mathbf{B})$ and $T=T_1 \circ T_2$. Obviously, $T_1 \in \mathcal{TA}_s, T_2 \in \mathcal{TDA}_s$.

Since both **A** and **B** contain only one synthesized attribute it is obvious to show by induction on n that

$$T|_{f^n(x_1)} = \{(f^n(x_1), h_1(q)) | q \in T|_{f^{n-1}(x_1)}\} \cup \{(f^n(x_1), h_2(q, q)) | q \in T|_{f^{n-1}(x_1)}\}$$

for each $n(\geqq 1)$. Taking into account that $T|_{x_1}=(x_1, x_1)$ it is easy to show that the images of tree $f^n(x_1)$ are "simmetrical" for all $n(\geqq 1)$. Moreover, it can be seen that the tree $f^n(x_1)$ has 2^n images, and 2^{n-1} of them are of form $h_2(q, q)$.

Assume that $T=T(\mathbf{C})$ for some AT transducer $\mathbf{C}(=(T_F(X_1), C, T_H(X_1), C_s', P'', rt))$. Let

$$K = |C_s|, \quad L = |C_i| \quad (\text{where } C = C_s \cup C_i),$$

$$M = |\{q \mid q \text{ is the right side of some rule of } P'' \text{ and has the form } h_2(q_1, q_2)\}|.$$

Let us fix an arbitrary integer $n(\geqq 1)$. Consider the derivation of some image $h_2(q, q)$ of the tree $p=f^n(x_1)$. This derivation can be written in the following way:

$$\begin{aligned}(a, \lambda) &\overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} (b, w) \underset{p,\mathbf{C}}{\Leftarrow} h_2(q_1((a_1, v_1), \dots, (a_n, v_n)), q_2((b_1, w_1), \dots, (b_m, w_m))) \\ &\overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} h_2(q_1(r_1, \dots, r_n), q_2(s_1, \dots, s_m)) = h_2(q, q)\end{aligned} \tag{35}$$

for some $a(\in C_s')$. Observe that it holds: if $(a_j, v_j) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} r_j'$ then $r_j = r_j'$ $(j=1, \dots, n)$ and if $(b_k, w_k) \overset{*}{\underset{p,\mathbf{C}}{\Leftarrow}} s_k'$ then $s_k = s_k'$ $(k=1, \dots, m)$. Indeed, in the opposite case if $r_j \neq r_j'$ would hold for some $j(=1, \dots, n)$ then — since the images of p are simmetrical — we should have $q_1(r_1, \dots, r_j, \dots, r_n)=q$ and $q_1(r_1, \dots, r_j', \dots, r_n)=q$ and it is a contradiction.

Thus, each derivation (35) is determined by the attribute occurence (b, w) and the alternative of the rule applied there. As the number of attribute occurences is $(K+L)(n+1)$ and the number of alternatives of a rule is at most M we obtain that the number of images of p, which has the form $h_2(q, q)$, is at most $(K+L)M(n+1)$. This is again a contradiction provided n is sufficiently large. □

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

References

[1] Knuth, D. E., Semantics of context free languages, *Math. Systems Theory*, v. 2, 1968, pp. 127—145.

[2] Engelfriet, J., Some open questions and recent results on tree transducers and tree languages, Memorandum 293, Twente University of Technology, 1980.

[3] Engelfriet, J., Bottom up and top down tree transformations — A comparison, *Math. Systems Theory*, v. 9, 1975, pp. 198—231.

[4] Gécseg, F. and M. Steinby, A faautomaták algebrai elmélete I.—II., *Mat. Lapok*, v. 26, 1975, pp. 169—207; v. 27, 1976—79, pp. 283—336.

(Received March 2, 1981)

On finite nilpotent automata

By B. Imreh

In this paper we consider the isomorphic and homomorphic realizations of finite nilpotent automata. First we characterize all finite subdirectly irreducible nilpotent automata. Secondly we give necessary and sufficient conditions for a system of automata to be isomorphically complete for the class of all finite nilpotent automata with respect to the α_i-products (see [2]). Finally, we characterize the homomorphically complete systems for the class of all finite nilpotent automata with respect to the α_i-products.

The terminology and notations will be used in accordance with [3]. By an automaton we always mean a finite automaton. It can be seen from the definition of nilpotent automata that if $\mathbf{A}=(X, A, \delta)$ is a nilpotent automaton with absorbent state a_0 then

(i) $\mathbf{A}$ is connected in the sense that for any $a, b \in A$ there exist $p, q \in X^*$ with $ap=bq$,

(ii) the binary relation $a \leqq b \Leftrightarrow (\exists p)\ (p \in X^*\ \&\ ap=b)$ is a partial ordering in A and a_0 is the greatest element in $(A, \leqq)$.

Theorem 1. A nilpotent automaton $\mathbf{A}=(X, A, \delta)$ $(|A| \geqq 2)$ is subdirectly irreducible if and only if

(1) there exists an $a_1 \in A \setminus \{a_0\}$ such that a_1 is the greatest element in $(A \setminus \{a_0\}, \leqq)$,

(2) for any $a, b \in A$ if $a \neq b$ and $\{a, b\} \not\subseteq \{a_0, a_1\}$ then there exists a $p \in X^+$ with $ap \neq bp$.

Proof. Theorem 1 will be proved in a similar way as the corresponding statement for commutative automata in [1].

In order to prove the necessity assume that $\mathbf{A}$ is subdirectly irreducible and (1) does not hold. Then $(A \setminus \{a_0\}, \leqq)$ has at least two maximal elements. Denote them by a_1 and a_2. Consider the following relations: for any $a, b \in A$

$a \sigma_1 b$ if and onyl if $\{a, b\} \subseteq \{a_0, a_1\}$ or $a=b$,

$a \sigma_2 b$ if and only if $\{a, b\} \subseteq \{a_0, a_2\}$ or $a=b$.

It is not difficult to see that σ_1 and σ_2 are nontrivial congruence relations of $\mathbf{A}$ and $\sigma_1 \cap \sigma_2 = \Delta_A$, where Δ_A denotes the equality relation of A. This is a contradiction. Now assume that (1) holds and (2) does not hold. Then there exist $u, v \in A$ such that $u \neq v$, $\{u, v\} \nsubseteq \{a_0, a_1\}$ and $up = vp$ for any $p \in X^+$. Consider the following relations: for any $a, b \in A$

$a\sigma_1 b$ if and only if $\{a, b\} \subseteq \{a_0, a_1\}$ or $a = b$,

$c\sigma_2 b$ if and only if $\{a, b\} \subseteq \{u, v\}$ or $a = b$.

It is clear that σ_1 and σ_2 are nontrivial congruence relations of **A** and $\sigma_1 \cap \sigma_2 = \Delta_A$, which is a contradiction.

To prove the sufficiency assume that (1) and (2) are satisfied by **A**, and **A** is subdirectly reducible. Then there exists a congruence relation ϱ of **A** such that $\varrho \neq \Delta_A$ and $a_0 \not\equiv a_1(\varrho)$. By $\varrho \neq \Delta_A$, there exist $u \neq v \in A$ with $u \equiv v(\varrho)$. Consider the nonvoid set

$$B = \{\{a, b\}: a, b \in A,\ a \neq b,\ (\exists p)\ (p \in X^* \text{ and } \{u, v\}p = \{a, b\})\}.$$

Define the binary relation $\leqq$ on B as follows: $\{a, b\} \leqq \{a', b'\}$ if and only if there is a word $p \in X^*$ satisfying $\{a, b\}p = \{a', b'\}$. It is obvious that $\leqq$ is a partial ordering in B. Let $\{\bar{a}, \bar{b}\}$ denote a maximal element of B. Then, by the definition of B, $\bar{a} \neq \bar{b}$ and $\bar{a} \equiv \bar{b}(\varrho)$. Therefore, $\{\bar{a}, \bar{b}\} \nsubseteq \{a_0, a_1\}$. On the other hand, $\{\bar{a}, \bar{b}\}$ is a maximal element in $(B, \leqq)$, thus, $\bar{a}p = \bar{b}p$ for any $p \in X^+$, contradicting condition (2). This ends the proof of Theorem 1.

By Theorem 1, we can give all subdirectly irreducible nilpotent automata directly. Indeed, let $m \geqq 1$ be a fixed natural number and consider the input set $X_m = \{x_1, \ldots, x_m\}$. Take the sets $A_1^{(m)} = \{0\}$, $A_2^{(m)} = \{0, 1\}$,

$$A_{n+1}^{(m)} = A_n^{(m)} \cup \{(u_1, \ldots, u_m): u_1, \ldots, u_m \in A_n^{(m)} \text{ and } \{u_1, \ldots, u_m\} \cap (A_n^{(m)} \setminus A_{n-1}^{(m)}) \neq \emptyset\}$$

for all $n \geqq 2$. Now, define the automata $\mathbf{A}_n^{(m)}$ $n = 1, 2, \ldots$ in the following way:

$$\mathbf{A}_1^{(m)} = (X_m, A_1^{(m)}, \delta_1), \text{ where } \delta_1(0, x_t) = 0 \quad (t = 1, \ldots, m),$$

$$\mathbf{A}_2^{(m)} = (X_m, A_2^{(m)}, \delta_2), \text{ where } \delta_2(0, x_t) = \delta_2(1, x_t) = 0 \quad (t = 1, \ldots, m),$$

and in case of $n > 2$

$$\mathbf{A}_n^{(m)} = (X_m, A_n^{(m)}, \delta_n) \text{ with } \delta_{n|A_{n-1}^{(m)} \times X_m} = \delta_{n-1} \text{ and } \delta_n((u_1, \ldots, u_m), x_t) = u_t$$

$(t = 1, \ldots, m)$ for any $(u_1, \ldots, u_m) \in A_n^{(m)} \setminus A_{n-1}^{(m)}$, where the restriction of δ_n to $A_{n-1}^{(m)} \times X_m$ is denoted by $\delta_{n|A_{n-1}^{(m)} \times X_m}$.

Using Theorem 1 it is not difficult to prove that a nilpotent automaton **A** with the input set X_m is subdirectly irreducible if and only if there exists a natural number n such that **A** can be embedded isomorphically into a quasi-direct product $\mathbf{A}_n^{(m)}(X_m, \varphi)$ of $\mathbf{A}_n^{(m)}$ with a single factor. From this we get the following

Corollary. A system Σ of automata is isomorphically complete for the class of all nilpotent automata with respect to the quasi-direct product if and only if for any pair (n, m) of natural numbers $n, m \geqq 1$ there is an automaton $\mathbf{A} \in \Sigma$ such that $\mathbf{A}_n^{(m)}$ can be embedded isomorphically into a quasi-direct product of **A** with a single factor.

This Corollary shows that there exists no system of automata which is isomorphically complete for the class of all nilpotent automata with respect to the quasi-direct product and minimal.

We say that an automaton **A** can be *realized homomorphically* by an α_i-product of automata $\mathbf{A}_t$ $(t=1, \ldots, k)$ if there exists a subautomaton **B** of an α_i-product of automata $\mathbf{A}_t$ $(t=1, \ldots, k)$ such that **A** is a homomorphic image of **B**.

We are going to use the following obvious statements.

Lemma 1. If an automaton **A** can be embedded isomorphically into an α_0-product of automata $\mathbf{A}_t$ $(t=1, \ldots, k)$ and for some $1 \leqq i \leqq k$ the automaton $\mathbf{A}_i$ can be embedded isomorphically into an α_0-product of automata $\mathbf{B}_j$ $(j=1, \ldots, s)$ then the automaton **A** can be embedded isomorphically into an α_0-product of automata $\mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{B}_1, \ldots, \mathbf{B}_s, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_k$.

Lemma 2. If an automaton **A** can be realized homomorphically by an α_0-product of automata $\mathbf{A}_t$ $(t=1, \ldots, k)$ and for some $1 \leqq i \leqq k$ the automaton $\mathbf{A}_i$ can be realized homomorphically by an α_0-product of automata $\mathbf{B}_j$ $(j=1, \ldots, s)$ then the automaton **A** can be realized homomorphically by an α_0-product of automata $\mathbf{A}_1, \ldots, \mathbf{A}_{i-1}, \mathbf{B}_1, \ldots, \mathbf{B}_s, \mathbf{A}_{i+1}, \ldots, \mathbf{A}_k$.

Let us denote by $\mathbf{R}_n = (\{x_1, \ldots, x_{n-1}\}, \{1, \ldots, n\}, \delta_n)$ the automaton, where $\delta_n(t, x_s) = \min(t+s, n)$ for any $1 \leqq t \leqq n$, $x_s \in \{x_1, \ldots, x_{n-1}\}$ and $n \geqq 2$.

Concerning the isomorphic realizations of nilpotent automata with respect to the α_0-product we have the following result.

Theorem 2. A system Σ of automata is isomorphically complete for the class of all nilpotent automata with respect to the α_0-product if and only if one of the following four conditions is satisfied by Σ:

(1) there exists an automaton in Σ which has three different states b, c, d and four input signs y, z, v, w (need not be different) such that $by=b, bz=c, cv=dv=bw=d$ hold,

(2) Σ contains an automaton which has two different states b, c and two input signs y, z such that $b=cy=by$ and $bz=c$ hold,

(3) Σ contains an automaton which has two different states b, c and two input signs y, z with $b=by, bz=cz=c$,

(4) for any natural number $n \geqq 3$ there exists an automaton in Σ which has n different states a_t $(t=1, \ldots, n)$ and input signs $x_k^{(t)}$ $(t=1, \ldots, n-1)$ $(k=1, \ldots, n-t)$ such that $a_t x_k^{(t)} = a_{t+k}$ if $1 \leqq t \leqq n-1$, $1 \leqq k \leqq n-t$ furthermore, $a_n x_1^{(n-1)} = a_n$ hold.

Proof. In order to prove the necessity assume that Σ is isomorphically complete for the class of all nilpotent automata with respect to the α_0-product. Let $n \geqq 3$ be arbitrary and consider the automaton $\mathbf{R}_n$. Since $\mathbf{R}_n$ is nilpotent, by our assumption, $\mathbf{R}_n$ can be embedded isomorphically into an α_0-product $\prod_{t=1}^{s} \mathbf{A}_t(\{x_1, \ldots, x_{n-1}\}, \varphi)$ of automata from Σ. Let μ denote a suitable isomorphism, and for any $i \in \{1, \ldots, n\}$ let $(a_{i1}, \ldots, a_{is})$ be the image of i under μ. Denote by m the least index for which $a_{nm} \neq a_{n-1m}$ holds. Observe that if $a_{im} = a_{nm}$ for some $1 \leqq i < n-1$ then (2) holds, while (3) holds if $a_{im} = a_{n-1m}$. Furthermore, if $a_{im} \notin \{a_{n-1m}, a_{nm}\}$ $(i=1, \ldots, n-2)$ and $a_{im} = a_{jm}$ for some indices $1 \leqq i < j < n-1$ then Σ satisfies condition (1) by $\mathbf{A}_m$. In the remaining case the elements a_{im} $(i=1, \ldots, n)$ are pairwise different and this implies that $\mathbf{A}_m$ has the property required in (4). Therefore, since n was arbitrary, if none of conditions (1), (2) and (3) is satisfied by Σ then (4) holds.

We have already shown the necessity of our statement. Conversely, assume that (1) holds by $\mathbf{B}\in\Sigma$. We shall prove that every nilpotent automaton can be embedded isomorphically into an α_0-power of $\mathbf{B}$. We proceed by induction on the number of states of the automaton. The case $n\leqq 2$ is trivial. Now let $n>2$ and assume that for any $m<n$ the statement is valid. Denote by $\mathbf{A}=(X, A, \delta)$ an arbitrary nilpotent automaton with n states. If $\mathbf{A}$ is subdirectly reducible then $\mathbf{A}$ can be embedded isomorphically into a direct product of nilpotent automata with fewer states than n. Therefore, by our induction hypothesis and Lemma 1, the statement is valid. Now assume that $\mathbf{A}$ is subdirectly irreducible. Then $\mathbf{A}$ has elements a_0 and a_1 satisfying (1) in Theorem 1. Define the congruence relation σ of $\mathbf{A}$ in the following manner: for any $a, b\in A$ $a\sigma b$ if and only if $\{a, b\}\subseteq\{a_0, a_1\}$ or $a=b$. The quotient automaton $\mathbf{A}_1=\mathbf{A}/\sigma$ is nilpotent with $n-1$ states. Consider the α_0-product $\mathbf{A}_1\times\mathbf{B}(X, \varphi)$, where $\varphi_1(x)=x$ and

$$\varphi_2(\sigma(a), x)=\begin{cases} y & \text{if } \sigma(a)\neq\sigma(a_0) \text{ and } \delta(a, x)\in A\setminus\sigma(a_0), \\ z & \text{if } \sigma(a)\neq\sigma(a_0) \text{ and } \delta(a, x)=a_1, \\ w & \text{if } \sigma(a)\neq\sigma(a_0) \text{ and } \delta(a, x)=a_0, \\ v & \text{if } \sigma(a)=\sigma(a_0), \end{cases}$$

for any $x\in X$, $\sigma(a)\in A/\sigma$. It can be easily proved that the correspondence

$$\nu(a)=\begin{cases} (\sigma(a), b) & \text{if } a\in A\setminus\sigma(a_0), \\ (\sigma(a), c) & \text{if } a=a_1, \\ (\sigma(a), d) & \text{if } a=a_0, \end{cases}$$

is an isomorphism of $\mathbf{A}$ into the α_0-product $\mathbf{A}_1\times\mathbf{B}(X, \varphi)$. Therefore, by our induction assumption and Lemma 1, $\mathbf{A}$ can be decomposed in the required form.

The sufficiencies of conditions (2) and (3) can be proved in a similar way as the sufficiency of (1).

Now assume that condition (4) holds. We proceed by induction on the number of states of the automaton. The case $n\leqq 2$ is trivial. Let $n>2$ and assume that the statement is valid for any $v<n$. Denote by $\mathbf{A}=(X, A, \delta)$ an arbitrary nilpotent automaton with n states. If $\mathbf{A}$ is subdirectly reducible then, by our induction assumption and Lemma 1, the statement is valid. Now assume that $\mathbf{A}$ is subdirectly irreducible and let $X=\{x_1, \ldots, x_m\}$. Then, by the observation connecting with Theorem 1, there exists an automaton $\mathbf{A}_s^{(m)}$ such that $\mathbf{A}$ can be embedded isomorphically into $\mathbf{A}_s^{(m)}(X_m, \psi)$. Denote by $\bar{s}$ the least natural number for which $\mathbf{A}$ can be embedded isomoriphcally into $\mathbf{A}_{\bar{s}}^{(m)}(X_m, \psi)$. Let μ denote a suitable isomorphism. Since Σ satisfies (4) there exists an automaton $\mathbf{B}\in\Sigma$ which has $\bar{s}$ different states a_j ($j=1, \ldots, \bar{s}$) and input signs $x_k^{(t)}$ ($t=1, \ldots, \bar{s}-1$) ($k=1, \ldots, \bar{s}-t$) such that $a_t x_k^{(t)}= =a_{t+k}$ ($t=1, \ldots, \bar{s}-1$) ($k=1, \ldots, \bar{s}-t$) and $a_{\bar{s}} x_1^{(\bar{s}-1)}=a_{\bar{s}}$ hold. Now consider the α_0-product $\mathbf{A}_1\times\mathbf{B}(X, \varphi)$ where $\mathbf{A}_1$ is defined in the same way as above and $\varphi_1(x)=x$,

$$\varphi_2(\sigma(a), x)=\begin{cases} x_{i-j}^{(\bar{s}-i+1)} & \text{if } \mu(a)\in A_i\setminus A_{i-1} \text{ for some } 3\leqq i\leqq\bar{s} \text{ and} \\ & \mu(\delta(a, x))\in A_j\setminus A_{j-1} \text{ for some } 1<j<i \text{ or} \\ & \mu(\delta(a, x))\in A_j \text{ with } j=1, \\ x_1^{(\bar{s}-1)} & \text{if } \sigma(a)=\sigma(a_0), \end{cases}$$

for any $x \in X$, $\sigma(a) \in A/\sigma$. It is not difficult to prove that the correspondence

$$\nu(a) = \begin{cases} (\sigma(a), a_{\bar{s}-i+1}) & \text{if } \mu(a) \in A_i \setminus A_{i-1} \text{ for some } 3 \leqq i \leqq \bar{s}, \\ (\sigma(a_0), a'_{\bar{s}-1}) & \text{if } \mu(a) \in A_2 \setminus A_1, \\ (\sigma(a_0), a_{\bar{s}}) & \text{if } \mu(a) \in A_1, \end{cases}$$

is an isomorphism of **A** into the α_0-product $\mathbf{A}_1 \times \mathbf{B}(X, \varphi)$. Thus, by our induction assumption and Lemma 1, we have a required decomposition of **A**. This completes the proof of Theorem 2.

The following theorem holds for α_i-products with $i \geqq 1$.

Theorem 3. A system Σ of automata is isomorphically complete for the class of all nilpotent automata with respect to the α_i-product $(i \geqq 1)$ if and only if one of the following three conditions is satisfied by Σ:

(1) there exists an automaton in Σ which has two different states b, c and three input signs y, z, v (need not be different) such that $by = b$ and $bz = cv = c$ hold,

(2) Σ contains an automaton which has two different states b, c and three input signs y, z, v (need not be different) such that $by = cv = b$ and $bz = c$ hold,

(3) for any natural number $n \geqq 3$ there exists an automaton in Σ which has n different states a_j $(j = 1, \ldots, n)$ and input signs $x_k^{(t)}$ $(t = 1, \ldots, n-1)$ $(k = 1, \ldots, n-t)$, y such that $a_t x_k^{(t)} = a_{t+k}$ $(t = 1, \ldots, n-1)$ $(k = 1, \ldots, n-t)$ and $a_n y = a_n$.

Proof. The necessity of these conditions can be proved in a similar way as in the proof of Theorem 2. To prove the sufficiency, again, by Theorem 2, it is enough to show that an α_0-product of α_1-products with single factors is an α_1-product. But this is immediate from the definition of the α_i-products.

For any natural, number $n \geqq 1$ denote by $\mathbf{I}_n = (\{x\}, \{1, \ldots, n\}, \delta_n)$ the automaton satisfying $\delta_n(i, x) = \min(i+1, n)$ for all $i \in \{1, \ldots, n\}$. Furthermore, for any natural number $n \geqq 3$ denote by $\mathbf{Q}_n = (\{x, y\}, \{1, \ldots, n\}, \delta_n)$ the automaton for which $\delta_n(i, x) = \delta_n(i, y) = \min(i+1, n)$ for all $i \neq n-2$, $i \in \{1, \ldots, n\}$ and $\delta_n(n-2, x) = n-1$, $\delta_n(n-2, y) = n$.

In the sequel we shall need a more general concept of a subautomaton. The automaton $\mathbf{B} = (Y, B, \delta')$ is an *X-subautomaton* of $\mathbf{A} = (X, A, \delta)$ if $Y \subseteq X$, $B \subseteq A$ and $\delta_{|B \times Y} = \delta'$.

Take an automaton $\mathbf{A} = (X, A, \delta)$. Let $a \in A$ and $x \in X$ be arbitrary. The X-subautomaton generated by a and x is called a *cycle* and it will be denoted by (a, x). (Also, this notation (a, x) will be used to denote the set of states of this X-subautomaton.) For a cycle (a, x) there exist natural numbers $n \geqq 1$ and $m \geqq 1$ such that

(i) $n-1$ is the least exponent for which there exists a $t > n-1$ with $ax^{n-1} = ax^t$,

(ii) m is the least nonzero natural number for which $ax^{n-1} = ax^{n+m-1}$ holds,

(iii) the states $a, ax, \ldots, ax^{n+m-2}$ are pairwise different.

In this case we say that (a, x) is a *cycle of type* (n, m).

Observe an important property of cycles which we are going to use in the proofs of Theorems 4 and 5. Let $\mathbf{A} = (a, x)$ be a cycle of type (n, m) and let $\mathbf{B} = (b, x)$ be a cycle of type $(\bar{n}, \bar{m})$, where **A** and **B** have the same input sign x. Then the automaton **B** is a homomorphic image of **A** if and only if $\bar{n} \leqq n$ and $\bar{m} | m$ hold.

Theorem 4. A system Σ of automata is homomorphically complete for the class of all nilpotent automata with respect to the α_0-product if and only if one of the following three conditions is satisfied by Σ:

(1) there exists an automaton in Σ which has states b, c, d, input sign z and input words p, r, q such that $|p|\geqq 1, b\neq c, bz=cz, czq=c, dp=d$ and $dr=b$ hold,

(2) (i) Σ contains an automaton which has a state b and input signs $x_1, \dots, x_k, y$ such that $bx_1\dots x_k=b$ and $bx_1\neq by$,

(ii) for any natural number $n\geqq 3$ there exist a nonzero natural number m and an automaton in Σ having $n+m-1$ different states a_t $(t=1, \dots, n+m-1)$ and input signs x_t $(1\leqq t<n+m-1)$ for which $a_tx_t=a_{t+1}$ $(1\leqq t<n+m-1)$ and $a_{n+m-1}x_{n-1}=a_n$ hold,

(3) (i) for any natural number $n\geqq 3$ there exists an automaton in Σ which has n different states b_t $(t=1, \dots, n)$ and input signs x_t $(1\leqq t<n)$ such that $b_tx_t=b_{t+1}$ if $1\leqq t\leqq n-2$ and $b_{n-2}x_{n-1}=b_n$,

(ii) for any natural number $n\geqq 3$ there exist $m\geqq 1$ and an automaton in Σ having $n+m-1$ different states a_t $(t=1, \dots, n+m-1)$ and input signs x_t $(1\leqq t<n+m-1)$ for which $a_tx_t=a_{t+1}$ $(1\leqq t<n+m-1)$ and $a_{n+m-1}x_{n-1}=a_n$ hold.

Proof. In order to prove the necessity assume that Σ is homomorphically complete for the class of all nilpotent automata with respect to the α_0-product. If Σ satisfies condition (1) then we are ready. Consider the case when Σ does not satisfy condition (1). We shall show that in this case (2) (ii) and, henceforth, (3) (ii) also hold. Indeed, let $n\geqq 3$ and consider the automaton $\mathbf{I}_n$. As Σ is homomorphically complete $\mathbf{I}_n$ can be realized homomorphically by an α_0-product of automata from Σ, i.e. there exists a subautomaton $\mathbf{A}$ of an α_0-product of automata from Σ such that $\mathbf{I}_n$ is a homomorphic image of $\mathbf{A}$. Let us denote by $\prod_{t=1}^{s} \mathbf{A}_t(\{x\}, \varphi)$ such an α_0-product and let μ be a suitable homomorphism. Let a be a counter image of the state 1 under μ, i.e. $\mu(a)=1$. Consider the cycle (a, x) in $\mathbf{A}$. It is obvious that (a, x) is a cycle of type $(\bar{n}, m)$ for some $m\geqq 1$ and $\bar{n}\geqq n$. From this we get that a cycle of type (n, m) can be embedded isomorphically into the α_0-product $\prod_{t=1}^{s} \mathbf{A}_t(\{x\}, \varphi)$. Let us denote by $\mathbf{B}=(b, x)$ the cycle of type (n, m) and by ν the isomorphism in question. Further on, we write $b_1=b, b_{t+1}=bx^t(1\leqq t<n+m-1)$. For any t $(1\leqq t\leqq n+m-1)$ let $(a_{t1}, \dots, a_{ts})$ be the image of b_t under ν. Now consider the congruence relations $\pi_1\geqq\pi_2\geqq\dots\geqq\pi_s$ on $\mathbf{B}$ which are defined in the following way: for any $1\leqq r\leqq s$ $b_i\equiv b_j(\pi_r)$ $b_i, b_j\in(b, x)$ if and only if $a_{it}=a_{jt}$ $(t=1, \dots, r)$. Since the quotient automaton $\mathbf{B}/\pi_r$ is a homomorphic image of $\mathbf{B}$ we obtain that $\mathbf{B}/\pi_r$ is a cycle of type (n_r, m_r) for some natural numbers n_r, m_r, where $n_r\leqq n$ and $m_r|m$. On the other hand, by $\pi_1\geqq\pi_2\geqq\dots\geqq\pi_s$ we get $n_1\leqq \leqq n_2\leqq\dots\leqq n_s=n$. Now, if $n_1=n$ then the automaton $\mathbf{A}_1$ has the property required in (2) (ii). In the opposite case there exists a natural number r $(1\leqq r<s)$ such that $n_r<n$ and $n_{r+1}=n$. It is not difficult to see that in this case a cycle of type (n, m_{r+1}) can be embedded isomorphically into the α_0-product $\mathbf{B}/\pi_r\times\mathbf{A}_{r+1}(\{x\}, \psi)$, where $\psi_1(x)=x$, $\psi_2(\pi_r(b_t), x)=\varphi_{r+1}(a_{t1}, \dots, a_{tr}, x)$ for any $\pi_r(b_t)\in B/\pi_r$. For the sake of simplicity let $m_{r+1}=m$ and denote by $\mathbf{D}=(d, x)$ and $\mathbf{C}=(c, x)$ a cycle of type (n, m) and (n_r, m_r), respectively. Therefore, we obtain that $\mathbf{D}$ can be embedded isomorphically into an α_0-product $\mathbf{C}\times\mathbf{A}_{r+1}(\{x\}, \varphi')$ under a suitable isomorphism τ.

We write $d_1=d$, $d_{t+1}=dx^t$ $(1\leqq t<n+m-1)$, and for any t $(1\leqq t\leqq n+m-1)$ let (c_t, a_t) be the image of d_t under τ. Since $n_r<n$ and $m_r|m$, $c_{n-1}=c_{n+m-1}$. From this it follows that $a_{n-1}\neq a_{n+m-1}$ and $\delta_{r+1}(a_{n-1}, z)=\delta_{r+1}(a_{n+m-1}, z)$ for some input sign $z\in X_{r+1}$. Now observe that the states $a_1, \dots, a_n$ are pairwise different and $\{a_1, \dots, a_{n-1}\}\cap\{a_n, \dots, a_{n+m-1}\}=\emptyset$. Indeed, in the opposite case it can easily be seen that the automaton $\mathbf{A}_{r+1}$ has the property required in (1) and this is a contradiction. On the other hand, if $a_1, \dots, a_n$ are pairwise different and $\{a_1, \dots, a_{n-1}\}\cap \cap\{a_n, \dots, a_{n+m-1}\}=\emptyset$ then it is not difficult to prove that $\mathbf{A}_{r+1}$ satisfies the conditions required in (2) (ii). Since n was arbitrary we get that Σ satisfies condition (2) (ii).

Now assume that Σ does not satisfy condition (2) (i). We shall show that in this case (3) holds. Indeed, let $n\geqq 3$ be arbitrary and consider the automaton $\mathbf{Q}_n$. By our assumption, $\mathbf{Q}_n$ can be realized homomorphically by an α_0-product $\prod_{t=1}^{s} \mathbf{A}_t(\{x, y\}, \varphi)$ of automata from Σ. Denote by μ a suitable homomorphism. Let b be a counter image of the state 1 under μ. Consider the states $b_1=b$, $b_{t+1}=bx^t$ $(1\leqq t<n-1)$, $b_n=b_{n-2}y$ in the α_0-product. They are pairwise different since their images under μ are pairwise different. Let $b_t=(a_{t1}, \dots, a_{ts})$ for any t $(1\leqq t\leqq n)$. Denote by k the least index for which $a_{n-1k}\neq a_{nk}$. It can be easily seen that if there exist indices i, j $(1\leqq i<j\leqq n)$ with $a_{ik}=a_{jk}$ then Σ satisfies (2) (i) by $\mathbf{A}_k$, which is a contradiction. Therefore, the states a_{tk} $(1\leqq t\leqq n)$ are pairwise different. Then $\mathbf{A}_k$ has the property required in (3) (i). Since n is arbitrary we obtain that Σ satisfies (3). This ends the proof of the necessity.

The proof of sufficiency consists of two steps. First we shall show that if one of the conditions (1), (2), (3) is satisfied by Σ then the automaton $\mathbf{Q}_n$ can be realized homomorphically by an α_0-product of automata from Σ for any $n\geqq 3$. Secondly, it is proved that every nilpotent automaton can be realized homomorphically by an α_0-product of automata from $\{\mathbf{Q}_n: n\geqq 3\}$. By Lemma 2, this will complete the proof of sufficiency.

Indeed, suppose that Σ satisfies (1) by the automaton $\mathbf{A}(\in\Sigma)$. We show that the automaton $\mathbf{I}_n$ can be realized homomorphically by an α_0-power of $\mathbf{A}$ for any $n\geqq 2$. This statement is proved by induction on n. Let $n=2$ and take the states $b, c(\in A)$ and the input sign z of $\mathbf{A}$ for which $b\neq c$ and $bz=cz$. Consider the cycle (b, z). Let (k, l) be the type of (b, z). If $k>1$, then $\mathbf{I}_2$ can be realized homomorphically by an α_0-product of (b, z) with a single factor. If $k=1$ then, by $b\neq c$ and $bz=cz$, it can be easily seen that $c\notin(b, z)$. In this case $\mathbf{I}_2$ can be realized homomorphically by an α_0-product of (c, z) with a single factor. Now let $n>2$ and assume that our statement is valid for any $m<n$. We distinguish two cases depending on the value of k.

First suppose that $k>1$ in the type (k, l) of (b, z). Since Σ satisfies (1) by $\mathbf{A}$, there exist a state $d(\in A)$ and input words p, r with $|p|\geqq 1$, $dp=d$, $dr=b$. Let $p=x_1\dots x_i$ and let $r=y_1\dots y_j$ if r is nonempty. Consider the α_0-product $\mathbf{I}_{n-1}\times \times\mathbf{A}(\{x\}, \varphi)$, where $\varphi_1(x)=x$,

$$\varphi_2(t, x)=\begin{cases} z & \text{if } t=n-1,\\ y_{j-v} & \text{if } |r|\geqq 1 \text{ and } t=n-2-v \text{ for some } 0\leqq v<j,\\ x_{i-v} & \text{if } t=n-2-|r|-ui-v \quad [\text{for some } 0\leqq v<i \text{ and } u=0, 1, \dots \end{cases}$$

for all $1 \leqq t < n$. Define the state a of **A** in the following manner: $a = dy_1 \dots y_v$ if $|r| \geqq 1$ and $n = j+2-v$ for some $1 \leqq v < j$; $a = d$ if $|r| \geqq 1$, $n = j+2$; $a = dx_1 \dots x_{i-v}$ if $|r| \geqq 1$ and $n = j+2+ui+v$ for some $0 \leqq v < i$ and $u = 0, 1, \dots$; $a = bx_1 \dots x_{i-v}$ if $|r| = 0$ and $n = 2+ui+v$ for some $0 \leqq v < i$ and $u = 0, 1, \dots$. It can be easily seen that $\mathbf{I}_n$ is a homomorphic image of the subautomaton generated by $(1, a)$ in the α_0-product $\mathbf{I}_{n-1} \times \mathbf{A}(\{x\}, \varphi)$. From this, by our induction assumption and Lemma 2, we obtain a required decomposition of $\mathbf{I}_n$.

Now assume that $k = 1$. In this case $c \notin (b, z)$ and thus, by $cz = bz$, we have $c \neq cz$. On the other hand $czq = c$, thus $q = z_1 \dots z_i$ where $i \geqq 1$. Consider the α_0-product $\mathbf{I}_{n-1} \times \mathbf{A}(\{x\}, \varphi)$, where $\varphi_1(x) = x$ and

$$\varphi_2(t, x) = \begin{cases} z & \text{if } t = n-1-u(i+1) \quad \text{for some } u = 0, 1, \dots, \\ z_{i-v+1} & \text{if } t = n-1-u(i+1)-v \text{ for some } 1 \leqq v \leqq i \text{ and } u = 0, 1, \dots \end{cases}$$

for all $1 \leqq t < n$. Define the state $a (\in A)$ in the following way: $a = cz$ if $n = 1 + u(i+1)$ for some $u = 0, 1, \dots$; $a = czz_1 \dots z_{i-v+1}$ if $n = 1 + u(i+1) + v$ for some $1 \leqq v \leqq i$ and $u = 0, 1, \dots$. It is not difficult to see that $\mathbf{I}_n$ is a homomorphic image of the subautomaton generated by $(1, a)$ in $\mathbf{I}_{n-1} \times \mathbf{A}(\{x\}, \varphi)$. This yields a required decomposition of $\mathbf{I}_n$.

Now let $n \geqq 3$ be arbitrary and consider the automaton $\mathbf{Q}_n$. We know that $dx_1 \dots x_i = d$. We write $d = d_1$ and $d_{t+1} = d_t x_t$ $(1 \leqq t < i)$. Without loss of generality we may assume that the states $d_1, \dots, d_i$ are pairwise different. We show that there exist an index j $(1 \leqq j \leqq i)$ and an input sign w of **A** such that $d_j x_j \neq d_j w$. Indeed, in the opposite case $d_t x_t = d_t x$ holds for any input sign x and d_t $(t = 1, \dots, i)$. Since $d_1 r = b$ and $d_1 rzq = bzq = czq = c$, there exist $1 \leqq t_1, t_2 \leqq i$ with $b = d_{t_1}$ and $c = d_{t_2}$. On the other hand, $bz = cz$ from which $t_1 = t_2$ and, henceforth $b = c$ follows, yielding a contradiction. (Observe that we have proved that **A** has the property required in (2) (i).) Now let j $(1 \leqq j \leqq i)$ denote an index such that $d_j x_j \neq d_j w$ for some input sign w of **A**. Take the following α_0-product $\mathbf{I}_n \times \mathbf{A}(\{x, y\}, \varphi)$, where $\varphi_1(x) = x$,

$$\varphi_2(t, x) = \begin{cases} x_{j+v} & \text{if } i > 1 \text{ and } t = n-2+v \quad \text{for some } 0 \leqq v \leqq i-j, \\ x_{j+v-i} & \text{if } i > 1 \text{ and } t = n-2+v \quad \text{for some } i-j < v \leqq 2, \\ x_{j-v} & \text{if } i > 1 \text{ and } t = n-2-v \quad \text{for some } 1 \leqq v < j, \\ x_{i-v} & \text{if } i > 1 \text{ and } t = n-2-j-ui-v \text{ for some } 0 \leqq v < i \\ & \qquad \text{and } u = 0, 1, \dots, \\ x_1 & \text{if } i = 1, \end{cases}$$

and

$$\varphi_2(t, y) = \begin{cases} w & \text{if } t = n-2, \\ \varphi_2(t, x) & \text{otherwise,} \end{cases}$$

for all $1 \leqq t \leqq n$. Define the state $a (\in A)$ in the following manner: $a = d_{v+1}$ if $i > 1$ and $n = j+2-v$ for some $0 \leqq v < j$; $a = d_{i-v}$ if $i > 1$ and $n = j+3+ui+v$ for some $0 \leqq v < i$ and $u = 0, 1, \dots$; $a = d_1$ if $i = 1$. It can be easily seen that the automaton $\mathbf{Q}_n$ is a homomorphic image of the subautomaton generated by $(1, a)$ in $\mathbf{I}_n \times \mathbf{A}(\{x, y\}, \varphi)$. By Lemma 2, we got a required decomposition of $\mathbf{Q}_n$, and thus we have proved the homomorphic realizations of automata $\mathbf{Q}_n$ by Σ if Σ satisfies condition (1).

Now assume that Σ satisfies condition (2). First we show that the automaton $\mathbf{I}_n$ can be realized homomorphically by an α_0-product of automata from Σ for any $n\geqq 2$. We prove this by induction on n. Let $n=2$. Since Σ satisfies (2) (ii), there exists an automaton $\mathbf{A}$ in Σ which has $m+2$ different states $a_1, \dots, a_{m+2}$ and input signs x_t $(1\leqq t<m+2)$ such that $a_t x_t = a_{t+1}$ if $1\leqq t<m+2$, and $a_{m+2}x_2=a_3$. Take the cycle (a_2, x_2) in $\mathbf{A}$. If the type of (a_2, x_2) is (k, l) with $k>1$ then $\mathbf{I}_2$ is a homomorphic image of an α_0-product of (a_2, x_2) with a single factor, and thus $\mathbf{I}_2$ can be realized homomorphically by an α_0-product of $\mathbf{A}$ with a single factor. In the opposite case, it is not difficult to see that $a_{m+2}\notin(a_2, x_2)$, and thus $\mathbf{I}_2$ is a homomorphic image of an α_0-product of the cycle (a_{m+2}, x_2) with a single factor. Therefore, $\mathbf{I}_2$ can be realized homomorphically by an α_0-product of $\mathbf{A}$ with a single factor. Now let $n>2$ and assume that our statement is valid for any $j<n$. Since Σ satisfies (2) (ii) there exists an automaton $\mathbf{A}$ in Σ having different states $a_t(t=1, \dots, n+m-1)$ and input signs x_t $(1\leqq t<n+m-1)$ such that $a_t x_t=a_{t+1}$ if $1\leqq t<n+m-1$ and $a_{n+m-1}x_{n-1}=a_n$. We distinguish two cases.

First assume that $k>1$ in the type (k, l) of the cycle (a_{n-1}, x_{n-1}). Consider the α_0-product $\mathbf{I}_{n-1}\times\mathbf{A}(\{x\}, \varphi)$, where $\varphi_1(x)=x$ and $\varphi_2(t, x)=x_t$ for all $1\leqq t<n$. It is clear that $\mathbf{I}_n$ is a homomorphic image of the subautomaton generated by $(1, a_1)$ in $\mathbf{I}_{n-1}\times\mathbf{A}(\{x\}, \varphi)$. From this, similarly as above, we get a required decomposition of $\mathbf{I}_n$.

Now suppose that $k=1$. Then one can prove that $a_{n+m-1}\notin(a_{n-1}, x_{n-1})$ and thus $m>1$. Consider the α_0-product $\mathbf{I}_{n-1}\times\mathbf{A}(\{x\}, \varphi)$, where $\varphi_1(x)=x$ and

$$\varphi_2(t, x) = \begin{cases} x_{n-1} & \text{if } t=n-1-um \text{ for some } u=0, 1, \dots, \\ x_{n+m-v} & \text{if } t=n-um-v \text{ for some } 2\leqq v\leqq m \text{ and } u=0, 1, \dots \end{cases}$$

for all $1\leqq t<n$. Let $a=a_{n+m-1}$ if $n=um+2$ for some $u=0, 1, \dots$ and $a=a_{n+m-v}$ if $n=1+um+v$ for some $2\leqq v\leqq m, u=0, 1, \dots$. It is not difficult to see that $\mathbf{I}_n$ is a homomorphic image of the subautomaton generated by $(1, a)$ in $\mathbf{I}_{n-1}\times\mathbf{A}(\{x\}, \varphi)$. This yields a required decomposition of $\mathbf{I}_n$.

It remained to decompose the automata $\mathbf{Q}_n$. Since condition (2) (i) is satisfied by Σ and only this condition was used in the previous decomposition of $\mathbf{Q}_n$ (see the observation made in the proof) the automaton $\mathbf{Q}_n$ can be realized homomorphically by an α_0-product of automata from Σ for any $n\geqq 3$.

Now let us suppose that Σ satisfies condition (3). Since conditions (3) (ii) and (2) (ii) coincide, by the proof of the decomposition of automata $\mathbf{I}_n$ in the case Σ satisfies (2), we have that the automaton $\mathbf{I}_n$ can be realized homomorphically by an α_0-product of automata from Σ for any $n\geqq 2$. Let $n\geqq 3$ be arbitrary and consider the automaton $\mathbf{Q}_n$. Since Σ satisfies (3) (i) there exists an automaton $\mathbf{B}$ in Σ which has n different states b_t $(t=1, \dots, n)$ and input signs x_t $(1\leqq t<n)$ such that $b_t x_t=b_{t+1}$ if $1\leqq t<n-1$ and $b_{n-2}x_{n-1}=b_n$. Take the α_0-product $\mathbf{I}_n\times\mathbf{B}(\{x, y\}, \varphi)$, where $\varphi_1(x)=x$ and

$$\varphi_2(t, x) = \begin{cases} x_t & \text{if } 1\leqq t<n-2, \\ x_{n-2} & \text{otherwise,} \end{cases}$$

$$\varphi_2(t, y) = \begin{cases} x_t & \text{if } 1\leqq t<n-2, \\ x_{n-1} & \text{otherwise,} \end{cases}$$

for all $1 \leqq t \leqq n$. It can be easily seen that $\mathbf{Q}_n$ is a homomorphic image of the subautomaton generated by $(1, b_1)$ in $\mathbf{I}_n \times \mathbf{B}(\{x, y\}, \varphi)$. Therefore, we have a required decomposition of $\mathbf{Q}_n$. This ends the first step of the proof of the sufficiency.

To prove that arbitrary nilpotent automaton can be realized homomorphically by an α_0-product of automata from $\{\mathbf{Q}_m: m \geqq 3\}$, by Theorem 2 and Lemma 2, it is enough to show that the automaton $\mathbf{R}_n$ can be realized homomorphically by an α_0-product of automata from $\{\mathbf{Q}_m: m \geqq 3\}$ for any $n \geqq 3$.

Let $n \geqq 3$ be arbitrary. In order to decompose $\mathbf{R}_n$ consider the automata $\mathbf{R}_n^{(j)}$ $(1 \leqq j < n)$ given by $\mathbf{R}_n^{(j)} = (\{x_1, \ldots, x_{n-1}\}, \{1, \ldots, n\}, \delta_n^{(j)})$, where

$$\delta_n^{(j)}(t, x_s) = \begin{cases} \min(t+1, n) & \text{if } s \neq j, \\ \min(t+j, n) & \text{if } s = j, \end{cases}$$

for any $1 \leqq t \leqq n$ and $x_s \in \{x_1, \ldots, x_{n-1}\}$. Take the direct product $\prod_{j=1}^{n-1} \mathbf{R}_n^{(j)}$ and let $\mathbf{W}$ denote its subautomaton generated by $(1, \ldots, 1)$. Observe that $a_i \geqq a_1$ holds if $1 \leqq i \leqq n-1$ for any state $(a_1, \ldots, a_{n-1})$ of the subautomaton $\mathbf{W}$. Therefore, if $a_s = k$ holds for some $1 \leqq s \leqq n-1$ and $1 \leqq k \leqq n$ then $a_1 + \sum_{i=2}^{n-1} (a_i - a_1) \geqq k$. Now define the mapping μ: $W \to \{1, \ldots, n\}$ in the following way:

$$\mu(a_1, \ldots, a_{n-1}) = \min\left(a_1 + \sum_{i=2}^{n-1} (a_i - a_1), n\right).$$

By the observation above, it is not difficult to prove that the mapping μ is a homomorphism of $\mathbf{W}$ onto $\mathbf{R}_n$.

Now let $1 \leqq j < n$ be arbitrary. For the decomposition of $\mathbf{R}_n^{(j)}$ consider the automaton $\mathbf{R}_{n,k}^{(j)} = (\{x_1, \ldots, x_{n-1}\}, \{1, \ldots, n\}, \delta_{n,k}^{(j)})$ for all k $(1 \leqq k \leqq n-1)$, where

$$\delta_{n,k}^{(j)}(t, x_s) = \begin{cases} \min(t+s, n) & \text{if } t = k \text{ and } j = s, \\ \min(t+1, n) & \text{otherwise}, \end{cases}$$

for any $1 \leqq t \leqq n$ and $x_s \in \{x_1, \ldots, x_{n-1}\}$. Take the direct product $\prod_{k=1}^{n-1} \mathbf{R}_{n,k}^{(j)}$ and denote by $\mathbf{U}$ its subautomaton generated by $(1, \ldots, 1)$. Observe that for any state $(a_1, \ldots, a_{n-1}) \in U$ $0 \leqq a_i - a_{n-1} \leqq j-1$ holds provided $1 \leqq i \leqq n-1$ and $a_t = r$ $(t = r, \ldots, n-2)$ if $a_{n-1} = r$ for some r, where $1 \leqq r < n-1$. Now define the mapping μ: $U \to \{1, \ldots, n\}$ in the following way:

$$\mu(a_1, \ldots, a_{n-1}) = \min\left(a_{n-1} + \sum_{i=1}^{n-2} (a_i - a_{n-1}), n\right).$$

By the observation above, it can be seen that the mapping μ is a homomorphism of $\mathbf{U}$ onto $\mathbf{R}_n^{(j)}$.

Now let $1 \leqq k \leqq n-1$ be arbitrary. If $j = 1$ or $n-2 \leqq k$ then $\mathbf{R}_{n,k}^{(j)}$ can be embedded isomorphically into an α_0-product of $\mathbf{Q}_n$ with a single factor. Let us suppose that $1<j$ and $1\leqq k<n-2$. For the decomposition of $\mathbf{R}^{(j)}_{n,k}$ consider the α_0-product $\mathbf{A}=\prod_{i=k+2}^{n}\mathbf{Q}_i(\{x_1,\dots,x_{n-1}\},\varphi)$, where

$$\varphi_1(x_s)=\begin{cases}x & \text{if } s\neq j,\\ y & \text{if } s=j,\end{cases}\quad \varphi_2(a_1,x_s)=\begin{cases}y & \text{if } a_1=k+2,\\ x & \text{otherwise},\end{cases}$$

$$\varphi_{i+1}(a_1,\dots,a_i,x_s)=\begin{cases}y & \text{if } a_1<a_2<\dots<a_i,\\ x & \text{otherwise},\end{cases}$$

for any $x_s\in\{x_1,\dots,x_{n-1}\}$, $2\leqq i\leqq n-k-2$, $a_t\in\{1,\dots,t+k+1\}$ $(1\leqq t\leqq n-k-2)$. Let $v=n-k-1$ and take the following sets of states of $\mathbf{A}$:

$A_1=\{(a_1,\dots,a_v)\colon a_1\leqq k+2 \text{ and } a_i=a_{i+1}\ (i=1,\dots,v-1)\}$,

$A_2=\{(a_1,\dots,a_v)\colon a_1=k+2 \text{ and } (\exists s)\ (2\leqq s\leqq v \text{ and } a_i<a_{i+1} \text{ if } i\leqq s-1 \text{ and } a_i=a_{i+1} \text{ if } s\leqq i<v)\}$,

$A_3=\{(a_1,\dots,a_v)\colon a_1=k+2 \text{ and } (\exists s)\ (1\leqq s<v \text{ and } a_i<a_{i+1} \text{ if } 1\leqq i\leqq s-1 \text{ and } a_i=a_s-1 \text{ if } s<i\leqq v)\}$.

It can be shown, by a sort computation, that $\mathbf{B}=(\{x_1,\dots,x_{n-1}\},\bigcup_{i=1}^{3}A_i,\delta|_{\left(\bigcup_{i=1}^{3}A_i\right)\times\{x_1,\dots,x_{n-1}\}})$ is a subautomaton of $\mathbf{A}$. Now define the mapping $\mu\colon\bigcup_{i=1}^{3}A_i\to\{1,\dots,n\}$ in the following way:

$$\mu(a_1,\dots,a_v)=\begin{cases}\max\limits_{1\leqq i\leqq v} a_i & \text{if } (a_1,\dots,a_v)\in A_1\cup A_2,\\ \min(a_v+j-1,n) & \text{if } (a_1,\dots,a_v)\in A_3.\end{cases}$$

It is not difficult to prove that the mapping μ is a homomorphism of $\mathbf{B}$ onto $\mathbf{R}^{(j)}_{n,k}$. This ends the proof of Theorem 4.

The following Theorem holds for α_i-products with $i\geqq 1$.

Theorem 5. A system Σ of automata is homomorphically complete for the class of all nilpotent automata with respect to the α_i-product $(i\geqq 1)$ if and only if one of the following two conditions is satisfied by Σ:

(1) Σ contains an automaton which has a state b and input signs $x_1,\dots,x_k,y$ such that $bx_1\dots x_k=b$ and $bx_1\neq by$,

(2) (i) for any natural number $n\geqq 3$ there exists an automaton in Σ which has n different states b_t $(t=1,\dots,n)$ and input signs x_t $(1\leqq t<n)$ such that $b_tx_t=b_{t+1}$ if $1\leqq t<n-1$ and $b_{n-2}x_{n-1}=b_n$,

(2) (ii) for any $n\geqq 3$ there exist $m\geqq 1$ and an automaton in Σ such that it has $n+m-1$ different states a_t $(t=1,\dots,n+m-1)$ and input signs x_t $(1\leqq t\leqq n+m-1)$ for which $a_tx_t=a_{t+1}$ $(1\leqq t<n+m-1)$ and $a_{n+m-1}x_{n+m-1}=a_n$ hold.

Proof. The necessity can be proved in a similar way as in the proof of Theorem 4. (One need consider the homomorphic realization of $\mathbf{Q}_n$.)

In order to verify the sufficiency assume that Σ satisfies (1) by $\mathbf{A}=(X, A, \delta)$. From the proof of Theorem 4 it follows that every nilpotent automaton can be realized homomorphically by an α_0-product of automata from $\{\mathbf{A}\}\cup\{\mathbf{I}_m\colon m\geqq 2\}$. Therefore, using the fact that the α_0-product of α_1-products is an α_1-product, it is enough to show that the automaton $\mathbf{I}_n$ can be realized homomorphically by an α_1-power of $\mathbf{A}$ for any $n\geqq 2$. Indeed, let $n\geqq 2$ be arbitrary. Write $b_1=b$ and $b_{t+1}=b_t x_t$ $(t=1, \ldots, k-1)$. Without loss of generality we may assume that the states $b_1, \ldots, b_k$ are pairwise different. We distinguish three cases.

First suppose that $\{b_1 y, b_1 y^2, \ldots\}\cap\{b_1, \ldots, b_k\}=\emptyset$. Then take the α_1-power $\mathbf{A}^{n-1}(\{x\}, \varphi)$, where $\varphi_1(u_1, x)=y$,

$$\varphi_t(u_1, \ldots, u_t, x)=\begin{cases} y & \text{if } \{u_1, \ldots, u_{t-1}\}\cap\{b_1, \ldots, b_k\}=\emptyset, \\ x_j & \text{if } \{u_1, \ldots, u_{t-1}\}\cap\{b_1, \ldots, b_k\}\neq\emptyset \text{ and } u_t=b_j, \\ \text{arbitrary input sign from } X \text{ otherwise,} \end{cases}$$

for any state $(u_1, \ldots, u_{n-1})\in A^{n-1}$ and $2\leqq t\leqq n-1$. Define the state $(a_1, \ldots, a_{n-1})$ of the α_1-product in the following way:

$$a_1=b_1,\ a_{t+1}=\begin{cases} b_{j-1} & \text{if } a_t=b_j \text{ for some } 1<j\leqq k, \\ b_k & \text{if } a_t=b_1 \end{cases}$$

where $t=1, \ldots, n-2$. Let $\mathbf{U}$ denote the subautomaton of $\mathbf{A}^{n-1}(\{x\}, \varphi)$ which is generated by $(a_1, \ldots, a_{n-1})$. It is not difficult to see that $\mathbf{I}_n$ is a homomorphic image of $\mathbf{U}$ and thus, we got a required decomposition of $\mathbf{I}_n$.

Now assume that $\{b_1 y, b_1 y^2, \ldots\}\cap\{b_1, \ldots, b_k\}\neq\emptyset$ and $b_1 y\notin\{b_1, \ldots, b_k\}$. Denote by $s>1$ the least natural number for which $b_1 y^s\in\{b_1, \ldots, b_k\}$. There exists such an s. Take the α_1-power $\mathbf{A}^{n-1}(\{x\}, \varphi)$, where

$$\varphi_1(u_1, x)=\begin{cases} y & \text{if } u_1\in\{b_1 y_1, \ldots, b_1 y^{s-1}\}, \\ x_j & \text{if } u_1=b_j \text{ for some } 1\leqq j\leqq k, \\ \text{arbitrary input sign from } X \text{ otherwise,} \end{cases}$$

$$\varphi_t(u_1, \ldots, u_t, x)=\begin{cases} y & \text{if } u_t\in\{b_1 y, \ldots, b_1 y^{s-1}\}, \\ y & \text{if } u_t=b_1 \text{ and } u_{t-1}\in\{b_1 y, b_1 y^2, \ldots\}, \\ x_j & \text{if } u_t=b_j \text{ for some } 1<j\leqq k, \\ x_1 & \text{if } u_t=b_1 \text{ and } u_{t-1}\notin\{b_1 y, b_1 y^2, \ldots\}, \\ \text{arbitrary input sign from } X \text{ otherwise,} \end{cases}$$

for any $(u_1, \ldots, u_{n-1})\in A^{n-1}$, $2\leqq t\leqq n-1$. Define the state $(a_1, \ldots, a_{n-1})\in A^{n-1}$ in the following way:

$$a_1=b_1 y^{s-1}, \ldots, a_{s-1}=b_1 y, \quad a_s=b_1 \text{ and}$$

$$a_{t+1}=\begin{cases} b_{j-1} & \text{if } a_t=b_j, \text{ for some } 1<j\leqq k, \\ b_k & \text{if } a_t=b_1, \end{cases}$$

where $s\leqq t<n-1$. Denote by $\mathbf{U}$ the subautomaton generated by $(a_1, \ldots, a_{n-1})$. It can be seen easily that $\mathbf{I}_n$ is a homomorphic image of $\mathbf{U}$, which yields a required decomposition of $\mathbf{I}_n$.

Finally, assume that $b_1 y \in \{b_1, \ldots, b_k\}$. Then $k \geqq 2$ and $b_1 y = b_i$ for some $i \neq 2$, $1 \leqq i \leqq k$. Let $D = \{b_2, \ldots, b_{i-1}\}$ if $i \neq 1$ and $D = \{b_2, \ldots, b_k\}$ if $i = 1$. Consider the α_1-power $\mathbf{A}^{n-1}(\{x\}, \varphi)$, where

$$\varphi_1(u_1, x) = \begin{cases} y & \text{if} \quad u_1 = b_1, \\ x_j & \text{if} \quad u_1 = b_j \quad \text{for some} \quad 2 \leqq j \leqq k, \\ \text{arbitrary input sign from } X \text{ otherwise,} \end{cases}$$

$$\varphi_t(u_1, \ldots, u_t, x) = \begin{cases} y & \text{if} \quad u_t = b_1 \quad \text{and} \quad \{u_1, \ldots, u_{t-1}\} \cap D = \emptyset, \\ x_j & \text{if} \quad u_t = b_j \quad \text{for some} \quad 2 \leqq j \leqq k, \\ x_1 & \text{if} \quad u_t = b_1 \quad \text{and} \quad \{u_1, \ldots, u_{t-1}\} \cap D \neq \emptyset, \\ \text{arbitrary input sign from } X \text{ otherwise,} \end{cases}$$

for any $(u_1, \ldots, u_{n-1}) \in A^{n-1}$, $2 \leqq t \leqq n-1$. Let b_r denote that element of D which has the greatest index. Define the state $(a_1, \ldots, a_{n-1}) \in A^{n-1}$ in the following way:

$$a_1 = b_r, \quad a_2 = b_{r-1}, \ldots, a_r = b_1 \text{ and}$$

$$a_{t+1} = \begin{cases} b_{j-1} & \text{if} \quad a_t = b_j \quad \text{for some} \quad 1 < j \leqq k, \\ b_k & \text{if} \quad a_t = b_1, \end{cases}$$

where $r \leqq t < n-1$. Denote by $\mathbf{U}$ the subautomaton generated by $(a_1, \ldots, a_{n-1})$. It is not difficult to prove that $\mathbf{I}_n$ is a homomorphic image of $\mathbf{U}$ and thus, we have a required decomposition of $\mathbf{I}_n$.

It remained to prove the sufficiency of condition (2). But this can be seen easily, using Theorem 4 and the fact that the α_0-product of α_1-products is an α_1-product. This ends the proof of Theorem 5.

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

References

[1] ÉSIK, Z. and B. IMREH, Subdirectly irreducible commutative automata, *Acta Cybernet.*, to appear.

[2] GÉCSEG, F., Composition of automata, Proceedings of the 2nd Colloquium on Automata, Languages and Programming, Saarbrücken, Springer Lecture Notes in Computer Science v. 14, 1974, pp. 351—363.

[3] IMREH, B., On isomorphic representations of commutative automata with respect to the α_i-products, *Acta Cybernet.*, v. 5, 1980, pp. 21—32.

(Received Jan. 19, 1981)

On the functional dependency and some generalizations of it

By J. DEMETROVICS and GY. GYEPESI

§ 0. Introduction

According to E. F. CODD [6] a relation is a matrix without two identical rows. Rows correspond to data records and columns to the attributes that are to be stored of a data item. He also introduced [7] the concept of *functional dependency:* a set of columns depends on another if fixing the values in a row taken on the first determines those on the second.

Other concepts of his are the *key* (a set of attributes on which all depend) and the *candidate key* (a minimal key).

Candidate keys clearly do not contain each other [10].

The possible mathematical structure of functional dependencies was first investigated by W. W. ARMSTRONG [1]. Among others he found that this structure is determined by the *maximal dependencies* (those which have maximal attribute subsets depending on minimal ones) and even by the dependent sides of the maximal dependencies. We also heavily use these "maximal dependent subsets of attributes" as technical tools.

Different kinds of functional dependency have also been introduced [3], [11], [13], [14], and axiomatized, usually in systems similar to those investigated by Armstrong.

The harder problems of the topic are usually of combinatorial nature (see [4], [5], [9], [15]).

In this paper in § 1 we give the formal definition of the functional, dual, strong and weak dependencies and give new axioms for full *f*- *d*- and *s*-families.

In § 2 we show the analogy and differences among the dependencies of different types and give an axiom for full *w*-families.

In § 3 we deal with a question stated in [9].

Before starting § 1 we make some remarks concerning the practice:

The functional, dual, strong and weak dependencies studied in this paper are those restrictions which allow the characterization of a relation by restrictions of it to certain proper subsets of the attribute set.

Certain dependencies of a relational data base are known by its designer. We call these *initial dependencies.* In general initial dependencies imply new dependencies. W. W. ARMSTRONG [2] has developed a method to find the dependencies implied by a given set of initial functional dependencies. He also gave a characterization of the sets of initial dependencies that imply all the dependencies of a given full f-family and are of minimal cardinality. This characterization has a logical nature; we give a combinatorial equivalent of it.

We use the following notational conventions: Ω denotes the set of attributes, $P(\Omega)$ denotes its power set. If g is a function with X as its domain and $Z \subseteq X$ then $g \upharpoonright Z$ denotes the function which has domain Z and for any $z \in Z$ $g(z) = g \upharpoonright Z(z)$. $\subset$ means strict inclusion.

§ 1. Old and new axioms

We start with the definitions of functional, dual, strong and weak dependencies based on [1] and [8].

Definition 1.1. Let A, B be subsets of Ω and let R be a relation over Ω. Then we say that B

(i) *functionally;*
(ii) *dually;*
(iii) *strongly;*
(iv) *weakly*

depends on A in R if

(i) $(\forall g, h \in R)(g \upharpoonright A = h \upharpoonright A \Rightarrow g \upharpoonright B = h \upharpoonright B)$;

(ii) $(\forall g, h \in R)((\exists a \in A)(g(a) = h(a)) \Rightarrow (\exists b \in B)(g(b) = h(b)))$;

(iii) $(\forall g, h \in R)((\exists a \in A)(g(a) = h(a)) \Rightarrow g \upharpoonright B = h \upharpoonright B)$;

(iv) $(\forall g, h \in R)(g \upharpoonright A = h \upharpoonright A \Rightarrow (\exists b \in B)(g(b) = h(b)))$;

holds respectively and denote these by $A \xrightarrow[R]{f} B$, $A \xrightarrow[R]{d} B$, $A \xrightarrow[R]{s} B$, $A \xrightarrow[R]{w} B$ corresponding to the type of the denoted dependency.

The following example [8] illustrates the effect of the dual dependency.

EXAMPLE. Let $\Omega = \{\text{author, title, hall, shelf}\}$. Let we have a library with eighteen books, three halls and three shelves in every hall; one shelf holds two books. Let the relation R containing the data of the library given by the following table:

author	*title*	*hall*	*shelf*	*author*	*title*	*hall*	*shelf*
1	1	1	2	10	10	3	2
2	2	1	3	11	11	3	3
3	3	1	1	12	12	3	1
4	4	1	2	1	4	1	1
5	5	2	3	5	8	3	3
6	6	2	1	4	1	1	3
7	7	2	2	7	10	3	2
8	8	2	3	6	10	2	2
9	9	3	1	6	9	2	1

Thus {author, title} $\overset{d}{\underset{R}{\rightarrow}}$ {hall, shelf} holds, and for $i=1, \ldots, 12$ the book by author i and entitled i is on the $\left(1+3\cdot\left\{\frac{i}{3}\right\}\right)$-th shelf of the $\left[\frac{i+3}{4}\right]$-th hall ($[x]$ denotes the integer part and $\{x\}$ the fraction part of x). The reader, knowing the author or the title of the required book, may find it without examining the whole library: for example if i is the author of the book, then it is enough to look the $\left[\frac{i+3}{4}\right]$-th hall, and the $\left(1+3\cdot\left\{\frac{i}{3}\right\}\right)$-th shelves of the other two halls.

In R {author, title} $\overset{f}{\underset{R}{\rightarrow}}$ {hall, shelf} holds too, but to store this functional dependency is equivalent to store the table of R; the {author, title} $\overset{d}{\underset{R}{\rightarrow}}$ {hall, shelf} dependency is more effective.

If R is a relation over Ω, $\mathcal{Y}\in\{\mathcal{F}, \mathcal{D}, \mathcal{S}, \mathcal{W}\}$ and $y\in\{f, d, s, w\}$ corresponds to $\mathcal{Y}$, then we write

$$\mathcal{Y}_R = \{(A, B)\colon A \overset{y}{\underset{R}{\rightarrow}} B\}.$$

We call the sets which have the form $\mathcal{Y}_R$ *full y-families*, where y corresponds to $\mathcal{Y}$.

In order to investigate the various dependencies the first step is the axiomatization of full y-families for $y\in\{f, d, s, w\}$. In [1] there is a system of axioms for full f-family and in [8] there are for full d- and s-families. For the sake of completeness we reproduce them here.

Let $\mathcal{Y}\subseteq P(\Omega)\times P(\Omega)$. Then we say that $\mathcal{Y}$ satisfies the $\mathfrak{f}$-axioms, if for all $A, B, C, D\subseteqq\Omega$

(F1) $(A, A)\in\mathcal{Y}$;

(F2) $(A, B)\in\mathcal{Y}$, $(B, C)\in\mathcal{Y} \Rightarrow (A, C)\in\mathcal{Y}$;

(F3) $(A, B)\in\mathcal{Y}$, $C \supseteqq A$, $D \subseteqq B \Rightarrow (C, D)\in\mathcal{Y}$;

(F4) $(A, B)\in\mathcal{Y}$, $(C, D)\in\mathcal{Y} \Rightarrow (A\cup C, B\cup D)\in\mathcal{Y}$.

$\mathcal{Y}$ satisfies the ϑ-axioms if for all $A, B, C, D\subseteqq\Omega$

(D1) $(A, A)\in\mathcal{Y}$;

(D2) $(A, B)\in\mathcal{Y}$, $(B, C)\in\mathcal{Y} \Rightarrow (A, C)\in\mathcal{Y}$;

(D3) $(A, B)\in\mathcal{Y}$, $C \subseteqq A$, $B \subseteqq D \Rightarrow (C, D)\in\mathcal{Y}$;

(D4) $(A, B)\in\mathcal{Y}$, $(C, D)\in\mathcal{Y} \Rightarrow (A\cup C, B\cup D)\in\mathcal{Y}$;

(D5) $(A, \emptyset)\in\mathcal{Y} \Rightarrow A = \emptyset$.

$\mathcal{Y}$ satisfies the γ-axioms if for all $A, B, C, D\subseteqq\Omega$ and for any $a\in\Omega$

(S1) $(\{a\}, \{a\})\in\mathcal{Y}$;

(S2) $(A, B)\in\mathcal{Y}$, $(B, C)\in\mathcal{Y}$, $B\neq\emptyset \Rightarrow (A, C)\in\mathcal{Y}$;

(S3) $(A, B)\in\mathcal{Y}$, $C \subseteqq A$, $D \subseteqq B \Rightarrow (C, D)\in\mathcal{Y}$;

(S4) $(A, B)\in\mathscr{Y},\quad (C, D)\in\mathscr{Y}\Rightarrow(A\cap C, B\cup D)\in\mathscr{Y};$

(S5) $(A, B)\in\mathscr{Y},\quad (C, D)\in\mathscr{Y}\Rightarrow(A\cup C, B\cap D)\in\mathscr{Y}.$

We need the following technical lemma.

Lemma 1.1. Let $\mathscr{F}\subseteq P(\Omega)\times P(\Omega)$ be such that $(X, Y)\in\mathscr{F}$ and $Y\neq\emptyset$ imply $X\neq\emptyset$. Then $\mathscr{F}$ satisfies the $\mathfrak{f}$-axioms iff $\mathscr{D}=\{(A, B)\colon (B, A)\in\mathscr{F}\}$ satisfies the ϑ-axioms.

Proof. Trivial by the $\mathfrak{f}$- and ϑ-axioms. (D5) makes necessary the assumption that $(X, Y)\in\mathscr{F}$ and $Y\neq\emptyset$ imply $X\neq\emptyset$. □

REMARK. The assumption $((X, Y)\in\mathscr{F}$ and $Y\neq\emptyset$ imply $X\neq\emptyset)$ in Lemma 1.1 is not an important restriction: if $\mathscr{F}$ satisfies the $\mathfrak{f}$-axioms let $\mathscr{F}'=\mathscr{F}\setminus\{(\emptyset, X)\colon X\neq\emptyset\}$. Then $\mathscr{F}'$ obviously satisfies the $\mathfrak{f}$-axioms and the critical assumption as well, and we have that $X\neq\emptyset$ implies $(X, Y)\in\mathscr{F}\Leftrightarrow(X, Y)\in\mathscr{F}'$.

In the following we give new axioms instead of the $\mathfrak{f}$- ϑ- and γ-axioms and give an axiom that characterizes the *weak full w-families* which is such a full w-family that whenever (X, Y) is an element of the family then X is not void.

F-axiom. Let $\mathscr{F}\subseteq P(\Omega)\times P(\Omega)$. Then we say that *$\mathscr{F}$ satisfies the F-axiom* if for any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$ there is an $E\subseteq\Omega$ such that

(i) $X\subseteq E$ and $Y\nsubseteq E$;
(ii) if $(X', Y')\in\mathscr{F}$ and $X'\subseteq E$ then $Y'\subseteq E$.

D-axiom. Let $\mathscr{D}\subseteq P(\Omega)\times P(\Omega)$. Then we say that *$\mathscr{D}$ satisfies the D-axiom* if for any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathscr{D}$ there is an $E\subseteq\Omega$ such that

(i) $X\cap E\neq\emptyset$ and $Y\cap E=\emptyset$;
(ii) if $(X', Y')\in\mathscr{D}$ and $X'\cap E\neq\emptyset$ then $Y'\cap E\neq\emptyset$.

S-axiom. Let $\mathscr{S}\subseteq P(\Omega)\times P(\Omega)$. Then we say that *$\mathscr{S}$ satisfies the S-axiom* if for any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathscr{S}$ there is an $E\subseteq\Omega$ such that

(i) $X\cap E\neq\emptyset$ and $Y\nsubseteq E$
(ii) if $(X', Y')\in\mathscr{S}$ and $X'\cap E\neq\emptyset$ then $Y'\subseteq E$.

W-axiom. Let $\mathscr{W}\subseteq P(\Omega)\times P(\Omega)$. Then we say that *$\mathscr{W}$ satisfies the W-axiom* if for any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathscr{W}$ there is an $E\subseteq\Omega$ such that

(i) $X\subseteq E$ and $Y\cap E=\emptyset$;
(ii) if $(X', Y')\in\mathscr{W}$ and $X'\subseteq E$ then $Y'\cap E\neq\emptyset$.

Theorem 1.1. (i) Let $\mathscr{F}\subseteq P(\Omega)\times P(\Omega)$. Then $\mathscr{F}$ satisfies the $\mathfrak{f}$-axioms iff $\mathscr{F}$ satisfies the F-axiom.

(ii) Let $\mathscr{D}\subseteq P(\Omega)\times P(\Omega)$. Then $\mathscr{D}$ satisfies the ϑ-axioms iff $\mathscr{D}$ satisfies the D-axiom.

(iii) Let $\mathscr{S}\subseteq P(\Omega)\times P(\Omega)$. Then $\mathscr{S}$ satisfies the γ-axioms iff $\mathscr{S}$ satisfies the S-axiom.

Proof. (i) Suppose that $\mathscr{F}$ satisfies the F-axiom. Then

(F1) If $(A, A)\notin\mathscr{F}$ then there is an $E\subseteq\Omega$ such that $A\subseteq E$ and $A\nsubseteq E$ which is a contradiction.

(F2) If $(A, B)\in\mathscr{F}$, $(B, C)\in\mathscr{F}$ and $(A, C)\notin\mathscr{F}$ then there is an $E\subseteq\Omega$ such that $A\subseteq E$ and $C\nsubseteq E$. Furthermore $(A, B)\in\mathscr{F}$, $A\subseteq E$ imply $B\subseteq E$, and using $(B, C)\in\mathscr{F}$, $C\subseteq E$ which is a contradiction.

(F3) If $(A, B)\in\mathscr{F}$, $A'\supseteq A$, $B'\subseteq B$ and $(A', B')\notin\mathscr{F}$ then there is an $E\subseteq\Omega$ such that $A'\subseteq E$ and $B'\nsubseteq E$ and $(A, B)\in\mathscr{F}$, $A\subseteq E$ imply that $B\subseteq E$. Thus, by $B'\subseteq B$, $B'\subseteq E$ which is again a contradiction.

(F4) If $(A, B)\in\mathscr{F}$, $(C, D)\in\mathscr{F}$ and $(A\cup C, B\cup D)\notin\mathscr{F}$ then there is an $E\subseteq\Omega$ such that $A\cup C\subseteq E$ and $B\cup D\nsubseteq E$; e.g. $B\nsubseteq E$. But $(A, B)\in\mathscr{F}$ and $A\subseteq E$ imply that $B\subseteq E$, which is a contradiction.

Suppose now that $\mathscr{F}$ satisfies the $\mathfrak{f}$-axioms. Let $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$.

Claim. There is an $E\supseteq A$ such that $(E, B)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$ and $E'\supset E$ implies $(E', B)\in\mathscr{F}$.

$(\Omega, \Omega)\in\mathscr{F}$ by (F1). Thus, by (F3), $(\Omega, B)\in\mathscr{F}$ holds. $A\subseteq\Omega$ and $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$, consequently there is an $E\subseteq\Omega$ which is maximal w.r. to the properties $(E, B)\in\mathscr{F}$ and $E\supseteq A$.

This E clearly satisfies the restrictions of the Claim.

Let $E\supseteq A$ which is guaranteed by the Claim. We state that E satisfies (i) and (ii) of the F-axiom. Namely, by the choice of E, $A\subseteq E$ holds. By (F1) and (F3), $B\subseteq E$ implies $(E, B)\in\mathscr{F}$. Thus we have $B\nsubseteq E$.

Let $(C, D)\in\mathscr{F}$ and $C\subseteq E$. $D\nsubseteq E$ implies $E'=D\cup E\supset E$, and, by the maximality of E, $(E', B)\in\mathscr{F}$ holds. Since $(E, D)\in\mathscr{F}$, by (F3), and $(E, E)\in\mathscr{F}$ by (F1), we have $(E, E')\in\mathscr{F}$. Now $(E, E')\in\mathscr{F}$, and $(E', B)\in\mathscr{F}$ and (F2) imply that $(E, B)\in\mathscr{F}$, which is a contradiction.

(ii) Let $\mathscr{F}=\{(A, B): (B, A)\in\mathscr{D}\}$. Then, by Lemma 1.1, $\mathscr{F}$ satisfies the $\mathfrak{f}$-axioms iff $\mathscr{D}$ satisfies the ϑ-axioms. Hence, by (i), it is enough to show that $\mathscr{F}$ satisfies the F-axiom iff $\mathscr{D}$ satisfies the D-axiom.

Suppose that $\mathscr{F}$ satisfies the F-axiom. For $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$ let $E(A, B)$ be a subset of Ω such that $A\subseteq E(A, B)$, $B\nsubseteq E(A, B)$ and if both $(A', B')\in\mathscr{F}$ and $A'\subseteq E(A, B)$ hold, then $B'\subseteq E(A, B)$. By the F-axiom such an $E(A, B)$ exists. By the definition of $\mathscr{F}$ whenever $(A, B)\in P(\Omega)\times P(\Omega)$ then $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathscr{F}$ iff $(B, A)\in P(\Omega)\times P(\Omega)\setminus\mathscr{D}$.

Now it is easy to check that for $(B, A)\in P(\Omega)\times P(\Omega)\setminus\mathscr{D}$, $\Omega\setminus E(A, B)$ satisfies the D-axiom.

If $\mathscr{D}$ satisfies the D-axiom, then $\mathscr{F}$ satisfies the F-axiom; this can be shown by the same argument.

(iii) Suppose that $\mathscr{S}$ satisfies the S-axiom. Then the proof of the fact that $\mathscr{S}$ satisfies the γ-axioms is an easy modification of the proof of (i). We deal with but (S1) and (S2).

(S1) if $(\{a\}, \{a\})\notin\mathscr{S}$, then there is an $E\subseteq\Omega$ such that $\{a\}\cap E\neq\emptyset$ and $\{a\}\cap(\Omega-E)\neq\emptyset$ which contradicts $|\{a\}|=1$.

(S2) if $(A, B)\in\mathscr{S}$, $(B, C)\in\mathscr{S}$, $B\neq\emptyset$ and $(A, C)\notin\mathscr{S}$ then there is an $E\subseteq\Omega$ such that

(a) $A\cap E\neq\emptyset$

(b) $C\cap(\Omega\setminus E)\neq\emptyset$ and

(c) $(F, D)\in\mathscr{S}$, $F\cap E\neq\emptyset$ imply that $D\subseteq E$.

(a) and (c) imply $B\subseteq E$. By $B\neq\emptyset$ we have $B\cap E\neq\emptyset$. Hence, by $(B, C)\in\mathscr{S}$, $C\subseteq E$ holds which is a contradiction.

Suppose now that $\mathcal{S}$ satisfies the γ-axioms. Let $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathcal{S}$

Claim. There is an $a\in A$ and an $E\subseteq\Omega$ such that
(a) $a\in E$;
(b) $(\{a\}, E)\in\mathcal{S}$ and
(c) $E'\supset E$ implies that $(\{a\}, E')\notin\mathcal{S}$.

If for any $a\in A$ we have $(\{a\}, B)\in\mathcal{S}$ then $(A, B)\in\mathcal{S}$ by the repeated application of (S5).[1] Hence there is an $a\in A$ such that $(\{a\}, B)\notin\mathcal{S}$. Now if for every $b\in B$ $(\{a\}, \{b\})\in\mathcal{S}$ holds then by the repeated application of (S4) we have $(\{a\}, B)\in\mathcal{S}$. Thus there is a $b\in B$ such that $(\{a\}, \{b\})\notin\mathcal{S}$.

By (S1) and (S3) there is an $E\subseteq\Omega$ such that $a\in E$, $(\{a\}, E)\in\mathcal{S}$ and E is maximal w.r. to this property. This E is appropriate for the Claim.

Let $E\subseteq\Omega$ and $a\in A$ guaranteed by the Claim. Then by (S3) we have $b\notin E$. Hence $A\cap E\neq\emptyset$ and $B\cap(\Omega\setminus E)\neq\emptyset$. Now let $(C, D)\in\mathcal{S}$ such that $C\cap E\neq\emptyset$; let $c\in C\cap E$. Suppose that $D\cap(\Omega\setminus E)\neq\emptyset$; let $d\in D\cap(\Omega\setminus E)$. By (S3) we have $(\{c\}, \{d\})\in\mathcal{S}$ and by (S1) we have $(\{c\}, \{c\})\in\mathcal{S}$. $(\{a\}, E)\in\mathcal{S}$ implies that $(\{a, c\}, \{c\})\in\mathcal{S}$, by (S5). Hence (S3) implies that $(\{a\}, \{c\})\in\mathcal{S}$. Now $(\{a\}, \{c\})\in\mathcal{S}$, $(\{c\}, \{d\})\in\mathcal{S}$ and (S2) imply that $(\{a\}, \{d\})\in\mathcal{S}$. Thus by (S4) we have $(\{a\}, E\cup\{d\})\in\mathcal{S}$ which is a contradiction as $E'=E\cup\{d\}\supset E$.

Consequently the E guaranteed by the Claim demonstrates that $\mathcal{S}$ satisfies the S-axiom. □

§ 2. The equality-set

Definition 2.1. Let R be a relation over Ω. We define *the equality-set of R, $\mathcal{E}_R$* as follows: For $h, g\in R$ let $E(h,g)=\{a\in\Omega\colon h(a)=g(a)\}$ and let $\mathcal{E}_R= =\{E(h, g)\colon h, g\in R$ and $h\neq g\}$.

Definition 2.2. Let $\mathcal{A}$ be a-set system. Then *$\mathcal{A}$ is a Δ-system* if for any $A, B, C, D\in\mathcal{A}$, $A\neq B$ and $C\neq D$ implies that $A\cap B=C\cap D$.

Remark. It is easy to see that $\mathcal{A}$ is a Δ-system iff for any $A, B\in\mathcal{A}$, $A\neq B$ implies that $A\cap B=\bigcap\mathcal{A}$.

Theorem 2.1 (i) Let R be a relation over Ω and let h, f, g different elements of R. Then $E(h, g)$, $E(h, f)$, $E(g, f)$ form a Δ-system.

(ii) Let $\mathcal{E}=\{E_{i,j}\colon 1\leqq i<j\leqq k\}$ such that for each $1\leqq i<j<l\leqq k$, $\{E_{i,j}, E_{i,l}, E_{j,l}\}$ is a Δ-system. Then there is a relation R over Ω with $\mathcal{E}_R=\mathcal{E}$.

Proof. By symmetry it is enough to prove that $a\in E(h,g)\cap E(h,f)$ implies $a\in E(g,f)$. But $a\in E(h,g)\cap E(h,f)$ means that $g(a)=h(a)=f(a)$, hence $a\in E(g f)$.

We construct by induction the rows $h_1, \dots, h_k$ of R. Let $h_1(a)=0$ for each $a\in\Omega$, and assume that $n<k$ and the rows $h_1, \dots, h_n$ have been defined s.t. for each $1\leqq i<j\leqq n$, $E(h_i, h_j)=E_{i,j}$ holds. We construct h_{n+1} as follows:

$$h_{n+1}(a)=\begin{cases} h_i(a), & \text{if } a\in E_{i,n+1} \text{ for some } 1\leqq i\leqq n;\\ \max\big(h_i(b)\colon b\in\Omega \;\&\; 1\leqq i\leqq n\big)+1 & \text{else.}\end{cases}$$

[1] $A\neq\emptyset$, since $\forall b\in B$ $(\{b\},\{b\})\in\mathcal{S}$ by (S1), hence $(B, B)\in\mathcal{S}$ by (S4) and (S5). But if $A=\emptyset$, then $(A, B)\in\mathcal{S}$ by (S3).

Then

(a) h_{n+1} is well-defined.

To prove this we have to show that $a \in E_{i,n+1} \cap E_{j,n+1}$ implies $h_i(a)=h_j(a)$. But this is obvious because $E_{i,j}$, $E_{i,n+1}$, $E_{j,n+1}$ form a Δ-system and the induction hypothesis holds for $i, j \leqq n$.

(b) if $1 \leqq i \leqq n$ and $a \notin E_{i,n+1}$ then $h_i(a) \neq h_{n+1}(a)$.

Suppose first that $a \in E_{j,n+1}$ for some $1 \leqq j < n+1$. Then, by (a) and the definition of h_{n+1}, $h_{n+1}(a)=h_j(a)$ holds. Furthermore $a \notin E_{i,j}$ because $\{E_{i,j}, E_{j,n+1}, E_{i,n+1}\}$ is a Δ-system. Thus the induction hypothesis implies $h_i(a) \neq \neq h_j(a)$, that is $h_i(a) \neq h_{n+1}(a)$.

If $a \notin \bigcup_{1 \leqq j \leqq n} E_{j,n+1}$ then we have $h_{n+1}(a) \neq h_i(a)$ by the definition of h_{n+1}. This completes the proof of (b).

Now by (a) and (b) it is clear that for $1 \leqq i \leqq n$, $E(h_i, h_{n+1})=E_{i,n+1}$ and hence the induction step works. Let $R=\{h_1, \ldots, h_k\}$. Then $\mathscr{E}_R=\mathscr{E}$ obviously holds. □

After Theorem 2.1 there is a natural way to axiomatize full families of dependencies of any type. This follows next:

F'-axiom. Let $\mathscr{F} \subseteq P(\Omega) \times P(\Omega)$. Then $\mathscr{F}$ *satisfies the F'-axiom* if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ such that

(i) If $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{F}$ then there are $1 \leqq i < j \leqq k$ such that $X \subseteq E_{i,j}$ and $Y \nsubseteq E_{i,j}$.

(ii) If $(X, Y) \in \mathscr{F}$, $1 \leqq i < j \leqq k$ and $X \subseteq E_{i,j}$ then $Y \subseteq E_{i,j}$.

(iii) For any $1 \leqq i < j < l \leqq k$, $\{E_{i,j}, E_{i,l}, E_{j,l}\}$ is a Δ-system.

D'-axiom. Let $\mathscr{D} \subseteq P(\Omega) \times P(\Omega)$. Then $\mathscr{D}$ *satisfies the D'-axiom* if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ such that

(i) If $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{D}$ then there are $1 \leqq i < j \leqq k$ such that $X \cap E_{i,j} \neq \emptyset$ and $Y \cap E_{i,j} = \emptyset$.

(ii) If $(X, Y) \in \mathscr{D}$, $1 \leqq i < j \leqq k$ and $X \cap E_{i,j} \neq \emptyset$ then $Y \subseteq E_{i,j} \neq \emptyset$.

(iii) The same as (iii) of the F'-axiom.

S'-axiom. Let $\mathscr{S} \subseteq P(\Omega) \times P(\Omega)$. Then $\mathscr{S}$ *satisfies the S'-axiom* if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ such that

(i) If $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{S}$ then there are $1 \leqq i < j \leqq k$ such that $X \cap E_{i,j} \neq \emptyset$ and $Y \nsubseteq E_{i,j}$.

(ii) If $(X, Y) \in \mathscr{S}$, $1 \leqq i < j \leqq k$ and $X \cap E_{i,j} \neq \emptyset$ then $Y \subseteq E_{i,j}$.

(iii) The same as (iii) of the F'-axiom.

W'-axiom. Let $\mathscr{W} \subseteq P(\Omega) \times P(\Omega)$. Then $\mathscr{W}$ *satisfies the W'-axiom* if there is a natural number k and an indexed set of subsets of Ω, $\{E_{i,j}: 1 \leqq i < j \leqq k\}$ such that

(i) If $(X, Y) \in P(\Omega) \times P(\Omega) \setminus \mathscr{W}$ then there are $1 \leqq i < j \leqq k$ such that $X \subseteq E_{i,j}$ and $Y \cap E_{i,j} = \emptyset$.

(ii) If $(X, Y) \in \mathscr{W}$, $1 \leqq i < j \leqq k$ and $X \subseteq E_{i,j}$ then $Y \cap E_{i,j} \neq \emptyset$.

(iii) The same as (iii) of the F'-axiom.

REMARK. Observe that the $E_{i,j}$-s in the F'-axiom are maximal dependent sets, i.e. if $(X, Y)\in\mathcal{F}$ and $X\subseteq E_{i,j}$ then $Y\subseteq E_{i,j}$.

Theorem 2.2. (i) Let $\mathcal{Y}\subseteq P(\Omega)\times P(\Omega)$ and $Y\in\{F, D, S\}$. Then $\mathcal{Y}$ satisfies the Y-axiom iff $\mathcal{Y}$ satisfies the Y'-axiom.

(ii) Let Ω be a finite set, $|\Omega|\geqq 3$. Then there is a $\mathcal{W}\subseteq P(\Omega)\times P(\Omega)$ such that $\mathcal{W}$ satisfies the W-axiom and $\mathcal{W}$ does not satisfy the W'-axiom.

Proof. (i) Let first $Y=F$ and suppose that $\mathcal{Y}$ satisfies the F-axiom. Write $\mathcal{Y}=\mathcal{F}$. For any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathcal{F}$ take an $E(X, Y)\subseteq\Omega$ guaranteed by the F-axiom. List these $E(X, Y)$-s as $E_2, \ldots, E_k$ (the indices begin with 2). For $1<j\leqq k$ let $E_{1,j}=E_j$ and for $1<i<j\leqq k$ let $E_{i,j}=E_i\cap E_j$. We claim that $\{E_{i,j}\colon 1\leqq i<j\leqq k\}$ demonstrates that $\mathcal{F}$ satisfies the F'-axiom. The requirement (i) of the F'-axiom holds by $\{E_2, \ldots, E_k\}\subseteq\{E_{i,j}\colon 1\leqq i<j\leqq k\}$. We left to the reader to check that (ii) holds too. To prove (iii) of the F'-axiom let $1\leqq i<j<l\leqq k$.

We distinguish two cases:

(a) $i=1$. Then $E_{i,j}=E_j$; $E_{i,l}=E_l$ and $E_{j,l}=E_j\cap E_l$. Thus the intersection of any two members of $\{E_{i,j};\ E_{i,l};\ E_{j,l}\}$ is $E_j\cap E_l$. This means that $\{E_{i,j};\ E_{i,l};\ E_{j,l}\}$ is Δ-system.

(b) $1<i$. Then $E_{i,j}=E_i\cap E_j$; $E_{i,l}=E_i\cap E_l$ and $E_{j,l}=E_j\cap E_l$. Thus the intersection of any two members of $\{E_{i,j};\ E_{i,l};\ E_{j,l}\}$ is $E_i\cap E_j\cap E_l$. This means that $\{E_{i,j};\ E_{i,l};\ E_{j,l}\}$ is a Δ-system.

If $\mathcal{Y}$ satisfies the F'-axiom then $\mathcal{Y}$ obviously satisfies the F-axiom.

Now let $Y=D$ and suppose that $\mathcal{Y}$ satisfies the D-axiom. Write $\mathcal{Y}=\mathcal{D}$. For any $(X, Y)\in P(\Omega)\times P(\Omega)\setminus\mathcal{D}$ take an $E(X, Y)\subseteq\Omega$ guaranteed by the D-axiom. List these $E(X, Y)$-s as $E_1, \ldots, E_k$. For $1\leqq i\leqq k$ let $E_{2i-1,2i}=E_i$ and if $1\leqq i<j\leqq 2k$ and $E_{i,j}$ is still undefined then let $E_{i,j}=\emptyset$. It is easy to see that $\{E_{i,j}\colon 1\leqq i<j\leqq 2k\}$ shows the D'-axiom to hold for $\mathcal{D}$. If $\mathcal{Y}$ satisfies the D'-axiom then it trivially satisfies the D-axiom.

The case $Y=S$ is an easy modification of the proof in the case $Y=F$.

(ii) For the sake of simplicity suppose that $\Omega=\{a, b, c\}$. (In the general case pick two different elements a and b of Ω. The role of $\{c\}$ will played by $\Omega\setminus\{a, b\}$). Let $\mathcal{W}=\{(A, B)\in P(\Omega)\times P(\Omega)\colon A\subseteq\{a\}\Rightarrow a\in B$ and $A\subseteq\{b\}\Rightarrow b\in B\}$. Then $\mathcal{W}$ satisfies the W-axiom while if $(A, B)\in P(\Omega)\times P(\Omega)\setminus\mathcal{W}$ then either ($A\subseteq\{a\}$ and $a\notin B$) or ($A\subseteq\{b\}$ and $b\notin B$). For (A, B), $E=\{a\}$ taken in the 1st case and $E=\{b\}$ in the 2nd one shows the W-axiom to hold.

We claim that $\mathcal{W}$ does not satisfy the W'-axiom. Suppose indirectly that $\mathcal{E}=\{E_{i,j}\colon 1\leqq i<j\leqq k\}$ is a system that shows the W'-axiom to hold for $\mathcal{W}$.

Then

(1) $\{a\}\in\mathcal{E}$ and $\{b\}\in\mathcal{E}$ while $(\{a\}, \Omega\setminus\{a\})\in P(\Omega)\times P(\Omega)\setminus\mathcal{W}$ and $(\{b\}, \Omega\setminus\{b\})\in P(\Omega)\times P(\Omega)\setminus\mathcal{W}$ hold.

(2) $\emptyset\notin\mathcal{E}$ and $\{c\}\notin\mathcal{E}$ while $(\emptyset, \Omega)\in\mathcal{W}$ and $(\{c\}, \Omega\setminus\{c\})\in\mathcal{W}$ hold.

By (1), $\{a\}\in\mathcal{E}$ and $\{b\}\in\mathcal{E}$, that is $\{a\}=E_{i,j}$ and $\{b\}=E_{l,m}$ for some $1\leqq i, j, l, m\leqq k$. We distinguish two cases:

(a) $i=l$. Then $\{a\}=E_{i,j}$ and $\{b\}=E_{i,m}$. $\{E_{i,j};\ E_{i,m};\ E_{j,m}\}$ is a Δ-system, consequently either $E_{j,m}=\emptyset$ or $E_{j,m}=\{c\}$. Both cases contradict (2).

(b) $|\{i, j, l, m\}|=4$. We may suppose that $(i, j)=(1, 2)$ and $(l, m)=(3, 4)$ while we are interested in $\{E_{i,j};\ E_{i,l};\ E_{i,m};\ E_{i,l};\ E_{j,m};\ E_{l,m}\}$. What is $E_{2,3}$?

By (2) $E_{2,3}\neq\emptyset$ and $E_{2,3}\neq\{c\}$. The cases $E_{2,3}=\{a\}$ or $\{b\}$ arise to the case (a). $E_{2,3}\neq\{b, c\}$ while $E_{1,2}$; $E_{2,3}$; $E_{1,3}$ form a Δ-system and thus $E_{2,3}=\{b, c\}$ implies $E_{1,3}=\emptyset$, contradicting 2. Similarly $E_{2,3}\neq\{a, c\}$. Thus $\{a, b\}\subseteqq E_{2,3}$. The possibilities for $E_{1,3}$ are the same as for $E_{2,3}$ that is $\{a, b\}\subseteqq E_{1,3}$. But then $b\in E_{1,3}\cap E_{2,3}$ and $b\notin E_{1,2}$, contradicting $\{E_{1,2}; E_{2,3}; E_{1,3}\}$'s being a Δ-system.

The proof is complete. □

REMARK. Theorem 2.2 demonstrates the difference between the weak dependency and the rest.

Theorem 2.3. Let $\mathscr{Y}\subseteqq P(\Omega)\times P(\Omega)$ satisfy the Y'-axiom for some $Y\in\{F, D, S, W\}$. Then there is a relation R over Ω with $\mathscr{Y}=\mathscr{Y}_R$. Conversely, if R is a relation over Ω then $\mathscr{Y}_R$ satisfies the Y'-axiom.

Proof. Let $\mathscr{E}=\{E_{i,j}: 1\leqq i<j\leqq k\}$ show that $\mathscr{Y}$ satisfies the Y'-axiom. Then the requirement (iii) of the Y'-axiom and Theorem 2.1 (ii) imply that there is a relation R over Ω such that $\mathscr{E}_R=\mathscr{E}$. By the Y'-axiom it is obvious that $\mathscr{Y}=\mathscr{Y}_R$.

Conversely, if R is a relation over Ω, then writing $R=\{h_1, \ldots, h_k\}$, $E_{i,j}=E(h_i, h_j)$; $\{E_{i,j}: 1\leqq i<j\leqq k\}$ shows that $\mathscr{Y}_R$ satisfies the Y'-axiom. □

§ 3. Combinatorial results

Definition 3.1. Let $\mathscr{F}$ be a full f-family and let $A\subseteqq\Omega$. Then A is a *candidate key for* $\mathscr{F}$ if $(A, \Omega)\in\mathscr{F}$ and for any $A'\subset A$ $(A', \Omega)\notin\mathscr{F}$ holds. Let R be a relation over Ω, then the set of candidate keys of R is the set of candidate keys of $\mathscr{F}_R$.

Let $\mathscr{C}$ denote the set of candidate keys of $\mathscr{F}$. Then $\mathscr{C}$ is a Sperner system, i.e. $(\forall A, B\in\mathscr{C})$ $(A\subseteqq B\Rightarrow A=B)$. We deal with the following question of [9]:

(*) Let $r(n)$ denote the smallest integer for which *any* Sperner system $C\subseteqq P(\Omega)$ is the set of candidate keys of a suitable relation over the n-element set Ω with at most $r(n)$ rows. What can be said about $r(n)$?

In [9] it is shown that for any Sperner system there is a relation with this system as its set of candidate keys and that

$$\sqrt{2\binom{n}{[n/2]}}\leqq r(n)\leqq 2\cdot\binom{n}{[n/2]}.$$

We give sharper estimations for $r(n)$.

Theorem 3.1. $\frac{1}{n^2}\binom{n}{[n/2]}<r(n)\leqq\binom{n}{[n/2]}+1.$

Proof. First we prove the upper bound. Let $\mathscr{C}\subseteqq P(\Omega)$ be a Sperner system. Let $\mathscr{B}$ consist of the maximal sets that do not contain any members of $\mathscr{C}$. Let $B_2, \ldots, B_k$ be the members of $\mathscr{B}$. For $1<j\leqq k$ let $E_{1,j}=B_j$ and for $1<i<j\leqq k$ let $E_{i,j}=B_i\cap B_j$. Then $\{E_{i,j}: 1\leqq i<j\leqq k\}$ satisfies the requirements of the Theorem 2.1 (ii), hence there is a relation R over Ω with k rows such that $\mathscr{E}_R=\{E_{i,j}: 1\leqq i<j\leqq k\}$. Then obviously $\mathscr{C}$ is the set of candidate keys of R. It is

trivial that $\mathscr{B}$ is a Sperner system, and thus $|\mathscr{B}| \leqq \binom{n}{[n/2]}$ that is $k \leqq \binom{n}{[n/2]}+1$. The rest of the proof is due to L. Rónyai. We start with two trivial observations.

1. Let R be a relation over Ω with r rows. Then there is a relation R' over Ω such that R' uses no more than r symbols and $\mathscr{E}_R=\mathscr{E}_{R'}$.

2. Let R be a relation over Ω with r rows and let $r'>r$. Then there is a relation R' over Ω with r' rows such that $\mathscr{E}_R=\mathscr{E}_{R'}$.

By 1. and 2. the number of Sperner systems which may be represented as sets of candidate keys of relations with r rows is no more then $r^{r \cdot n}$. Hence

$$r(n)^{r(n)\cdot n} > 2^{\binom{n}{[n/2]}}$$

which implies

$$r(n) > \frac{1}{n^2}\binom{n}{[n/2]}. \quad \square$$

It is natural to ask the following analogon of $(*)$:

Let $R(n)$ denote the smallest integer for which *any* full family $\mathscr{F} \subseteqq P(\Omega)\times P(\Omega)$ is the set of functional dependencies of a suitable relation over the n-element set Ω with at most $R(n)$ rows. What can be said about $R(n)$?

By the proof of Theorem 2.2 (i) it is obvious that $R(n) \leqq$(the maximal number of subsets of Ω such that the intersection of any two of them is not a third). Thus, by a theorem of D. Kleitman [12], $R(n) \leqq c \cdot \binom{n}{[n/2]}$ where $c=3/2$. Z. Füredi and J. Pach have shown, that this number is less then $(1+(c\cdot \log n)/n)\binom{n}{[n/2]}$.

Lastly we give the combinatorial characterization — according to § 0 — of the sets which are of minimal cardinality with respect to the property that they imply all the dependencies of a given full f-family.

We need some definitions and a lemma.

Definition 3.2. Let $\mathscr{M} \subseteqq P(\Omega)$.

(i) We say that $\mathscr{M}$ *has the intersection property* if for any $\mathscr{M}' \subseteqq \mathscr{M}$, $\bigcap \mathscr{M}' \in \mathscr{M}$ holds.

(ii) An $M \in \mathscr{M}$ *is irreducible* if $M \neq \bigcap \{M' \in \mathscr{M}\colon M \subset M'\}$ (recall that $\subset$ means strict inclusion).

(iii) An $\mathscr{N} \subseteqq \mathscr{M}$ *generates* $\mathscr{M}$ if $\mathscr{M}=\{\bigcap \mathscr{N}'\colon \mathscr{N}' \subseteqq \mathscr{N}\}$.

Lemma 3.1. Let $\mathscr{M}$ have the intersection property and let $\mathscr{N}=\{M \in \mathscr{M}\colon M$ is irreducible$\}$. Then an $\mathscr{N}' \subseteqq \mathscr{M}$ generates $\mathscr{M}$ iff $\mathscr{N} \subseteqq \mathscr{N}'$.

Proof. The following proof is standard in lattice theory. If $\mathscr{N}'$ generates $\mathscr{M}$, then $\mathscr{N} \subseteqq \mathscr{N}'$ is obvious. For the converse we have to prove that $\mathscr{N}$ generates $\mathscr{M}$. Suppose indirectly that there is an $X \in \mathscr{M} \setminus \mathscr{N}$ such that $X \neq \bigcap \{Y\colon Y \in \mathscr{N}$ & $X \subset Y\}$. Let X be of minimal cardinality with respect to this property. $X \notin \mathscr{N}$ means that $X=\bigcap \{Y\colon Y \in \mathscr{M}$ & $X \subset Y\}$, hence $X \subset Y$ implies that there is an $\mathscr{N}_y \subseteqq \mathscr{N}$ such that $Y=\bigcap \mathscr{N}_y$. Let $\mathscr{N}_x=\bigcup \{\mathscr{N}_y\colon X \subset Y$ and $Y \in \mathscr{M}\}$. Then $\mathscr{N}_x \subseteqq \mathscr{N}$ and $X=\bigcap \mathscr{N}_x$ which is a contradiction. $\square$

Remark. Observe that the proofs of the Theorems in [2] are essentially our proof of Lemma 3.1.

Corollary. If $\mathscr{M}$ has the intersection property then there is exactly one $\mathscr{N} \subseteqq \mathscr{M}$ which generates $\mathscr{M}$ and has minimal cardinality.

Theorem 3.2. Let $\mathscr{F}$ be a full f-family, let $\mathscr{B}$ be the set of maximal dependent sets for $\mathscr{F}$ and let $\mathscr{C}$ be the set which generates $\mathscr{B}$ and has minimal cardinality (in [1] it is shown that $\mathscr{B}$ has the intersection property).

Then for any $\mathscr{F}' \subseteqq \mathscr{F}$ we have the following: $\mathscr{F}'$ implies all the dependencies of $\mathscr{F}$ and $\mathscr{F}'$ has minimal cardinality with respect to this property if and only if for any $C \in \mathscr{C}$ there is an $A_c \subseteqq \Omega$ such that $\mathscr{F}' = \{(A_c, C): C \in \mathscr{C}\}$.

We left the easy proof of the Theorem to the reader. We think that it is interesting to compare Theorem 3.2 with the Theorem on pp. 16 of [2].

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 Publ. Co., 1974, pp. 580—583.
[2] ARMSTRONG, W. W., On the generation of dependency structures of relational data bases, *Publication* 272, Université de Montréal, 1977.
[3] BEERI, C., R. FAGIN, J. H. HOWARD, A complete axiomatization for functional and multivalued dependencies in database relations, *Proc. ACM SIGMOD* Int. Conf. an Management of Data, Toronto, 1977, pp. 47—61.
[4] BÉKÉSSY, A., J. DEMETROVICS, Contribution to the theory of data base relations, *Discrete Math.* v. 27, 1979, pp. 1—10.
[5] BÉKÉSSY, A., J. DEMETROVICS, L. HANNÁK, P. FRANKL, G. KATONA, On the number of maximal dependencies in a data base relation of fixed order, *Discrete Math.*, v. 29, 1980, pp. 83—88.
[6] CODD, E. F., A relational model of data for large shared data banks, *Comm. ACM*, v. 13, 1970, pp. 377—387.
[7] CODD, E. F., Further normalization of the data base relational model, Courant Computer Science Symposia 6 Data Base System, Prentice Hall, Englewood Cliffs, N. J., 1971, pp. 33—64.
[8] CZÉDLI, G., Függőségek relációs adatbázis modellben, *Alkalmaz. Mat. Lapok*, 1980.
[9] DEMETROVICS, J., Candidate keys and antichains, *SIAM J. on Algebraic and Discrete Methods*, v. 1, 1980, p. 92.
[10] DEMETROVICS, J., On the equivalence of candidate keys with Sperner systems, *Acta Cybernet.*, v. 4, 1979, pp. 247—252.
[11] FAGIN, R., Multivalued dependencies and a new normal form for relational data-bases, *ACM Trans. Database Sys.*, v. 2, 1977, pp. 262—278.
[12] KLEITMAN, D., On a problem of Erdős, *Proc. Amer. Math. Soc.*, v. 18, 1966, pp. 139—141.
[13] MENDELZON, A. O., On axiomatizing multivalued dependencies in relational databases, *J. Assoc. Comput. Sci.*, v. 26, 1979, pp. 37—44.
[14] RISSANEN, J., Independent components of relations, *ACM Trans. Database Sys.*, v. 2, 1977, pp. 317—325.
[15] YU, C. T., D. T. JOHNSON, On the complexity of finding the set of candidate keys for a given set of functional dependencies, *Inform. Process. Lett.*, v. 5, 1976, pp. 100—101.

(Received Jan. 19, 1981)

Algebraic representation of language hierarchies

By T. GERGELY

1. Introduction

The investigation of the connections between completely different languages or between theories formulated within these languages is a problem of growing importance in System Science, in Theoretical Linguistics and in many branches of Computer Science. E.g. this problem has arisen in high level program specification (see e.g. BURSTALL—GOGUEN [6, 7] and DÖMÖLKI [9]) in abstract data type research (see e.g. HUPBACH [13]) and in computer system modelling (see e.g. RATTRAY—RUS [17]).

In order to establish a connection between two languages first a connection i.e. a method of translation between their syntax might be looked for. Another possibility is connected with the interpretation of one syntax into another by introducing appropriate mathematical tools (see e.g. MONK [15] and BLUM—ESTES [5]). However usually there are a lot of possibilities of interpretation. As to handle them together, i.e. to investigate the possible connections in a complex way, the so called theory morphisms have been introduced (see e.g. AGN [3], BURSTALL—GOGUEN [6] and WINKOWSKI [19]). It turned out that category theory provides an adequate frame for the required complex analysis. However it would be quite useful to characterize the category corresponding to language hierarchy by the use of a well developed "culture" like universal algebra. Here we show that this characterization is possible by the use of the culture of cylindric algebras.

Throughout the paper it is supposed that the reader is familiar with basic notions of universal algebra and category theory.

2. Locally finite dimensional cylindric algebras

Cylindric algebras provide a tool to handle classical first order logic properly in algebraical way. They are in the same relationship to first order logic as Boolean algebras are to propositional logic. Here we present the basic notions and properties of the theory of these algebras relevant to our aim.

Definition 2.1. A similarity type t is a pair of functions $\langle t_F, t_R\rangle$ such that $\mathrm{Rg}\, t_F \subseteq \omega$ and $\mathrm{Rg}\, t_R \subseteq \omega \setminus \{0\}$, $\mathrm{Do}\, t_F \cap \mathrm{Do}\, t_R = \emptyset$. The elements of $\mathrm{Do}\, t_F$ and $\mathrm{Do}\, t_R$

are called function and relation symbols, respectively. Here Dof and Rgf stand for the domain and range of the function f respectively. □

Note that a similarity type could be defined in such a way that it contains only relation symbols because functions are but special relations (cf. AGN [4]).

Let t be an arbitrary similarity type with $t_R=\emptyset$. The class of all t-type algebras will be denoted by Alg (t). The class of all t-type algebras forms a category denoted by **Alg** (t) in the usual way i.e. the class of objects is Ob $(\mathbf{Alg}(t))=\mathrm{Alg}(t)$ and the class of morphisms consist of all the homomorphisms. Further on, the boldface version of a notion corresponding to a class of algebras refers to the corresponding category.

Let us fix and ordinal α and the following similarity type $l_\alpha=\{\langle +, 2\rangle, \langle \cdot, 2\rangle, \langle -, 1\rangle, \langle 0, 0\rangle, \langle 1, 0\rangle\}\cup\{\langle c_i, 0\rangle: i<\alpha\}\cup\{\langle d_{ij}, 0\rangle: i, j<\alpha\}$, which for the sake of convenience is denoted by

$$l_\alpha = \{\langle +, 2\rangle, \langle \cdot, 2\rangle, \langle -, 1\rangle, \langle 0, 0\rangle, \langle 1, 0\rangle, \langle c_i, 0\rangle, \langle d_{ij}, 0\rangle: i, j<\alpha\}.$$

Now we define a special subclass of Alg (l_α) as follows.

Definition 2.2. An l_α-type algebra $\mathfrak{A}=\langle A, +^{\mathfrak{A}}, \cdot^{\mathfrak{A}}, -^{\mathfrak{A}}, 0^{\mathfrak{A}}, 1^{\mathfrak{A}}, c_i^{\mathfrak{A}}, d_{ij}^{\mathfrak{A}}\rangle_{i,j<\alpha}$ is said to be a cylindric algebra of dimension α iff it satisfies the conditions below. (For the sake of convenience we omit the supercript $\mathfrak{A}$ speaking about the concret operations of a model $\mathfrak{A}$, i.e. where it does not lead to ambiguity we simply write $\mathfrak{A}=\langle A, +, \cdot, -, 0, 1, c_i, d_{ij}\rangle_{i,j<\alpha}$.)

(i) $\langle A, +, \cdot, -, 0, 1\rangle$ is a Boolean algebra,
(ii) $c_i 0=0$,
(iii) $c_i x\cdot x=x$,
(iv) $c_i(x\cdot c_i y) = c_i x\cdot c_i y$
(v) $c_i c_j x=c_j c_i x$,
(vi) $d_{ii}=1$,
(vii) if $i\neq j$, n then $d_{jn}=c_i(d_{ji}\cdot d_{in})$,
(viii) if $i\neq j$ then $c_i(d_{ij}\cdot x)\cdot c_i(d_{ij}\cdot -x)=0$ for any $i, j<\alpha$. □

Further on the Gothic capital letters refer to algebras while the corresponding Roman capital letters do to their universe.

Let CA_α denote the class of all cylindric algebras of dimension α. The homomorphisms on CA_α are defined as usually, i.e. such that they preserve all operations of the cylindric algebras. The intuition for CA_α theory comes from cylindric set algebras a systematic exposition of which is HMTAN [12].

NOTATION. Sb $K\overset{d}{=}\{X: X\subseteq K\}$ for any class K.

Definition 2.3. Let $\mathfrak{A}\in\mathrm{Alg}(l_\alpha)$. The function $\Delta^{\mathfrak{A}}: A\to\mathrm{Sb}\,\alpha$, which renders to any $a\in A$ the following set $\Delta^{\mathfrak{A}}a\overset{d}{=}\{i\in\alpha: c_i^{\mathfrak{A}}\, a\neq a\}$ is said to be the *dimension-sensitivity function.* □

Definition 2.4. The following class of l_α-type algebras $\mathrm{LF}_\alpha=\{\mathfrak{A}\in\mathrm{Alg}(l_\alpha)$: for any $a\in A$, $|\Delta^{\mathfrak{A}}a|<\omega\}$ is said to be the class of *locally finite dimensional algebras.* □

Proposition 2.1. Let $\mathfrak{A}, \mathfrak{B} \in \mathrm{Alg}(l_\alpha)$, $a \in A$ and let $f: \mathfrak{A} \to \mathfrak{B}$ be a homomorphism. Then $\Delta^{\mathfrak{B}} f(a) \subseteq \Delta^{\mathfrak{A}} a$.

Proof. Let $i \in \Delta f(a)$, i.e. $c_i f(a) \neq f(a)$. Since f is a homomorphism this is possible only in the case $c_i a \neq a$, i.e. when $i \in \Delta a$. □

Now let us define the locally finite dimensional cylindric algebras as follows.

Definition 2.5. $\mathrm{Lf}_\alpha \stackrel{d}{=} \mathrm{CA}_\alpha \cap \mathrm{LF}_\alpha$. □

Now we turn to the relationships between first order logic and cylindric algebras.

First we recall some well-known notions of first order logic.

Let t be an arbitrary similarity type and α be an arbitrary ordinal. A t-type first order language of α variables with equality is a triple $\langle F_t^\alpha, M_t, \models \rangle$ where F_t^α is the set of all t-type formulas containing variable symbols belonging to the set $\{x_i: i \in \alpha\}$ of variables of cardinality $|\alpha|$, M_t denotes the class of all t-type models; $\models \subseteq M_t \times F_t^\alpha$ is the validity relation. It is supposed that the symbol $=$ of equality relation is interpreted in each model as identity.

If $\mathrm{Ax} \subseteq F_t^\alpha$ and $\varphi \in F_t^\alpha$ then $\mathrm{Ax} \models \varphi$ means that φ is a semantical consequence of Ax.

To each F_t^α there corresponds an l_α-type algebra the so called formula algebra $\mathfrak{F}_t^\alpha = \langle F_t^\alpha, +, \cdot, -, 0, 1, c_i, d_{ij}: i, j < \alpha \rangle$ where for any $\varphi, \psi \in F_t^\alpha$, $i, j < \alpha$

$\varphi + \psi$	stands for	$\varphi \vee \psi$,
$\varphi \cdot \psi$	stands for	$\varphi \wedge \psi$,
$-\varphi$	stands for	$\neg \varphi$,
0	stands for	$\neg x = x$,
1	stands for	$x = x$,
$c_i \varphi$	stands for	$\exists x_i \varphi$ and
d_{ij}	stands for	$x_i = x_j$.

Definition 2.6. A pair $T = \langle \mathrm{Ax}, F_t^\alpha \rangle$, where $\mathrm{Ax} \subseteq F_t^\alpha$ is said to be a *theory* in α variables. □

Note that a theory provides a sublanguage of $\langle F_t^\alpha, M_t, \models \rangle$, namely, the triple $\langle F_t^\alpha, \mathrm{Mod}(\mathrm{Ax}), \models \rangle$, where $\mathrm{Mod}(\mathrm{Ax}) \stackrel{d}{=} \{\mathfrak{A} \in M_t: \mathfrak{A} \models \mathrm{Ax}\}$.

Let $T = \langle \mathrm{Ax}, F_t^\alpha \rangle$ be a theory and let $\equiv_T \subseteq F_t^\alpha \times F_t^\alpha$ be the semantic equivalence w.r.t. T defined as follows: For any $\varphi, \psi \in F_t^\alpha$, $\varphi \equiv_T \psi$ iff $\mathrm{Ax} \models \varphi \leftrightarrow \psi$. Further on for any $\varphi \in F_t^\alpha$ let $\varphi / \equiv_T$ denote the corresponding equivalence class, i.e. $\varphi / \equiv_T \stackrel{d}{=} \{\psi \in F_t^\alpha: \varphi \equiv_T \psi\}$.

Definition 2.7. The equivalence classes $\varphi / \equiv_T$ $(\varphi \in F_t^\alpha)$ are said to be *concepts* of the corresponding theory T. The set of concepts of a theory T is $C_T \stackrel{d}{=} F_t^\alpha / \equiv_T$, where $F_t / \equiv_T$ means the factorization of the set of formulas into such classes any two elements of which are semantically equivalent w.r.t. T. □

Note that the classes of C_T contain both open and closed formulas. (A formula is closed if each variable symbol occurs bound in it.) With respect to the open formulas it is important to remark that interpreting them in a model the variable symbols occurring free should be handled as constants. (See Examples below.)

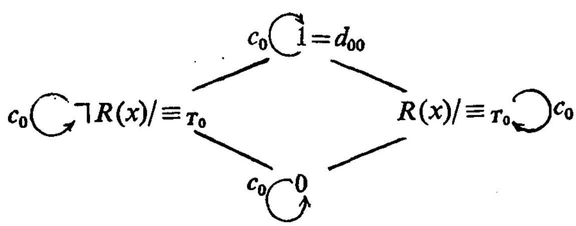
On the base of the set of concepts of a theory T we define another l_α-type algebra.

Definition 2.8. The concept algebra of a theory T is defined as follows. $\mathfrak{C}_T = \mathfrak{F}_t^\alpha / \equiv_T$, hence $\mathfrak{C}_T = \langle C_T, +, \cdot, -, 0, 1, c_i, d_{ij}: i, j<\alpha\rangle$. □

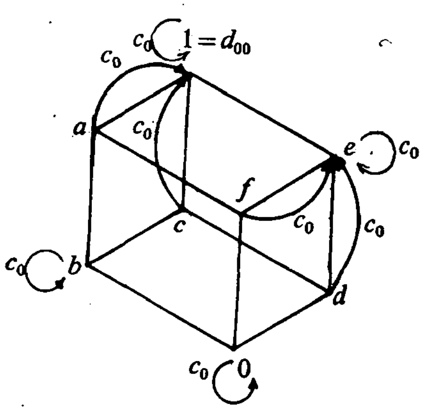
To see that this definition is correct one has to check that $\equiv_T$ is a congruence relation on the algebra $\mathfrak{F}_t^\alpha$.

Let us illustrate the notion of concept algebra by the following

EXAMPLES. a) Let $T_0 = \langle \mathrm{Ax}_0, F_{t_0}^1\rangle$ be a theory, where $t_0 = \langle \emptyset, \{\langle R, 1\rangle\}\rangle$ and $\mathrm{Ax}_0 = \{(\exists x R(x) \to \forall x R(x))\}$. Then the corresponding concept algebra is as follows. (About the graphical representation of algebras see AGN [4].)



b) Let $T_1 = \langle \mathrm{Ax}_1, F_{t_1}^1\rangle$ be a theory where $t_1 = \langle \emptyset, \{\langle A, 1\rangle\}\rangle$ and $\mathrm{Ax}_1 = \{\exists x \neg A(x)\}$. Then the corresponding concept algebra is as follows, where



$a = \neg A(x)/\equiv_{T_1}$,
$b = \neg\exists x A(x)/\equiv_{T_1}$,
$c = (\neg\exists x A(x) \vee A(x))/\equiv_{T_1}$,
$d = A(x)/\equiv_{T_1}$,
$e = \exists x A(x)/\equiv_{T_1}$ and
$f = (\neg A(x) \vee \exists x A(x))/\equiv_{T_1}$. □

Let C_α be the class of concept algebras with α variables, i.e. $C_\alpha \overset{\mathrm{d}}{=} \{\mathfrak{C}_T: T = \langle \mathrm{Ax}, F_t^\alpha\rangle, \mathrm{Ax} \subseteq F_t^\alpha, t$ is an arbitrary similarity type$\}$.

Note that concept algebras $\mathfrak{C}_T$ are denoted in Definition 12.22 of MONK [15] by $\mathfrak{M}_\Gamma^L$ (where L is a first order language and Γ is a set of sentences in L).

No we turn to the investigation of the connection of the classes C_α and Lf_α.

Proposition 2.2. Let $\mathfrak{C}_{\langle \mathrm{Ax}, F_t^\alpha\rangle} \in C_\alpha$. Then $\mathfrak{C}_{\langle \mathrm{Ax}, F_t^\alpha\rangle} \in \mathrm{Lf}_\alpha$.

Proof. Any formula $\varphi \in F_t^\alpha$ contains finitely many variables, the set of which, say, is $\mathrm{Var}\ \varphi$. Let $x_k \in \mathrm{Var}\ \varphi$ for some $k<\alpha$, then $\varphi \equiv_T \exists x_k \varphi$. Thus $\Delta\varphi \subseteq$

$\subseteq \{i: x_i \in \text{Var}\, \varphi\}$ so it is finite. It is easy to verify that $\mathfrak{C}_{\langle Ax, F_t \rangle}$ satisfies conditions (i)—(viii) of Definition 2.2. □

Let $\mathbf{C}_\alpha$ be defined to be the full subcategory of $\mathbf{Alg}\,(l_\alpha)$ such that $\text{Ob}\,\mathbf{C}_\alpha = C_\alpha$.

Now we turn to the investigation of the role of the category $\mathbf{C}_\alpha$ w.r.t. other subcategories of $\mathbf{Alg}\,(l_\alpha)$. First we recall (see MAC LANE [14])

Definition 2.9. Let $\mathbf{A}_1$ and $\mathbf{A}_2$ be two arbitrary categories. A functor F from $\mathbf{A}_1$ into $\mathbf{A}_2$ is defined to be a pair $F=(F_{\text{Ob}}, F_{\text{Mor}})$ of functions F_{Ob}: $\text{Ob}\,\mathbf{A}_1 \to \text{Ob}\,\mathbf{A}_2$ and F_{Mor}: $\text{Mor}\,\mathbf{A}_1 \to \text{Mor}\,\mathbf{A}_2$ such that (i)—(iii) below hold:

(i) If $f \in \text{Hom}\,(A, B)$ in $\mathbf{A}_1$ then $F_{\text{Mor}}(f) \in \text{Hom}\,(F_{\text{Ob}}(A), F_{\text{Ob}}(B))$ in $\mathbf{A}_2$;

(ii) $F_{\text{Mor}}(f \circ g) = F_{\text{Mor}}(f) \circ F_{\text{Mor}}(g)$ for all $f, g \in \text{Mor}\,\mathbf{A}_1$;

(iii) $F_{\text{Mor}}(\text{Id}_A) = \text{Id}_{F_{\text{Ob}}}(A)$ for any $A \in \text{Ob}\,\mathbf{A}_1$.

Here Id_A: $A \to A$ is the identity morphism corresponding to A. Note that instead of F_{Ob} and F_{Mor} we often write only F.

For a category $\mathbf{A}$ the identity functor $\text{Id}_{\mathbf{A}}$ sends A to A and f to f for all $A \in \text{Ob}\,\mathbf{A}$ and $f \in \text{Mor}\,\mathbf{A}$.

The categories $\mathbf{A}_1$ and $\mathbf{A}_2$ are *equivalent* iff there is a functor F: $\mathbf{A}_1 \to \mathbf{A}_2$, to which there is a backward functor G: $\mathbf{A}_2 \to \mathbf{A}_1$ and there are two natural isomorphisms θ: $F \circ G \dot{\to} \text{Id}_{\mathbf{A}_2}$ and ν: $G \circ F \dot{\to} \text{Id}_{\mathbf{A}_1}$.

The categories $\mathbf{A}_1$ and $\mathbf{A}_2$ are *isomorphic* iff there are functors F: $\mathbf{A}_1 \to \mathbf{A}_2$ and $G = \mathbf{A}_2 \to \mathbf{A}_1$ such that $G \circ F = \text{Id}_{\mathbf{A}_1}$ and $F \circ G = \text{Id}_{\mathbf{A}_2}$. □

Theorem 2.3. Let $\alpha \geqq \omega$ be an arbitrary infinite ordinal. The categories $\mathbf{Lf}_\alpha$ and $\mathbf{C}_\alpha$ are equivalent.

This theorem immediately follows from the following

Theorem 2.4. Let $\alpha \geqq \omega$. There are two full and faithful one-one functors F: $\mathbf{C}_\alpha \to \mathbf{Lf}_\alpha$ and G: $\mathbf{Lf}_\alpha \to \mathbf{C}_\alpha$ and two natural isomorphisms θ: $F \circ G \to \text{Id}_{\mathbf{Lf}_\alpha}$ and ν: $G \circ F \dot{\to} \text{Id}_{\mathbf{C}_\alpha}$ such that the functions F, G, θ and ν are definable (in a parameter free way) in ZFC set theory by formulas which are absolute (in set theoretical sense) and moreover these functions are primitive recursive (in the sense of DEVLIN [8] p. 29).

Proof. I. First we define the functors.

1. Let $\mathfrak{A} \in \text{Ob}\,\mathbf{Lf}_\alpha$. From 12.18, 12.25 and 12.28 of MONK [15], see also Theorem 5.2 of AGN [1] and Proposition 1 in [16], it follows that there is a theory $T_{\mathfrak{A}}$, i.e. a similarity type $t_{\mathfrak{A}}$ together with the corresponding set of formulas $F^\alpha_{t\,\mathfrak{A}}$ and a set $Ax_{\mathfrak{A}}$ of axioms such that $\mathfrak{A} \cong \mathfrak{C}_{T_{\mathfrak{A}}}$. Moreover from the proof of 12.28 of MONK [15] it follows that there is a function F_{Ob}: $\text{Ob}\,\mathbf{Lf}_\alpha \to \text{Ob}\,\mathbf{C}_\alpha$ such that

(i) for any $\mathfrak{A} \in \text{Ob}\,\mathbf{Lf}_\alpha$ $F_{\text{Ob}}(\mathfrak{A}) = \mathfrak{C}_{T_{\mathfrak{A}}}$;

(ii) there exists a function θ: $\text{Ob}\,\mathbf{Lf}_\alpha \to \text{Mor}\,\mathbf{Lf}_\alpha$ such that $\theta(\mathfrak{A}) = \text{Is}\,(F_{\text{Ob}}(\mathfrak{A}), \mathfrak{A})$ for any $\mathfrak{A} \in \text{Ob}\,\mathbf{Lf}_\alpha$. Here $\text{Is}\,(\mathfrak{A}, \mathfrak{B})$ denotes the set of isomorphisms from $\mathfrak{A}$ onto $\mathfrak{B}$.

(iii) the functions F_{Ob} and θ are definable in ZFC, i.e. there are set theoretic formulas $\varphi(x, y)$ and $\psi(x, y)$ such that

$$\text{ZFC} \vdash (\forall x \in \text{Ob}\,\mathbf{Lf}_\alpha)(\exists !\, y \varphi(x, y) \wedge \exists !\, y \psi(x, y))$$

and

$$\text{ZFC} \vdash (\forall x \in \text{Ob}\,\mathbf{Lf}_\alpha)\, \forall y, z((\varphi(x, y) \wedge \psi(x, z)) \to (y \in \text{Ob}\,\mathbf{C}_\alpha \wedge z \in \text{Is}\,(x, y))).$$

Above we assumed that Ob $\mathbf{Lf}_\alpha$ and Ob $\mathbf{C}_\alpha$ are also definable in ZFC, i.e. the expression "$y\in$Ob $\mathbf{C}_\alpha$" and "$y\in$Ob $\mathbf{Lf}_\alpha$" are formulas of one free variable y in ZFC. We omit the proof that this assumption is justified. Similarly "$z\in$Is (x, y)" is also a formula of ZFC of free variables x, y and z.

Moreover the formulas $\varphi(x, y)$ and $\psi(x, y)$ are absolute (in set theoretical sense).

(iv) The functions F_{Ob} and θ are primitive recursive in the sense of DEVLIN [8], i.e. they can be generated by the schemata (i)—(vii) of [8], p. 29. (And, even more we believe that these functions are rudimentary.)

Let $f\in$Mor $\mathbf{Lf}_\alpha$, namely let $f\in$Hom $(\mathfrak{A}, \mathfrak{B})$ for some $\mathfrak{A}, \mathfrak{B}\in$Ob $\mathbf{Lf}_\alpha$. We define $F_{\mathrm{Mor}}(f)\stackrel{\mathrm{d}}{=}[\theta(\mathfrak{B})]^{-1}\circ f\circ\theta(\mathfrak{A})$. Then clearly $F_{\mathrm{Mor}}(f)\in$Hom $\big(F_{\mathrm{Ob}}(\mathfrak{A}), F_{\mathrm{Ob}}(\mathfrak{B})\big)\subseteq$ $\subseteq$Mor $\mathbf{C}_\alpha$.

It is not difficult to verify that this function preserves composition and identity. Thus the pair $F=\langle F_{\mathrm{Ob}}, F_{\mathrm{Mor}}\rangle$ is a functor. Since the function θ is definable by an absolute formula of ZFC so is F_{Mor} and thus so is the functor F as well.

Now we show that the functor F is one-one.

a) Let $\mathfrak{A}$ and $\mathfrak{B}$ be two different elements of Ob $\mathbf{Lf}_\alpha$. Recall that at the beginning of the proof to every $\mathfrak{A}\in$Ob $\mathbf{Lf}_\alpha$ a theory $T_{\mathfrak{A}}$ was associated in a fixed way such that $T_{\mathfrak{A}}$ should be the theory constructed from $\mathfrak{A}$ in the proof of 12.28 [15]. We also recall that for any $\mathfrak{A}\in$Ob $\mathbf{Lf}_\alpha$ $F(\mathfrak{A})=C_{T_{\mathfrak{A}}}$.

(i) First we suppose that $\mathfrak{A}\neq\mathfrak{B}$ because $A\neq B$. In this case using the construction provided by MONK in the proof of 12.28 [15] we get different $F_{t_{\mathfrak{A}}}, F_{t_{\mathfrak{B}}}$, i.e. $F_{t_{\mathfrak{A}}}\neq F_{t_{\mathfrak{B}}}$. Hence $\mathbf{C}_{T_{\mathfrak{A}}}\neq\mathbf{C}_{T_{\mathfrak{B}}}$.

(ii) Let $A=B$. Since $\mathfrak{A}\neq\mathfrak{B}$ there is at least one operation symbol h say of n arguments and there are $a_1, \dots, a_n\in A$ such that $h^{\mathfrak{A}}(a_1, \dots, a_n)=a_0$ but $h^{\mathfrak{B}}(a_1, \dots, a_n)\neq a_0$. Therefore $\mathrm{Ax}_{\mathfrak{A}}\neq\mathrm{Ax}_{\mathfrak{B}}$.

Hence $\mathbf{C}_{T_{\mathfrak{A}}}\neq\mathbf{C}_{T_{\mathfrak{B}}}$. Thus F_{Ob} is one-one.

b) Since F_{Ob} is one-one it is sufficient to prove that F_{Mor} is one-one on Hom $(\mathfrak{A}, \mathfrak{B})$ for each $\mathfrak{A}, \mathfrak{B}\in$Ob $\mathbf{Lf}_\alpha$.

Let $f\circ g\in$Hom $(\mathfrak{A}, \mathfrak{B})$ be two elements of Mor $\mathbf{Lf}_\alpha$ such that $f\neq g$. By the definition of F_{Mor} obviously $F_{\mathrm{Mor}}(f)\neq F_{\mathrm{Mor}}(g)$. Thus F_{Ob} and F_{Mor} are one-one functions and F is so as well.

2. Now let us define the functor G: $\mathbf{C}_\alpha\to\mathbf{Lf}_\alpha$. From Proposition 2.2 it follows that for G we can choose the identical embedding, i.e. let $G=\langle G_{\mathrm{Ob}}, G_{\mathrm{Mor}}\rangle$ be such that for any $\mathfrak{A}\in$Ob $\mathbf{C}_\alpha$ and $f\in$Mor $\mathbf{C}_\alpha$, $G_{\mathrm{Ob}}(\mathfrak{A})=\mathfrak{A}$ and $G_{\mathrm{Mor}}(f)=f$. Clearly the functor G is definable by an absolute set theoretic formula and it is one-one, full and faithful.

From the above observations we have the following

Lemma 2.4.1. For any $\mathfrak{A}\in$Ob $\mathbf{Lf}_\alpha$ and $f\in$Mor $\mathbf{Lf}_\alpha$

$$G\circ F(\mathfrak{A})=F(\mathfrak{A}),\ G\circ F(f)=F(f)$$

and for any $\mathfrak{A}\in$Ob $\mathbf{C}_\alpha$ and $f\in$Mor $\mathbf{C}_\alpha$

$$F\circ G(\mathfrak{A})=F(\mathfrak{A}),\ F\circ G(f)=F(f).$$

II. Now we turn to the construction of the appropriate natural isomorphisms.

1. First we show that the function θ: Ob $\mathbf{Lf}_\alpha \to$ Mor $\mathbf{Lf}_\alpha$ defined in I.1 (ii) of this proof is a natural transformation from $G \circ F$ to $\mathrm{Id}_{\mathbf{Lf}_\alpha}$ which we denote following MAC LANE [14] by θ: $G \circ F \dot{\to} \mathrm{Id}_{\mathbf{Lf}_\alpha}$.

We would need a diagram of type

$$\begin{array}{ccc} G \circ F(\mathfrak{A}) & \xrightarrow{G \circ F(f)} & G \circ F(\mathfrak{B}) \\ \downarrow{\scriptstyle \theta(\mathfrak{A})} & & \downarrow{\scriptstyle \theta(\mathfrak{B})} \\ \mathrm{Id}_{\mathbf{Lf}_\alpha}(\mathfrak{A}) & \xrightarrow{\mathrm{Id}_{\mathbf{Lf}_\alpha}(f)} & \mathrm{Id}_{\mathbf{Lf}_\alpha}(\mathfrak{B}) \end{array} \qquad (*)$$

By Lemma 2.4.1 instead of the above diagram it is enough to consider the following one:

$$\begin{array}{ccc} F(\mathfrak{A}) & \xrightarrow{F(f)} & F(\mathfrak{B}) \\ \downarrow{\scriptstyle \theta(\mathfrak{A})} & & \downarrow{\scriptstyle \theta(\mathfrak{B})} \\ \mathfrak{A} & \xrightarrow{f} & \mathfrak{B} \end{array}$$

This diagram exists, so by Lemma 2.4.1 the diagram $(*)$ does exist as well.

By the definition of F_{Mor} we have: $F_{\mathrm{Mor}}(f) = [\theta(\mathfrak{B})]^{-1} \circ f \circ \theta(\mathfrak{A})$. Now it is easy to establish that the diagram commutes.

$$\theta(\mathfrak{B}) \circ F(f) = \theta(\mathfrak{B}) \circ [\theta(\mathfrak{B})]^{-1} \circ f \circ \theta(\mathfrak{A}) = f \circ \theta(\mathfrak{A}).$$

So θ: $G \circ F \to \mathrm{Id}_{\mathbf{Lf}_\alpha}$ is a natural transformation. Since for each $\mathfrak{A} \in$ Ob $\mathbf{Lf}_\alpha$, $\theta(\mathfrak{A}) \in \mathrm{Is}\,(G \circ F(\mathfrak{A}), \mathrm{Id}_{\mathbf{Lf}_\alpha}(\mathfrak{A}))$ we have that θ is a natural isomorphism.

2. Now we define ν: $F \circ G \dot{\to} \mathrm{Id}_{\mathbf{C}_\alpha}$. Let $\nu \stackrel{\mathrm{d}}{=} \theta \restriction \mathbf{C}_\alpha$. That is ν: Ob $\mathbf{C}_\alpha \to$ Mor $\mathbf{Lf}_\alpha$ such that for any $\mathfrak{A} \in$ Ob $\mathbf{C}_\alpha$, $\nu(\mathfrak{A}) = \theta(\mathfrak{A})$. Then for any $\mathfrak{A} \in$ Ob $\mathbf{C}_\alpha$, $\nu(\mathfrak{A}) \in \mathrm{Is}$ $(F \circ G(\mathfrak{A}), \mathrm{Id}_{\mathbf{C}_\alpha}(\mathfrak{A}))$. Let $\mathfrak{A}, \mathfrak{B} \in$ Ob $\mathbf{C}$ and $f \in$ Hom $(\mathfrak{A}, \mathfrak{B})$. Consider the following diagram

$$\begin{array}{ccc} F \circ G(\mathfrak{A}) & \xrightarrow{F \circ G(f)} & F \circ G(\mathfrak{B}) \\ \downarrow{\scriptstyle \nu(\mathfrak{A})} & & \downarrow{\scriptstyle \nu(\mathfrak{B})} \\ \mathrm{Id}_{\mathbf{C}_\alpha}(\mathfrak{A}) & \xrightarrow{\mathrm{Id}_{\mathbf{C}_\alpha}(f)} & \mathrm{Id}_{\mathbf{C}_\alpha}(\mathfrak{B}) \end{array}$$

By Lemma 2.4.1 instead of the above diagram it is enough to consider the following one

$$\begin{array}{ccc} F(\mathfrak{A}) & \xrightarrow{F(f)} & F(\mathfrak{B}) \\ {\scriptstyle v(\mathfrak{A})}\downarrow & & \downarrow{\scriptstyle v(\mathfrak{B})} \\ \mathfrak{A} & \xrightarrow{f} & \mathfrak{B} \end{array}$$

In II.1 we have already seen that this diagram commutes. Thus $v\colon F\circ G \dot{\to} \mathrm{Id}_{\mathbf{C}_\alpha}$ is a natural isomorphism.

III. The definiability of F, G, θ, v by absolute set theoretic parameter free formulas follows from this property of F_{Ob} and θ established in I.1 (iii) and from the construction of F, G, θ, v by using F_{Ob} and θ.

The primitive recursiveness of the functions $F_{\mathrm{Ob}}, F_{\mathrm{Mor}}, G_{\mathrm{Ob}}, G_{\mathrm{Mor}}, \theta, v$ can be established analogously. □

The above theorem raises the question about the isomorphism of the categories under consideration. We show that isomorphism does occur, indeed.

Theorem 2.5. Let $\alpha \geqq \omega$. The categories $\mathbf{Lf}_\alpha$ and $\mathbf{C}_\alpha$ are isomorphic, i.e. $\mathbf{Lf}_\alpha \cong \mathbf{C}_\alpha$.

Proof. To prove the statement we construct an isomorphism $H\colon \mathbf{Lf}_\alpha \to \mathbf{C}_\alpha$, which is a one-one and onto functor, both on objects and on morphisms. For the construction of H first we define a covering of the category $\mathbf{Lf}_\alpha$ and then we define H on this covering such that the image of H covers the category $\mathbf{C}_\alpha$.

By Theorem 2.4 we have a one-one endofunctor $F\colon \mathbf{C}_\alpha \to \mathbf{Lf}_\alpha$ and a natural isomorphism $\theta\colon F \dot{\to} \mathrm{Id}_{\mathbf{Lf}_\alpha}$, which sends F into $\mathrm{Id}_{\mathbf{Lf}_\alpha}$.

(Note that here we use the fact provided by Lemma 2.4.1 that $G\colon \mathbf{Lf}_\alpha \to \mathbf{C}_\alpha$ is an identity functor.)

First we construct the covering of $\mathrm{Ob}\,\mathbf{Lf}_\alpha$ by induction as follows.

Take $L_0 \stackrel{\mathrm{d}}{=} \mathrm{Ob}\,\mathbf{Lf}_\alpha$.

We need the following notation. Let $\mathbf{A}$ be an arbitrary category and R be a functor on $\mathbf{A}$. Then for any subclass $S \subseteqq \mathrm{Ob}\,\mathbf{A}$ the R image of S is defined as follows

$$R^* S \stackrel{\mathrm{d}}{=} \{R_{\mathrm{Ob}}(\mathfrak{A}) : \mathfrak{A} \in S\}.$$

Take $K_0 \stackrel{\mathrm{d}}{=} \mathrm{Ob}\,\mathbf{C}_\alpha$. (It is evident that $K_0 \subseteqq L_0$.)

Furthermore let

$$L_1 \stackrel{\mathrm{d}}{=} F^* L_0 \quad \text{(Clearly } L_1 \subseteqq K_0.)$$

$$K_1 \stackrel{\mathrm{d}}{=} F^* K_0 \quad \text{(Since } K_0 \subseteqq L_0 \text{ we have } K_1 \subseteqq L_1.)$$

Let us suppose that the classes L_n and K_n have already been defined up to some n.

Then let

$$L_{n+1} \stackrel{\mathrm{d}}{=} F^* L_n \quad \text{and} \quad K_{n+1} \stackrel{\mathrm{d}}{=} F^* K_n.$$

Thus the classes L_n and K_n have been defined for any $n \in \omega$ by induction. They are illustrated by Fig 1.

For any $n \in \omega$ let $W_n \overset{d}{=} K_n \setminus L_{n+1}$ and let $W \overset{d}{=} \bigcup_{n \in \omega} W_n$.

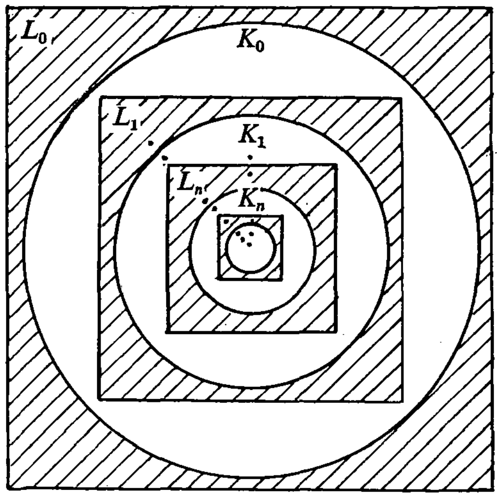


Fig. 1

Moreover let $D \overset{d}{=} L_0 \setminus W$ (note that $D = \bigcup_{n \in \omega} (L_n \setminus K_n)$).

On Fig. 1. the white area corresponds to W and the dark one to D.

It follows from the construction that $\mathrm{Ob}\,\mathbf{Lf}_\alpha$ is covered by the disjoint union of D and W, i.e. $\mathrm{Ob}\,\mathbf{Lf}_\alpha = D \cup W$.

Now we construct a covering to $\mathbf{C}_\alpha$ by giving a function $H_{\mathrm{Ob}} = \mathrm{Ob}\,\mathbf{Lf}_\alpha \rightarrowtail\!\!\!\twoheadrightarrow \mathrm{Ob}\,\mathbf{C}_\alpha$ as follows.

For any $\mathfrak{A} \in D$ let $H_{\mathrm{Ob}}(\mathfrak{A}) \overset{d}{=} F(\mathfrak{A})$ and for any $\mathfrak{B} \in W$ let $H_{\mathrm{ob}}(\mathfrak{B}) = \mathfrak{B}$, i.e. $H_{\mathrm{Ob}} = (F_{\mathrm{Ob}} \restriction D) \cup \mathrm{Id} \restriction W$. Clearly $H_{\mathrm{Ob}}\colon \mathrm{Ob}\,\mathbf{Lf}_\alpha \rightarrowtail\!\!\!\twoheadrightarrow \mathrm{Ob}\,\mathbf{C}_\alpha$ is one-one and onto $\mathrm{Ob}\,\mathbf{C}_\alpha$ since $\mathrm{Ob}\,\mathbf{Lf}_\alpha = L_0$ and $\mathrm{Ob}\,\mathbf{C}_\alpha = K_0$ that is $H_{\mathrm{Ob}}\colon L_0 \rightarrowtail\!\!\!\twoheadrightarrow K_0$. Note that $H_{\mathrm{Ob}} = F_{\mathrm{Ob}} \restriction D \cup G_{\mathrm{Ob}}^{-1} \restriction W$.

Now we define the mapping $H_{\mathrm{Mor}}\colon \mathrm{Mor}\,\mathbf{Lf}_\alpha \rightarrowtail\!\!\!\twoheadrightarrow \mathrm{Mor}\,\mathbf{C}_\alpha$. We distinguish four cases:

1. Let $\mathfrak{A}, \mathfrak{B} \in W$ and $f \in \mathrm{Hom}(\mathfrak{A}, \mathfrak{B})$. Then we define $H_{\mathrm{Mor}}(f) \overset{d}{=} f$.
2. Let $\mathfrak{A}, \mathfrak{B} \in D$ and $f \in \mathrm{Hom}(\mathfrak{A}, \mathfrak{B})$. Then we define $H_{\mathrm{Mor}}(f) \overset{d}{=} F(f)$.
3. Let $\mathfrak{A} \in D$, $\mathfrak{B} \in W$ and $f \in \mathrm{Hom}(\mathfrak{A}, \mathfrak{B})$.

Since $\theta\colon F \dashrightarrow \mathrm{Id}_{\mathbf{Lf}_\alpha}$ is a natural isomorphism we have $F(\mathfrak{B}) \overset{\theta(\mathfrak{B})}{\rightarrowtail\!\!\!\twoheadrightarrow} \mathfrak{B} = H(\mathfrak{B})$. Then $H(\mathfrak{A}) = F(\mathfrak{A}) \xrightarrow{F(f)} F(\mathfrak{B}) \xrightarrow{\theta(\mathfrak{B})} H(\mathfrak{B})$. We define $H_{\mathrm{Mor}}(f) \overset{d}{=} \theta(\mathfrak{B}) \circ F(f)$. It is evident that $H(f) \in \mathrm{Hom}(H(\mathfrak{A}), H(\mathfrak{B}))$.

4. Let $\mathfrak{A} \in D$, $\mathfrak{B} \in W$ and $f \in \mathrm{Hom}(\mathfrak{B}, \mathfrak{A})$. For this case we define $H_{\mathrm{Mor}}(f) \overset{d}{=} F(f) \circ [\theta(\mathfrak{B})]^{-1}$. By the above cases 1—4 the mapping $H_{\mathrm{Mor}}\colon \mathrm{Mor}\,\mathbf{Lf}_\alpha \to \mathrm{Mor}\,\mathbf{C}_\alpha$ is defined. Since by Theorem 2.4 the functor F is full, faithful and one-one, it is

easy to verify that the mapping H_{Mor} is onto and one-one such that for any $\mathfrak{A}, \mathfrak{B} \in \text{Ob } \mathbf{Lf}_\alpha$ and $f \in \text{Hom}(\mathfrak{A}, \mathfrak{B})$ we have $H_{Mor}(f) \in \text{Hom}(H_{Ob}(\mathfrak{A}), H_{Ob}(\mathfrak{B}))$. For illustration to H_{Mor} see Fig. 2.

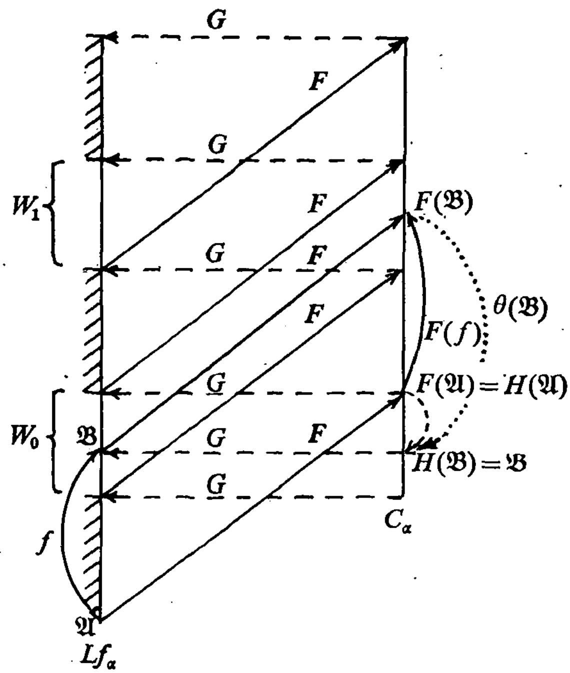


Fig. 2

Let $H \stackrel{d}{=} \langle H_{Ob}, H_{Mor} \rangle$. For the verification that H is a functor, properties (i)—(iii) displayed in Definition 2.9 should be established. The properties (i) and (iii) are satisfied by definition. Let $f \in \text{Hom}(\mathfrak{A}, \mathfrak{B})$ and $g \in \text{Hom}(\mathfrak{B}, \mathfrak{C})$. To verify property (ii) the following cases should be checked.

a) $\mathfrak{A}, \mathfrak{B}, \mathfrak{C} \in D$,

b) $\mathfrak{A}, \mathfrak{B}, \mathfrak{C} \in W$,

c) $\mathfrak{A}, \mathfrak{B} \in D, \quad \mathfrak{C} \in W$,

d) $\mathfrak{A}, \mathfrak{B} \in W, \quad \mathfrak{C} \in D$,

e) $\mathfrak{A} \in D, \quad \mathfrak{B}, \mathfrak{C} \in W$,

f) $\mathfrak{A} \in W, \quad \mathfrak{B}, \mathfrak{C} \in D$,

g) $\mathfrak{A}, \mathfrak{C} \in D, \quad \mathfrak{B} \in W$,

h) $\mathfrak{A}, \mathfrak{C} \in W, \quad \mathfrak{B} \in D$,

i) $\mathfrak{B}, \mathfrak{C} \in D, \quad \mathfrak{A} \in W$

and

j) $\mathfrak{B}, \mathfrak{C} \in W, \quad \mathfrak{A} \in D$.

From the above cases we check the most difficult ones, namely g) and j)

g) $\mathfrak{A}, \mathfrak{C} \in D, \quad \mathfrak{B} \in W$.

By using the corresponding definitions we have the following diagram

$$
\begin{array}{ccccc}
\mathfrak{C} & & H(\mathfrak{C}) = F(\mathfrak{C}) & = = = & F(\mathfrak{C}) \\
g \uparrow & & H(g) = F(g) \circ [\theta(\mathfrak{B})]^{-1} \uparrow & & \uparrow F(g) \\
\mathfrak{B} & & H(\mathfrak{B}) = \mathfrak{B} & \xleftarrow{\theta(\mathfrak{B})} & F(\mathfrak{B}) \\
f \uparrow & & H(f) = \theta(\mathfrak{B}) \circ F(f) \uparrow & & \uparrow F(f) \\
\mathfrak{A} & & H(\mathfrak{A}) = F(\mathfrak{A}) & = = = & F(\mathfrak{A})
\end{array}
$$

By using the fact that F is a functor, from the above diagram we have

$$
H(g) \circ H(f) = F(g) \circ [\theta(\mathfrak{B})]^{-1} \circ \theta(\mathfrak{B}) \circ F(f) =
$$

$$
= F(g) \circ \mathrm{Id}_{F(\mathfrak{B})} \circ F(f) = F(g) \circ F(f) = F(g \circ f).
$$

Hence, by definition, we get

$$
F(g \circ f) = H(g \circ f) \quad \text{since} \quad \mathfrak{A}, \mathfrak{C} \in D.
$$

j) $\mathfrak{A} \in D, \quad \mathfrak{B}, \mathfrak{C} \in W$.

By using the corresponding definitions we have the following diagram

$$
\begin{array}{ccccc}
\mathfrak{C} & & H(\mathfrak{C}) = \mathfrak{C} & \xleftarrow{\theta(\mathfrak{C})} & F(\mathfrak{C}) \\
g \uparrow & & H(g) = g \uparrow & & \uparrow F(g) \\
\mathfrak{B} & & H(\mathfrak{B}) = \mathfrak{B} & \xleftarrow{\theta(\mathfrak{B})} & F(\mathfrak{B}) \\
f \uparrow & & H(f) = \theta(\mathfrak{B}) \circ F(f) \uparrow & & \uparrow F(f) \\
\mathfrak{A} & & H(\mathfrak{A}) = F(\mathfrak{A}) & = = = & F(\mathfrak{A})
\end{array}
$$

By using the fact that θ is a natural transformation and that F is a functor we get from the above diagram

$$H(g)\circ H(f) = g\circ\theta(\mathfrak{B})\circ F(f) =$$
$$= \theta(\mathfrak{C})\circ F(g)\circ F(f) = \theta(\mathfrak{C})\circ F(g\circ f)$$

which, by definition, is $H(g\circ f)$, since $\mathfrak{A}\in D$ and $\mathfrak{C}\in W$.

Thus H is a functor and by its construction, H is one-one and onto and thus H establishes an isomorphic connection between the categories $\mathbf{Lf}_\alpha$ and $\mathbf{C}_\alpha$. □

Some questions w.r.t. the functor H arise. Namely, we have the following

OPEN PROBLEMS:

— Is there an absolute isomorphism M: $\mathbf{Lf}_\alpha \rightarrowtail\!\!\twoheadrightarrow \mathbf{C}_\alpha$?
— Is the functor H constructed in the above proof definable by a quantifier free formula in ZFC?
— Is the functor H primitive recursive in the sense of DEVLIN [8]?
— Is there any isomorphism I: $\mathbf{Lf}_\alpha \rightarrowtail\!\!\twoheadrightarrow \mathbf{C}_\alpha$ which is rudimentary in the sense of DEVLIN [8]?

3. Category of theories

Let α be an ordinal. Definition 2.6 provides the notion of theories of α variables. However without supposing further conditions two theories T_1 and T_2 can have e.g. different sets Ax_1 and Ax_2 but one of these sets might be derivated from the other one by the use of an appropriate calculus, i.e. by the use of pure syntactical transformations. I.e. despite of their differences in their presentations the theories are equivalent. To avoid such cases we slightly modify Definition 2.6.

Definition 3.1. Let α be a fixed ordinal. Let t be an arbitrary similarity type and $\mathrm{Ax}\subseteq F_t^\alpha$. Take $\mathrm{Ax}^* \overset{d}{=} \{\varphi: \mathrm{Ax}\models\varphi\}$.

The pair $\langle \mathrm{Ax}^*, F_t^\alpha\rangle$ is said to be a *saturated theory* of α variables. □

Further on when speaking about a theory we have a saturated one in mind.

In the case of saturated theories we often identify a theory $T=\langle \mathrm{Ax}, F_t^\alpha\rangle$ with the set Ax of axioms.

Now we define how a theory can be interpreted in an other one.

Definition 3.2. Let $T_1=\langle \mathrm{Ax}_1, F_{t_1}^\alpha\rangle$ and $T_2=\langle \mathrm{Ax}_2, F_{t_2}^\alpha\rangle$ be theories in α variables. Let $m: F_{t_1}^\alpha \to F_{t_2}^\alpha$.

The triple $\langle T_1, m, T_2\rangle$ is said to be an *interpretation* going from T_1 into T_2 iff the following conditions hold:

a) $m(x_i=x_j)=x_i=x_j$ for every $i, j<\alpha$;
b) $m(\varphi\wedge\psi)=m(\varphi)\wedge m(\psi)$, $m(\neg\varphi)=\neg m(\varphi)$;
$m(\exists x_i\varphi)=\exists x_i m(\varphi)$ for all $\varphi, \psi\in F_{t_1}^\alpha$, $i<\alpha$;
c) $\mathrm{Ax}_2\models m(\varphi)$ for all $\varphi\in F_{t_1}^\alpha$ such that $\mathrm{Ax}_1\models\varphi$.

We shall often say that m is an interpretation but in these cases we actually mean $\langle T_1, m, T_2\rangle$. By saying that $\langle T_1, m, T_2\rangle$ is an interpretation we mean that $\langle T_1, m, T_2\rangle$ is an interpretation of the theory T_1 in the theory T_2. □

Let m, n be two interpretations of T_1 in T_2.

The interpretations $\langle T_1, m, T_2\rangle$, $\langle T_1, n, T_2\rangle$ are defined to be *semantically equivalent,* in symbols $m\equiv n$, iff the following condition holds:

$$\mathrm{Ax}_2 \models (m(\varphi) \leftrightarrow n(\varphi)) \quad \text{for all } \varphi \in F_{t_1}^{\alpha}.$$

Let $\langle T_1, m, T_2\rangle$ be an interpretation. We define the equivalence class $m/\equiv$ of m or more precisely $\langle T_1, m, T_2\rangle /\equiv$ to be: $m/\equiv \stackrel{\mathrm{d}}{=} \{\langle T_1, n, T_2\rangle: n\equiv m$ and n is an interpretation of T_1 in $T_2\}$.

Now we are ready to define the connection between two theories T_1 and T_2.

Definition 3.3. Let T_1 and T_2 be two theories of α variables.

By a *theory morphism* μ: $T_1 \to T_2$ going from T_1 into T_2 we understand an equivalence class of interpretations of T_1 in T_2, i.e. μ is a theory morphism μ: $T_1 \to T_2$ iff $\mu = m/\equiv$ for some interpretation $\langle T_1, m, T_2\rangle$. □

Definition 3.4. (i) $\mathbf{TH}_\alpha$ is defined to be the quadruple $\mathbf{TH}_\alpha \stackrel{\mathrm{d}}{=} \langle \mathrm{Ob}\,\mathbf{TH}_\alpha, \mathrm{Mor}\,\mathbf{TH}_\alpha, \circ, \mathrm{Id}\rangle$, where the mappings $\circ$: $\mathrm{Mor}\,\mathbf{TH}_\alpha \times \mathrm{Mor}\,\mathbf{TH}_\alpha \to \mathrm{Mor}\,\mathbf{TH}_\alpha$ and Id: $\mathrm{Ob}\,\mathbf{TH}_\alpha \to \mathrm{Mor}\,\mathbf{TH}_\alpha$ are defined in (ii)—(iii) below and $\mathrm{Ob}\,\mathbf{TH}_\alpha \stackrel{\mathrm{d}}{=} \{T$: T is a saturated theory in α variables$\}$, $\mathrm{Mor}\,\mathbf{TH}_\alpha \stackrel{\mathrm{d}}{=} \{\langle T_1, \mu, T_2\rangle$: μ is a theory morphism μ: $T_1 \to T_2$ and $T_1, T_2 \in \mathrm{Ob}\,\mathbf{TH}_\alpha\}$.

(ii) Let μ: $T_1 \to T_2$ and ν: $T_2 \to T_3$ be two theory morphisms. We define the *composition* $\nu\circ\mu$: $T_1 \to T_3$ to be the unique theory morphism for which there exists $m \in \mu$ and $n \in \nu$ such that $\nu\circ\mu = (n\circ m)/\equiv$, where the function $(n\circ m)$: $F_{t_1}^{\alpha} \to F_{t_3}^{\alpha}$ is defined by $(n\circ m)(\varphi) = n(m(\varphi))$ for all $\varphi \in F_{t_1}^{\alpha}$.

(iii) Let $T = \langle \mathrm{Ax}, F_t^{\alpha}\rangle$ be a theory. The identity function $\mathrm{Id}_{F_t^{\alpha}}$ is defined to be $\mathrm{Id}_{F_t^{\alpha}} \stackrel{\mathrm{d}}{=} \{\langle \varphi, \varphi\rangle : \varphi \in F_t^{\alpha}\}$.

The *identity morphism* Id_T on T is defined to be $\mathrm{Id}_T \stackrel{\mathrm{d}}{=} (\mathrm{Id}_{F_t^{\alpha}})/\equiv$. □

Proposition 3.1. $\mathbf{TH}_\alpha$ is a category.

Proof. The statement follows from the two properties bellow:

a) the composition defined in (ii) of Definition 3.4 is associative, i.e. let μ_1: $T_1 \to T_2$, μ_2: $T_2 \to T_3$ and μ_3: $T_3 \to T_4$ be theory morphisms and let $m_i \in \mu_i$ for $i \in \{1, 2, 3\}$. By associativity of composition of ordinary mappings $m_3\circ m_2\circ m_1 \in \mu_3\circ(\mu_2\circ\mu_1)$ and $m_3\circ m_2\circ m_1 \in (\mu_3\circ\mu_2)\circ\mu_1$ proving $\mu_3\circ\mu_2\circ\mu_1 = (m_3\circ m_2\circ m_1)/\equiv = (\mu_3\circ\mu_2)\circ\mu_1$;

b) the identity morphism is Id_T defined by (iii) of Definition 3.4. Let μ: $T_1 \to T_2$, then for some $m \in \mu$, $m\circ\mathrm{Id}_{T_1}(\varphi) = m(\mathrm{Id}_{T_1}(\varphi)) = m(\varphi) = \mathrm{Id}_{T_2}\, m(\varphi)$, for any $\varphi \in F_t^{\alpha}$, i.e. $\mu\circ\mathrm{Id}_{T_1} = \mathrm{Id}_{T_2}\circ\mu = \mu$. □

The main properties of the category $\mathbf{TH}_\alpha$ are investigated in AGN [4]. Here we show how the category of theories can be characterized algebraically.

Theorem 3.2. The categories $\mathbf{C}_\alpha$ and $\mathbf{TH}_\alpha$ are isomorphic.

Proof. First we define a functor F: $\mathbf{TH}_\alpha \to \mathbf{C}_\alpha$.

a) We define the object part F_{Ob}: $\mathrm{Ob}\,\mathbf{TH}_\alpha \to \mathrm{Ob}\,\mathbf{C}_\alpha$ of F as follows. Let $T = \langle \mathrm{Ax}, F_t^{\alpha}\rangle \in \mathrm{Ob}\,\mathbf{TH}_\alpha$ be arbitrary. Recall that in Definition 2.8 the concept algebra $\mathfrak{C}_T$ of the theory T was defined to be $\mathfrak{F}_t/\equiv_T$ that is $\mathfrak{C}_T=\mathfrak{F}_T^\alpha/\{\langle\varphi,\psi\rangle: \mathrm{Ax}\models\models(\varphi\leftrightarrow\psi)\}$. We define $F(T)\stackrel{d}{=}F_{\mathrm{Ob}}(T)\stackrel{d}{=}\mathfrak{C}_T$ for every $T\in\mathrm{Ob}\,\mathbf{TH}_\alpha$. By this the function F_{Ob}: $\mathrm{Ob}\,\mathbf{TH}_\alpha\to\mathrm{Ob}\,\mathbf{C}_\alpha$ is defined.

b) Let μ: $T_1\to T_2\in\mathrm{Mor}\,\mathbf{TH}_\alpha$.

We define $F_{\mathrm{Mor}}(\mu)\stackrel{d}{=}\{\langle x, y\rangle\in C_{T_1}\times C_{T_2}$: there exist a $\varphi\in x$ and an $m\in\mu$ such that $m(\varphi)\in y\}$.

It is not hard to check that $F_{\mathrm{Mor}}(\mu)$: $\mathfrak{C}_{T_1}\to\mathfrak{C}_{T_2}$ is a function, and, by Definition 2.9, it follows that $F_{\mathrm{Mor}}(\mu)\in\mathrm{Hom}\,(\mathfrak{C}_{T_1},\mathfrak{C}_{T_2})=\mathrm{Hom}\,(F(T_1), F(T_2))\subsetneq\mathrm{Mor}\,\mathbf{C}_\alpha$, i.e. $F_{\mathrm{Mor}}(\mu)$ is a homomorphism.

c) We have defined a function F_{Mor}: $\mathrm{Mor}\,\mathbf{TH}_\alpha\to\mathrm{Mor}\,\mathbf{C}_\alpha$. Let $F\stackrel{d}{=}\langle F_{\mathrm{Ob}}, F_{\mathrm{Mor}}\rangle$. Now we prove that F is a functor. F_{Mor} satisfies the following properties:

(i) for any $T\in\mathrm{Ob}\,\mathbf{TH}_\alpha$, $F_{\mathrm{Mor}}\,(\mathrm{Id}_T)=\mathrm{Id}_{\mathfrak{C}_T}$,

(ii) let μ_1: $T_1\to T_2$ and μ_2: $T_2\to T_3$. Then $F_{\mathrm{Mor}}(\mu_2\circ\mu_1)(\varphi)=F_{\mathrm{Mor}}(m\circ n)/\equiv_{T_2}(\varphi)==n(m(\varphi))/\equiv_{T_2}=F_{\mathrm{Mor}}(\mu_2)(m(\varphi)/\equiv_{T_2})=F_{\mathrm{Mor}}(\mu_2)\circ F_{\mathrm{Mor}}(\mu_1)$ for any $\varphi\in F_t^\alpha$. Here $m\in\mu_1$ and $n\in\mu_2$.

Thus the pair of functions $F=\langle F_{\mathrm{Ob}}, F_{\mathrm{Mor}}\rangle$ is a functor F: $\mathbf{TH}_\alpha\to\mathbf{C}_\alpha$.

Next we prove that F_{Ob}: $\mathrm{Ob}\,\mathbf{TH}_\alpha\rightarrowtail\!\!\twoheadrightarrow\mathrm{Ob}\,\mathbf{C}_\alpha$ is a set theoretic isomorphism, that is F_{Ob} is one-one and onto.

(i) Let $T_i=\langle\mathrm{Ax}_i, F_{t_i}^\alpha\rangle\in\mathrm{Ob}\,\mathbf{TH}_\alpha$ for $i\in\{1,2\}$. Assume $T_1\neq T_2$.

Case 1. $t_1\neq t_2$. Then $F(T_1)\neq F(T_2)$ since $\bigcup C_{T_1}=F_{t_1}^\alpha\neq F_{t_2}^\alpha=\bigcup C_{T_2}$.

Case 2. $t_1=t_2$. Then $\mathrm{Ax}_1\neq\mathrm{Ax}_2$. Recall that by the definition of $\mathbf{TH}_\alpha$ we have $\mathrm{Ax}_i=\mathrm{Ax}_i^*$ for $i\in\{1,2\}$. Thus $1^{F(T_1)}=\mathrm{Ax}_1^*=\mathrm{Ax}_1\neq\mathrm{Ax}_2=\mathrm{Ax}_2^*=1^{F(T_2)}$.

Cases 1—2 prove $F(T_1)\neq F(T_2)$ and hence F_{Ob}: $\mathrm{Ob}\,\mathbf{TH}_\alpha\rightarrowtail\mathrm{Ob}\,\mathbf{C}_\alpha$ is proved to be one-one.

(ii) Let $\mathfrak{A}\in\mathrm{Ob}\,\mathbf{C}_\alpha$ be arbitrary. By the definition of $\mathbf{C}_\alpha$ then there exists a theory $T=\langle\mathrm{Ax}, F_t^\alpha\rangle$ such that $\mathfrak{A}\cong\mathfrak{C}_T$. Let $T^*=\langle\mathrm{Ax}^*, F_t^\alpha\rangle$. Clearly $T^*\in\mathrm{Ob}\,\mathbf{TH}_\alpha$ and $F(T^*)=\mathfrak{C}_{T^*}=\mathfrak{C}_T=\mathfrak{A}$.

We proved that $\mathrm{Rg}\,F_{\mathrm{Ob}}=\mathrm{Ob}\,\mathbf{C}_\alpha$ and hence F_{Ob}: $\mathrm{Ob}\,\mathbf{TH}_\alpha\rightarrowtail\!\!\twoheadrightarrow\mathrm{Ob}\,\mathbf{C}_\alpha$ is proved to be a set theoretic isomorphism.

Next we prove that F_{Mor} is a set theoretic isomorphism on the Hom-sets.

Let $T_i=\langle\mathrm{Ax}_i, F_{t_i}^\alpha\rangle\in\mathrm{Ob}\,\mathbf{TH}_\alpha$ for $i\in\{1,2\}$.

(i) Let μ: $T_1\to T_2$ and ν: $T_1\to T_2$ be different, i.e. $\mu\neq\nu$. Then $(\exists m\in\mu)(\exists n\in\nu)$ $(\exists\varphi\in F_{t_1}^\alpha)$ $\mathrm{Ax}_2\not\models(m(\varphi)\leftrightarrow n(\varphi))$. Let these m, n, φ be fixed. Then

$$F(\mu)(\varphi/\equiv_{T_1})=m(\varphi)/\equiv_{T_2}\neq n(\varphi)/\equiv_{T_2}=F(\mu)(\varphi/\equiv_{T_2}).$$

Thus F_{Mor} is *one-one*.

(ii) Let $f\in\mathrm{Hom}\,(F(T_1), F(T_2))$ be an arbitrary homomorphism from the algebra $\mathfrak{C}_{T_1}$ into the algebra $\mathfrak{C}_{T_2}$. Let $\mathrm{At}\subsetneq F_{t_1}^\alpha$ be the set of all atomic formulas in $F_{t_1}^\alpha$ not involving equality, i.e. $\mathrm{At}\stackrel{d}{=}\{R(x_{i_1},\dots,x_{i_n})$: $R\in\mathrm{Do}\,t_1$ and $t_1(R)=n$ and $i_1,\dots,i_n\in\alpha\}$. Note that $(x_i=x_j)\notin\mathrm{At}$ for any $i, j\in\alpha$.

For every $i\in\{1,2\}$ we define the homomorphism nat_i: $\mathfrak{F}_{t_i}\twoheadrightarrow\mathfrak{F}_{t_i/\equiv_{T_i}}$ as follows $\mathrm{nat}_i\,(\varphi)\stackrel{d}{=}\varphi/\equiv_{T_i}$ for each $\varphi\in F_{t_i}^\alpha$.

Let c: $F_{t_2}^\alpha/\equiv_{T_2}\to F_{t_2}^\alpha$ be a choice function that is $\mathrm{nat}_2\circ c=\mathrm{Id}_{C_{T_2}}$. Let $n\stackrel{d}{=}\stackrel{d}{=}(c\circ f\circ\mathrm{nat}_1)\restriction\mathrm{At}$. Then n: $\mathrm{At}\to F_{t_2}$ is such that $\mathrm{nat}_2\circ n=(f\circ\mathrm{nat}_1)\restriction\mathrm{At}$.

Since At freely generates the algebra $\mathfrak{F}_{t_1}^\alpha$ there is a unique homomorphic extension $m\colon \mathfrak{F}_{t_1}^\alpha \to \mathfrak{F}_{t_2}^\alpha$ of n to the algebra $\mathfrak{F}_{t_1}$, i.e. $m{\restriction}\mathrm{At}=n$. The diagram

$$\begin{array}{ccc} \mathfrak{F}_{t_1} & \xrightarrow{\;m\;} & \mathfrak{F}_{t_2} \\ {\scriptstyle \mathrm{nat}_1}\downarrow & & \downarrow{\scriptstyle \mathrm{nat}_2} \\ \mathfrak{C}_{t_1} & \xrightarrow{\;f\;} & \mathfrak{C}_{T_2} \end{array}$$

commutes since $f \circ \mathrm{nat}_1{\restriction}\mathrm{At} = \mathrm{nat}_2 \circ n = \mathrm{nat}_2 \circ m{\restriction}\mathrm{At}$ and At generates $\mathfrak{F}_{t_1}$.

Assume $\mathrm{Ax}_1 \models \varphi$. Then $f(\mathrm{nat}_1(\varphi)) = 1^{F(T_2)} = \mathrm{nat}_2(m(\varphi))$ and hence $m(\varphi) \in \mathrm{Ax}_2^* = \mathrm{nat}_2(1^{\mathfrak{F}_{t_2}^\alpha}) = \mathrm{nat}_2\,(x_0 = x_0)$.

This proves that $\langle T_1, m, T_2\rangle$ is an interpretation and hence $m/{\equiv}\colon T_1 \to T_2 \in \mathrm{Mor}\,\mathbf{TH}_\alpha$.

By the definition of F_{Mor} we have $F(m/{\equiv}) = f$. We have proved that $\mathrm{Rg}\, F_{\mathrm{Mor}} = \mathrm{Mor}\,\mathbf{C}_\alpha$. Then by the above considerations $F\colon \mathbf{TH}_\alpha \rightarrowtail\!\!\!\twoheadrightarrow \mathbf{C}_\alpha$ is an isomorphism proving $\mathbf{TH}_\alpha \cong \mathbf{C}_\alpha$. □

From Theorems 2.5 and 3.2 we have the following representation theorem.

Theorem 3.3. The categories $\mathbf{Lf}_\alpha$ and $\mathbf{TH}_\alpha$ are isomorphic. □

By the representation theorem (Theorem 3.3) we can investigate the category $\mathbf{TH}_\alpha$ through the investigation of the properties of the category $\mathbf{Lf}_\alpha$, since $\mathbf{TH}_\alpha \cong \mathbf{Lf}_\alpha$.

Before using this possibility we recall some well known notions.

By a small category we understand a category $\mathbf{C} = \langle \mathrm{Ob}\,\mathbf{C}, \mathrm{Mor}\,\mathbf{C}\rangle$ such that $\mathrm{Mor}\,\mathbf{C}$ is a set.

Definition 3.5. Let $\mathbf{K}$ be an arbitrary category. By a diagram in the category $\mathbf{K}$ we undestand a functor $D\colon \mathbf{C} \to \mathbf{K}$, where $\mathbf{C}$ is a small category.

The category $\mathbf{C}$ is called the *index* category of the diagram D.

Definition 3.6. Let $\mathbf{K}$ be an arbitrary category and let $D\colon \mathbf{I} \to \mathbf{K}$ be a diagram. Let $\mathbf{I} = \langle I, M\rangle$.

(i) A *cone* over D is a system $\langle H, \langle h_i\colon i \in I\rangle\rangle$ such that $H \in \mathrm{Ob}\,\mathbf{K}$ and for each $i \in I$, $h_i\colon H \to D(i) \in \mathrm{Mor}\,\mathbf{K}$ and for every $f \in M$ if $f\colon i \to j$ in $\mathbf{I}$ then $D(f) \circ h_i = h_j$ in $\mathbf{K}$.

(ii) The *limit* of D in $\mathbf{K}$ is a cone $\langle G, \langle g_i\colon i \in I\rangle\rangle$ over D such that for every cone $\langle H, \langle h_i\colon i \in I\rangle\rangle$ over D there is a unique morphism $\mu\colon H \to G$ such that for any $i \in I$, $h_i \circ \mu = g_i$.

(iii) The *colimit* of D is defined exactly as above but all the arrows are reversed. Thus a colimit is a cocone $\langle\langle g_i\colon i \in I\rangle, G\rangle$ with $g_i\colon D(i) \to G$ etc.

Definition 3.7. A category $\mathbf{K}$ is said to be complete and cocomplete if for every diagram D in $\mathbf{K}$ both the limit and the colimit of D exist in $\mathbf{K}$.

Theorem 3.4. The category $\mathbf{TH}_\alpha$ is complete and cocomplete if $\alpha \geqq \omega$.

Proof. Since $\mathbf{TH}_\alpha \cong \mathbf{Lf}_\alpha$ by Theorem 3.3 it is enough to prove that $\mathbf{Lf}_\alpha$ is complete and cocomplete. Let $\mathrm{Re}_\alpha \stackrel{\mathrm{d}}{=} \mathbf{HSP}\,\mathrm{Lf}_\alpha$, that is $\mathrm{Re}_\alpha \subseteq \mathrm{Alg}\,(l_\alpha)$ is the smallest variety containing Lf_α. Let $\mathbf{Re}_\alpha$ be the full subcategory of $\mathbf{Alg}\,(l_\alpha)$ with Ob $\mathbf{Re}_\alpha = \mathrm{Re}_\alpha$. Then $\mathbf{Lf}_\alpha$ is a full subcategory of $\mathbf{Re}_\alpha$. It is well known that any variety is complete and cocomplete, see e.g. Proposition III.5.11 of TSALENKO—SHULGEIFER [18]. Let $D\colon \mathbf{I} \to \mathbf{Lf}_\alpha$ be a diagram in $\mathbf{Lf}_\alpha$. Let $\langle \mathfrak{A}, \langle h_i\colon i\in I\rangle\rangle$ be the limit of D in $\mathbf{Re}_\alpha$. It is easy to prove (see e.g. Corollary 2.1.6 of HMT [11]) that the greatest Lf_α-subalgebra $\mathfrak{B}$ of $\mathfrak{A}$ exists, that is $\mathfrak{A} \supseteq \mathfrak{B} \in \mathrm{Lf}_\alpha$ and for every $\mathfrak{C} \in \mathrm{Lf}_\alpha$ such that $\mathfrak{C} \subseteq \mathfrak{A}$ then $\mathfrak{C} \subseteq \mathfrak{B}$. In other words $\mathfrak{B}$ is the greatest member ot $\mathrm{Lf}_\alpha \cap \mathrm{S}\mathfrak{A}$, where $\mathrm{S}\mathfrak{A}$ is the set of all subalgebras of $\mathfrak{A}$ and $\mathfrak{B} \subseteq \mathfrak{A}$ denotes that $\mathfrak{B}$ is a subalgebra of $\mathfrak{A}$. It is easy to check that $\langle \mathfrak{B}, \langle h_i{\restriction} B\colon i\in I\rangle\rangle$ is the limit of D in $\mathbf{Lf}_\alpha$.

Let $\langle h_i\colon i\in I, \mathfrak{A}\rangle$ be the colimit of D in $\mathbf{Re}_\alpha$. We prove that it is also the colimit of D in $\mathbf{Lf}_\alpha$. To this end it is enough to prove that $\mathfrak{A} \in \mathrm{Lf}_\alpha$. Let $X = \bigcup\{\mathrm{Rg}\, h_i\colon i\in I\}$. Then $X \subseteq A$, X generates $\mathfrak{A}$ and $(\forall y\in X)|\Delta y| < \omega$ since y is the homomorphic image of some $z \in D(i) \in \mathrm{Lf}_\alpha$. Then $\mathfrak{A} \in \mathrm{Lf}_\alpha$ by Theorem 2.1.5 in HMT [11]. □

We proved that $\mathbf{Lf}_\alpha$ is complete and cocomplete, moreover, we proved that $\mathbf{Lf}_\alpha$ is cocomplete in $\mathbf{Re}_\alpha$, that is the colimits of diagrams $D\colon \mathbf{I} \to \mathbf{Lf}_\alpha$ when computed in $\mathbf{Re}_\alpha$ coincide with those when computed in $\mathbf{Lf}_\alpha$. As a contrast we recall the following from GERGELY [10]. *$\mathbf{Lf}_\alpha$ is not cocomplete in $\mathbf{Alg}\,(l_\alpha)$, moreover, $\mathbf{Lf}_\alpha$ is not* cocomplete in $\mathbf{Bo}_\alpha$ as $\mathrm{Bo}_\alpha \subseteq \mathrm{Alg}\,(l_\alpha)$ was defined in HMT [11], neither is it cocomplete in the variety I Crs_α as defined in HMTAN [12] as these are proved in GERGELY [10]. I $\mathrm{Crs}_\alpha = \mathrm{HSP}\,\mathrm{Crs}_\alpha \supseteq \mathrm{Lf}_\alpha$ was proved in NÉMETI [16].

4. Conclusion

Here analysing the connection between the categories $\mathbf{TH}_\alpha$ and $\mathbf{CA}_\alpha$ only the theories were represented by cylindric algebras. However having a theory $T \subseteq F_t^\alpha$ not only the representation of T but that of the models $\mathfrak{A} \in \mathrm{Mod}\, T$ of the theory T, or that of the subclasses $K \subseteq \mathrm{Mod}\, T$ of the models can be done by the use of CA's. E.g. in NÉMETI [16], classes of models were represented by the use of the tools introduced in AGN [2] but from the point of view of the categories presently introduced only the objects were considered. Thus, for the entire investigation, morphisms should be considered as well. This investigation will be done elsewhere.

On the whole the present paper emphasizes the usefulness of certain universal algebraic tools to handle the category of all theories of α variables.

Thus all results concerning the subclass Lf_α of l_α-type algebras can be used directly to investigate language hierarchies.

This provides the possibility to represent and analyse formal semantics of language hierarchies by the use of a very important subclass of l_α-type cylindric algebras the so called locally independently-finite cylindric algebras, introduced in AGN [1]. These algebras were later called regular in HMTAN [12]. At the same time the established connection provides quite a concrete content to the notion of Lf_α which was introduced in HMT [11].

Theorem 3.3 provides an opposite possibility as well, namely, to establish some new results about Lf_α by using the tools of Category Theory.

RESEARCH INSTITUTE FOR
APPLIED COMPUTER SCIENCE
CSALOGÁNY U. 30—32.
BUDAPEST, HUNGARY
H—1536

References

[1] AGN, (ANDRÉKA, H., GERGELY, T., NÉMETI, I.): Toward a general theory of logics, Part 1, KFKI—73—67, Budapest, 1973.
[2] AGN: On universal algebraic construction of logics, *Studia Logica,* v. 36, 1977, pp. 9—47.
[3] AGN: Model theoretic semantics for many purpose languages and language hierarchies, Proceedings of COLING'80, Tokyo, 1980.
[4] AGN: Investigation in the language hierarchies, SZÁMKI-Working paper, Budapest, 1981.
[5] BLUM, E. K., D. R. ESTES, A generalization of the homomorphism concept, *Algebra Universalis,* v. 7, 1977, pp. 143—161.
[6] BURSTALL, R. M., J. A. GOGUEN, Putting theories together to make specifications, Proceedings of 5th IJCAI, Massachusets, 1977, pp. 1045—1058.
[7] BURSTALL, R. M., J. A. GOGUEN, The semantics of CLEAR, a specification language, D. BJØRNER, ed., *Abstract software specifications,* Lecture Notes in Computer Science, v. 86, Springer-Verlag, 1980, pp. 292—332.
[8] DEVLIN, K. J., *Aspects of constructibility,* Lecture Notes in Mathematics, v. 354, Springer-Verlag, Berlin, 1973.
[9] DÖMÖLKI, B., An example of hierarchical program specification, D. BJØRNER, ed., *Abstract software specifications,* Lecture Notes in Computer Science, v. 86, Springer-Verlag, 1980, pp. 333—353.
[10] GERGELY, T., Mono-coreflexivity of cylindric algebras, SZÁMKI-Working paper, Budapest, 1981.
[11] HMT (HENKIN, L., MONK, J. D., TARSKI, A.): *Cylindric algebras,* Part I, North Holland, Amsterdam, 1971.
[12] HMTAN (HENKIN, L., MONK, J. D., TARSKI, A., ANDRÉKA, H,, NÉMETI, I.): *Cylindric set algebras,* Lecture Notes in Mathematics, v. 883, Springer-Verlag, 1981.
[13] HUPBACH, U. L., Abstract implementation of abstract data type, P. DEMBINSK, ed., *Mathematical foundations of computer science,* Lecture Notes in Computer Science, v. 88, Springer-Verlag, 1980.
[14] MAC LANE, S., *Categories for the working mathematicians,* Springer-Verlag, N. Y., 1971.
[15] MONK, J. D., *Mathematical logic,* Springer-Verlag, New-York, 1976.
[16] NÉMETI, I., Connections between cylindric algebras and initial algebra semantics of CF languages, DÖMÖLKI, B., GERGELY, T. eds., *Mathematical logic in computer science,* North-Holland, 1981, pp. 561—606.
[17] RATTRAY, C. M. I., T. RUS, Task-hierarchy, a mathematical device for computer system modelling, Proceeding of First International Symposium on Mathematical Modelling, Missouri, 1977, pp. 1—15.
[18] TSALENKO, M. S., E. G. SHULGEĬFER, Osnovȳ teorii kategoriĭ, Nauka, Moskva, 1974.
[19] WINKOWSKI, J., Towards an understanding of computer simulation, *Fund. Inform.,* v. 1, 1978, pp. 277—289.

(Received April 27, 1981)

Многомерные предельные теоремы для времени ожидания в приоритетных системах $\vec{M}_r|\vec{G}_r|1|\infty$

Э. А. Даниелян, Р. Н. Читчян

1°. Введение. По мнению известного специалиста в области применений методов теории массового обслуживания в вычислительных системах Л. Клейнрока [1] «применения теории массового обслуживания для анализа распределения ресурсов и решения задач о потоках данных в вычислительных системах является, по-видимому, единственным доступным специалистам по вычислительной технике методом, который позволяет понять сложные связи в таких системах».

Современные ЭВМ работают в различных режимах (пакетная обработка, разделение времени процессора и т. д.) и в любом из них ряд способов обработки программ формализуются в математических приоритетных моделях.

Даже в одной из первых немарковских приоритетных моделей $\vec{M}_r|\vec{G}_r|1|\infty$ с относительным и абсолютным приоритетом, формализующей прохождение программ r типов на однопроцессорной ЭВМ, точный анализ такой характеристики, как время реакции процессора на программы разных типов, достаточно сложен и приводит к громоздким результатам. В то же время при загрузке процессора, близкой к единице, возникает ситуация сколь угодно длительного ожидания программ начала своего счета. Анализ такой ситуации проводится асимптотическими методами и приводит к доказательству предельных теорем в условиях так называемой «критической загрузки».

Настоящая работа посвящена изучению совместного стационарного распределения времен реакции программ разных типов в условиях «критической загрузки» для моделей с относительным и абсолютным приоритетом.

2°. Описание системы и вспомогательные результаты. В однолинейную систему массового обслуживания поступают независимые пуассоновские потоки 1-вызовов, ..., r-вызовов с параметрами $a_1>0, \ldots, a_r>0$ соответственно.

Длительности обслуживания вызовов независимы в совокупности, не зависят от процесса поступления и для i-вызовов имеют функцию распределения (ф. р.) $B_i(t)$, $B_i(+0)=0$ $(i=\overline{1, r})$.

Между вызовами разных потоков установлены приоритеты. Это означает, что при выборе очередного вызова на обслуживание на освободившийся прибор поступает один из вызовов наивысшего приоритета, имеющихся в очереди. Чем меньше индекс потока, тем выше его приоритет.

Внутри каждого приоритетного класса принята дисциплина обслуживания FIFO («первым пришел-первым обслужен»).

Рассматриваются система с относительным приоритетом (схема А) и системы с абсолютным приоритетом (схемы В). В схеме А вызовы обслуживаются без прерываний, а в схемах В обслуживание вызова прерывается поступившим вызовом высшего приоритета, который сразу же начинает обслуживаться. Прерванный вызов либо теряется (схема В2), либо вновь становится в очередь и при новом поступлении на прибор либо дообслуживается (схема В1), либо обслуживается заново (схема В3).

Предполагается, что в начальный момент система свободна от вызовов.

Пусть ϱ_{i1} $(i=\overline{1,r})$-загрузка системы 1-вызовами, ..., i-вызовами. Значения ϱ_{i1}, а также констант ϱ_{i2} $(i=\overline{1,r})$, имеющих смысл второго момента суммарного времени, затрачиваемого на обслуживание поступающих в среднем за единицу времени 1-вызовов, ..., i-вызовов, приведены в [2].

Там же введены понятия i-периода ii-периода, i-цикла и соответствующих им преобразований Лапласа—Стилтьеса (п. Л.—С.) $\pi_i(s)$, $\pi_{ii}(s)$ и $h_i(s)$ $(i=\overline{1,r})$.

Для всех рассматриваемых схем положим $(\mathrm{Re}\, s \geqq 0;\ i=\overline{1,r})$

$$\mu_1(s)=y_0(s)=v_0(s)=s,\quad \mu_{i+1}(s)=s+\sigma_i-\sigma_i\pi_i(s),$$

$$y_i(s)=s+a_i-a_i\pi_{ii}(s),\quad v_i(s)=s-a_i+a_ih_i(s).$$

В работе [3] доказана следующая

Лемма 1. *Если* $\varrho_{r1}\leqq 1$, то $(\mathrm{Re}\, s\geqq 0;\ i=\overline{1,r})$

$$\mu_{i+1}(s)=\mu_i(y_i(s)), \tag{2.1}$$

$$y_i(v_i(s))=v_i(y_i(s))=s, \tag{2.2}$$

где в схемах А и В1

$$\mu_{i+1}(s)=s+\sigma_i-\sum_{k=1}^{i}a_k\,\beta_k(\mu_{i+1}(s)), \tag{2.3}$$

$$v_i(s)=s-a_i+a_i\,b_i(\mu_i(s)), \tag{2.4}$$

в схеме В2

$$v_i(s)=s-\mu_i(s)\,a_i\,b_i(s+\sigma_{i-1}), \tag{2.5}$$

в схеме В3

$$v_i(s)=s-\mu_i(s)\cdot\frac{a_i\,b_i(s+\sigma_{i-1})}{\mu_i(s)\,b_i(s+\sigma_i)+\beta_i(s+\sigma_{i-1})}, \tag{2.6}$$

а $b_i(s)=[1-\beta_i(s)]\cdot s^{-1}$.

Далее, пусть $w_i(t)$ $(i=\overline{1,r})$-виртуальное время ожидания i-вызова в момент t и

$$W(x^{(r)})=\lim_{t\to+\infty}P\{w_i(t)<x_i\,(i=\overline{1,r})\},$$

$$w(s^{(r)})=\int\limits_{x^{(r)}=0^{(r)}}^{\infty^{(r)}}\exp\{-(s^{(r)},x^{(r)})\}d_{x^{(r)}}W(x^{(r)}),$$

где приняты следующие обозначения:

$$x^{(r)}=(x_1,\ldots,x_r),\quad s^{(r)}=(s_1,\ldots,s_r),\quad 0^{(r)}=(\underbrace{0,\ldots,0}_{r}),$$
$$\infty^{(r)}=(\underbrace{\infty,\cdots,\infty}_{r}),\quad d_{x^{(r)}}=dx_1\ldots dx_r$$

$(s^{(r)}, x^{(r)})$-скалярное произведение векторов $s^{(r)}$ и $x^{(r)}$.

Обозначим $(i=\overline{1,r})$: $s=s^{(r)}$

$$E_i(s)=a_i\frac{\mu_{i+1}(\theta_{i+1})-\mu_i(\theta_i)}{\theta_{i+1}-\nu_i(\theta_i)},\tag{2.7}$$

$$H_i(s)=\frac{\beta_i(\eta_1)-\beta_i(\theta_1)}{\eta_1-\theta_1},\tag{2.8}$$

$$V_k^{(i)}(s)=a_k\frac{\beta_i(\mu_k(\theta_k))-\beta_i(\mu_{k+1}(\theta_{k+1}))}{\theta_{k+1}-\nu_k(\theta_k)},\quad (k=\overline{1,i-1})\tag{2.9}$$

$$T_i(s)=\frac{\mu_{i+1}(\theta_{i+1})-\mu_i(\theta_i)}{\mu_i(\theta_i)-\mu_i(\eta_i)},\tag{2.10}$$

$$M_i(s)=[\delta_{\nu2}+\delta_{\nu3}h_i(\theta_i)\beta_i^{-1}(\sigma_{i-1}+\theta_i)](\delta_{\nu2}+\delta_{\nu3}h_i(\eta_i))b_i(\sigma_{i-1}+\theta_i),\tag{2.11}$$

$$\begin{aligned}N_i(s)=(\theta_i-\eta_i)^{-1}[\delta_{\nu2}+\delta_{\nu3}h_i(\theta_i)\beta_i^{-1}(\sigma_{i-1}+\theta_i)]\cdot[h_i(\eta_i)-\beta_i(\sigma_{i-1}+\theta_i)-\\-(\delta_{\nu2}+\delta_{\nu3}h_i(\eta_i))\cdot\sigma_{i-1}\pi_{i-1}(\eta_i)\cdot b_i(\sigma_{i-1}+\theta_i)],\end{aligned}\tag{2.12}$$

где индекс ν указывает на номер схемы В, δ_{ij}-символ Кронекера, а величины $\eta_i=\eta_i(s_i,\ldots,s_r)$, $\theta_i=\theta_i(s_i,\ldots,s_r)$ определяются рекуррентным образом

$$\begin{aligned}\eta_{r+1}&=\theta_{r+1}=0,\\ \eta_i&=s_i+y_i(\eta_{i+1})\quad (i=\overline{1,r}),\\ \theta_i&=\eta_i-\nu_i(\eta_i)+\theta_{i+1}\quad (i=\overline{1,r}).\end{aligned}\tag{2.13}$$

Обозначим через w_i $(i=\overline{1,r})$ стационарное время ожидания начала обслуживания i-вызова.

Имеет место следующая (см. [3])

Теорема 1. При выполнении условия $\varrho_{r1}<1$

$$\omega(s)=M\exp\{-(w,s)\}=\varrho_r(1+\Phi(s)),\tag{2.14}$$

где $\Phi(s)$ определяется соотношениями:

схема А

$$\Phi(s)=\sum_{i=1}^{r}\Psi_i(s)\cdot E_i(s),\tag{2.15}$$

$$\Psi_i(s)=H_i(s)+\sum_{k=1}^{i-1}\Psi_k(s)V_k^{(i)}(s),\quad (i=\overline{1,r})\tag{2.16}$$

схема B1

$$1+\Phi_i(s) = \prod_{k=1}^{i}\left(1+T_k(s)\right), \quad (i=\overline{1,r}) \tag{2.17}$$

схемы Bν (ν=2, 3)

$$\Phi_i(s) = \Phi_{i-1}(s)\left[1+E_i(s)\cdot M_i(s)\right]+E_i(s)N_i(s), \quad (i=\overline{1,r}) \tag{2.18}$$

здесь $\Phi(s)=\Phi_r(s)$, $\Phi_0(s)\equiv 0$.

3°. Пусть в схемах A и B1 для п. Л.—С. $\beta_i(s)$ $(i=\overline{1,r})$ длительности обслуживания i-вызова при $s^+\to 0$ ($\operatorname{Re} s\geqq 0$; $s\to 0$) имеет место разложение

$$\beta_i(s) = 1-\beta_{i1}\cdot s+\alpha_i^{(\beta)}s^{\gamma_{(i)}}\left(1+\varphi_{(i)}(s)\right), \tag{3.1}$$

где $1<\gamma_{(i)}\leqq 2$, $\alpha_i^{(\beta)}$-некоторая положительная постоянная, $\beta_{i1}=\int\limits_0^\infty t\,dB_i(t)$, а $\varphi_{(i)}(s)=0_s(1)$.

В схемах же B2 и B3 соответствующее разложение предполагаем выполненным лишь для первого потока $c\gamma_{(1)}=\gamma$.

Положим $(i=\overline{1,r})$:

$$\gamma_i = \min\left(\gamma_{(1)}, \ldots, \gamma_{(i)}\right), \quad L_i = \{k\leqq i\colon\ \gamma_{(k)}=\gamma_i\}, \quad \lambda_i = \gamma_i(\gamma_i-1)^{-1},$$
$$B_i = \sum_{k\in L_i} a_k\alpha_k^{(\beta)}, \quad B_1 = B.$$

Лемма 2 (см. [4]). Пусть выполнены разложения (3.1). Тогда при $s^+\to 0$ и $\varrho_{i1}<1$ справедливы следующие асимптотические разложения

$$\mu_{i+1}(s) = \varrho_i^{-1}s-\varrho_i^{-(\gamma_i+1)}K_is^{\gamma_i}+0(s^{\gamma_i}), \tag{3.2}$$

$$y_i(s) = \varrho_{i-1}\cdot\varrho_i^{-1}\{s-\varrho_i^{-\gamma_i}P_is^{\gamma_i}\}+0(s^{\gamma_i}), \tag{3.3}$$

$$v_{i+1}(s) = \varrho_{i+1}\varrho_i^{-1}s+\varrho_i^{-\gamma_{i+1}}P_{i+1}s^{\gamma_{i+1}}+0(s^{\gamma_{i+1}}), \tag{3.4}$$

где $P_i=K_i-K_{i-1}\dfrac{\varrho_i}{\varrho_{i-1}}\chi(\gamma_i=\gamma_{i-1})$, $\chi(A)$-индикатор события A, а

$$K_i = \begin{cases} B_i, & \text{если } \gamma_i<2,\\ 2\varrho_{i2}, & \text{если } \gamma_i=2.\end{cases}$$

Будем говорить, что система массового обслуживания $\vec{M}_r|\vec{G}_r|1|\infty$ находится в условиях критической загрузки, если $\varrho\stackrel{\text{def}}{=}1-\varrho_{r1}\downarrow 0$.

Условия критической загрузки создаются следующим образом.

Пусть при $\varrho\downarrow 0$ ф. р. $B_i(t)$ $(i=\overline{1,r})$ фиксированы и существуют пределы $(i,\ j=\overline{1,r})$

$$\bar{c}_{ij} = \lim_{\varrho\downarrow 0} c_{ij},\ c_{ij} = \varrho_i\varrho_j^{-1}(\bar{c}_i=\bar{c}_{ii-1},\ \varrho_i = 1-\varrho_{i1},\ \varrho_0 = 1).$$

Далее, пусть для индексов p_u $(1\leqq p=p_1<p_2<\ldots<p_m=r)$ и только для них $\bar{c}_{p_u}=0$ $(u=\overline{1,m};\ m\leqq r)$. Следовательно, у первых $p-1$ потоков суммарная

загрузка ϱ_{p-11} стремится при $\varrho\downarrow 0$ к числу, меньшему единицы ($\bar\varrho_{p-1}>0$), а ϱ_{p1}-к единице.

Не ограничивая общности, положим, что параметры $a_1, \ldots, a_{p-1}$ фиксированы и равны своим предельным при $\varrho\downarrow 0$ значениям. Поэтому считаем, что в условиях критической загрузки меняются только величины $a_p, \ldots, a_r$.

Обозначим:

$$\bar K_p = \lim_{\varrho\downarrow 1} K_i = \begin{cases} 2\bar\varrho_{p2}, \text{ если } \gamma_p = 2 \\ B_{p-1}+\bar\varrho_{p-1}\cdot\beta_{p1}^{-1}\alpha_p^{(\beta)}\chi(\gamma_{(p)}=\gamma_{p-1}), \text{ если } \gamma_p=\gamma_{p-1}<2, \\ \bar\varrho_{p-1}\cdot\beta_{p1}^{-1}\alpha_p^{(\beta)}, \text{ если } \gamma_p\neq\gamma_{p-1}, \gamma_{p-1}<2, \gamma_p<2. \end{cases}$$

4°. Прежде чем перейти к формулировке и доказательству основных результатов, сделаем ряд полезных замечаний.

Рассмотрим вначале схемы А и В1. Подставляя в (2.3) разложения функций $\beta_j(s)$ $(j=\overline{1,i})$ при $s^+\to 0$, получаем

$$\mu_{i+1}(s)=\varrho_i^{-1}s-\varrho_i^{-(\gamma_i+1)}K_i\{\mu_{i+1}(s)\}^{\gamma_i}(1+\varphi_i(\mu_{i+1}(s))), \tag{4.1}$$

где

$$K_i\varphi_i(\mu_{i+1}(s)) = \sum_{j\in L_i}\varphi_{(j)}(\mu_{i+1}(s))a_j\alpha_j^{(\beta)} + \sum_{\substack{j\notin L_i\\ j\le i}} a_j\alpha_j^{(\beta)}\mu_{(+)}^{\gamma_{(j)}-\gamma_i}(s)(1+\varphi_{(j)}(\mu_{i+1}(s))).$$

Очевидно, что для любого $\alpha\to 0$ при $\varrho_i\downarrow 0$ имеем $\mu_{i+1}(\alpha)\to 0$. Поскольку при $\varrho\downarrow 0$ функции $\beta_j(s)$ $(j=\overline{1,i})$ фиксированы, а $\varphi_{(j)}(s)$ фигурируют только в разложениях для $\beta_j(s)$, т. е. не зависят от загрузок, и стремятся к нулю при $s^+\to 0$, то при $\varrho_i\downarrow 0$

$$\varphi_i(\mu_{i+1}(\varrho_i^{\lambda_p}s)) = 0_{\varrho_i}(1).$$

При фиксированных загрузках и $s^+\to 0$ из соотношения (3.2) следует, что

$$\mu_{i+1}(s) = \varrho_i^{-1}s-\varrho_i^{-(\gamma_i+1)}K_is^{\gamma_i}(1+\psi_i(s)), \tag{4.2}$$

где $\psi_i(s)=0_s(1)$.

Пусть и в нашем случае переменных загрузок $\mu_{i+1}(s)$ задается соотношением (4.2). Тогда имеет место

Лемма 3. Для схем А и В1 равномерно по достаточно малым s при $\varrho\downarrow 0$

$$\max_{0\le u\le\varrho_i^{\lambda_i}s} |\psi_i(u)| = 0_{\varrho_i}(1) \quad (i\ge p).$$

Доказательство. Сравнивая соотношения (4.1) и (4.2), приходим к следующей связи между функциями $\varphi_i(s)$ и $\psi_i(s)$:

$$\psi_i(s) = \{\varrho_is^{-1}\mu_{i+1}(s)\}^{\gamma_i}\cdot[1+\varphi_i(\mu_{i+1}(s))]-1. \tag{4.3}$$

Обозначим $(i\ge p)$:

$$\varkappa_i(\varrho_i) = \frac{\mu_{i+1}(\varrho_i^{\lambda_i}s)}{\varrho_i^{\lambda_i-1}\cdot s}.$$

Тогда при $\varrho\downarrow 0$, подставив в (4.3) вместо s выражение $\varrho_i^{\lambda_i}s$, где s-достаточно мало, имеем:

$$\psi_i(\varrho_i^{\lambda_i}s) = \varkappa_i^{\gamma_i}(\varrho_i)(1+0_{\varrho_i}(1))-1.$$

Покажем, что равномерно по достаточно малым s при $\varrho\downarrow 0$

$$\varkappa_i(\varrho_i) = 1+0_{\varrho_i}(1).$$

Запишем для этого соотношение (2.3) в виде:

$$\mu_{i+1}(s) = s\cdot\left\{1-\sum_{j=1}^{i} a_j b_j(\mu_{i+1}(s))\right\}^{-1}. \tag{4.4}$$

В силу неравенств

$$0 \leqq \mu_{i+1}(s) \leqq s(1+\sigma_i\pi_{i1}) = \frac{s}{\varrho_i}$$

из (4.4) выводим двусторонние оценки для функции $\mu_{i+1}(s)$:

$$s\cdot\left\{1-\sum_{j=1}^{i} a_j b_j(s\varrho_i^{-1})\right\}^{-1} \leqq \mu_{i+1}(s) \leqq s\varrho_i^{-1}, \tag{4.5}$$

или, после подстановки вместо s величины $\varrho_i^{\lambda_i}s$, где s-достаточно мало,

$$\frac{\varrho_i}{1-\sum_{j=1}^{i} a_j b_j(\varrho_i^{\lambda_i-1}s)} = \frac{\varrho_i}{\varrho_i+0(\varrho_i)} \leqq \varkappa_i(\varrho_i) \leqq 1.$$

Последнее соотношение при $\varrho\downarrow 0$ дает, что $\varkappa_i(\varrho_i)\to 1$, т. е. $\psi_i(\varrho_i^{\lambda_i}s)=0_{\varrho_i}(1)$ равномерно по достаточно малым s.

Подставив же вместо s выражение $\varrho_i^{\lambda_i}\cdot\alpha\cdot s$, где $\alpha\to 0$ при $\varrho\downarrow 0$, в соотношение (4.5), убеждаемся в справедливости леммы 3 для всех $0\leqq u\leqq\varrho_i^{\lambda_i}s$. Лемма 3 доказана.

Перейдем к рассмотрению схем B2 и B3.

Лемма 4. Для схем Bν ($\nu=2, 3$) при $s^+\to 0$ равномерно по ϱ_i ($i\geqq p$)

$$\nu_i(s) = s-(\varrho_{i-1}-\varrho_i)\mu_i(s)+o\left[(\mu_i(s)-s)^2+s^2\right]. \tag{4.6}$$

Доказательство. Введем в рассмотрение функции ($0\leqq z\leqq 1$):

$$\varphi_i(z, s) = s-a_i(s+z)b_i(s+\sigma_{i-1})\begin{cases}1, & \text{для схемы B2,}\\ [(s+z)\beta_i(s+\sigma_{i-1})+\beta_i(s+\sigma_{i-1})]^{-1}, & \text{для B3.}\end{cases}$$

Отметим, что $\nu_i(s)=\varphi_i(\mu_i(s)-s, s)$. Поскольку функция $\varphi_i(z, s)$ аналитична в окрестности S точки ($z=0, s=0$) как функция двух переменных, то внутри S она разложима в ряд Тейлора

$$\varphi_i(z, s) = \varphi_i(0, 0)+\frac{\partial\varphi_i(0, 0)}{\partial z}\cdot z+\frac{\partial\varphi_i(0, 0)}{\partial s}s+A_i(z, s), \tag{4.7}$$

где $(0<\theta<1)$

$$A_i(z,s)=\frac{1}{2}\left[\frac{\partial^2\varphi_i(\theta z,\theta s)}{\partial z^2}z^2+2z\cdot s\frac{\partial^2\varphi_i(\theta z,\theta s)}{\partial z\partial s}+\frac{\partial^2\varphi_i(\theta z,\theta s)}{\partial s^2}s^2\right].$$

Поскольку a_i входит в $\varphi_i(z,s)$ линейным образом, а при $\varrho_i\downarrow 0$ параметры $a_1,\dots,a_i$ ограничены в совокупности, то равномерно по ϱ_i имеем $A_i(z,s)=o(z^2+s^2)$. Выберем s настолько малым, чтобы $(\mu_i(s)-s,s)\in S$. Подставив в соотношение (4.7) вместо z выражение $\mu_i(s)-s$, приходим к (4.6). Лемма 4 полностью доказана.

5°. При доказательство основного результата настоящей работы существенно используются асимптотические разложения функций $\mu_{i+1}(s)$, $y_i(s)$ и $v_i(s)$ в условиях критической загрузки, которые представляют самостоятельный интерес при асимптотическом исследовании различных характеристик многих приоритетных моделей.

Теорема 2. Пусть выполнены условия (3.1). При $\varrho\downarrow 0$ для всех схем А и В справедливы следующие равномерные по достаточно малым s асимптотические разложения $(i=\overline{p,r})$:

$$\mu_{i+1}(\varrho_i^{\lambda_p}s)=\varrho_i^{\lambda_p-1}\Lambda(s)(1+0(1)),\tag{5.1}$$

$$\varrho_i^{\lambda_p}s-v_{i+1}(\varrho_i^{\lambda_p}s)=\varrho_i^{\lambda_p}(1-c_{i+1})\Lambda(s)(1+0(1)),\tag{5.2}$$

$$y_i(\varrho_i^{\lambda_p}s)-\varrho_i^{\lambda_p}s=\varrho_{i-1}\varrho_i^{\lambda_p-1}(1-c_i)\Lambda(s)(1+0(1)),\tag{5.3}$$

где $\Lambda(s)$-решение уравнения

$$z+\overline{K}_pz^{\gamma_p}=s,\tag{5.4}$$

удовлетворяющее начальному условию $\Lambda(0)=0$.

Доказательство проведем математической индукцией по k.

При $k=p$ справедливость теоремы 2 следует из таких рассуждений. При $s^+\to 0$ в силу леммы 2 и условия $\varrho_{p-11}<1$ имеем

$$v_p(s)=c_p\cdot s+\varrho_{p-1}^{-\gamma_p}P_ps^{\gamma_p}(1+f_{p-1}(s)),\tag{5.5}$$

где $f_{p-1}(s)=0_s(1)$.

Поскольку $f_{p-1}(s)$ не зависит от параметров $a_p,\dots,a_r$, то подставляя вместо s выражение $\varrho_{p-1}\varrho^{\lambda_p-1}s$, где s-достаточно мало, при $\varrho\downarrow 0$ с учетом $\bar{c}_p=0$ из (5.5) имеем

$$v_p(\varrho_{p-1}\varrho_p^{\lambda_p-1}s)=\varrho_p^{\lambda_p}(s+\overline{K}_ps^{\lambda_p})(1+0(1)).\tag{5.6}$$

Тогда из соотношения (2.2) при $\varrho\downarrow 0$ следует, что

$$\varrho_{p-1}\varrho_p^{\lambda_p-1}s=y_p(v_p(\varrho_{p-1}\varrho_p^{\lambda_p-1}s))=y_p(\varrho_p^{\lambda_p}(s+\overline{K}_ps^{\gamma_p})(1+0(1))).\tag{5.7}$$

Рассмотрим неявное уравнение

$$F(z,s)=z+\overline{K}_pz^{\gamma_p}-s=0.$$

Поскольку $F(0,0)=0$, $F'_z(z,s)\neq 0$, то по теореме о неявной функции существует единственная непрерывно дифференцируемая функция $\Lambda(s)$, удовлетворяющая начальному условию $\Lambda(0)=0$.

Так как

$$s=\Lambda(s)+\bar{K}_p\Lambda^{\gamma_p}(s) \tag{5.8}$$

то, положив $s=\Lambda(\tau)$ в соотношение (5.7), получаем

$$y_p\big(\varrho_p^{\lambda_p}\tau(1+0(1))\big)=\varrho_{p-1}\varrho_p^{\lambda_p-1}\Lambda(\tau),$$

откуда, пользуясь монотонностью функции $y_p(s)$, с помощью стандартной техники оценок нетрудно вывести, что

$$y_p(\varrho_p^{\lambda_p}\tau)=\varrho_{p-1}\varrho_p^{\lambda_p-1}\Lambda(\tau)\big(1+0(1)\big), \tag{5.9}$$

которое совпадает с разложением (5.3) с $i=p$ и $\bar{c}_p=0$.

Далее, при $s^+\to 0$ из соотношения (3.2) следует

$$\mu_p(s)=\varrho_{p-1}^{-1}s-\varrho_{p-1}^{-\gamma_{p-1}-1}K_{p-1}s^{\gamma_p-1}\big(1+\psi_{p-1}(s)\big).$$

Поскольку функция $\psi_{p-1}(s)$ не зависит от параметров $a_p,\dots,a_r$, то из соотношения (2.1) при $\varrho\downarrow 0$ получаем:

$$\mu_{p+1}(\varrho_p^{\lambda_p}s)=\varrho_{p-1}^{-1}y_p(\varrho_p^{\lambda_p}s)-\varrho_{p-1}^{-\gamma_{p-1}-1}K_{p-1}y_p^{\gamma_p-1}(\varrho_p^{\lambda_p}s)\cdot\{1+\psi_{p-1}(y_p(\varrho_p^{\lambda_p}s))\}=$$
$$=\varrho_p^{\gamma_p-1}\Lambda(s)\big(1+0(1)\big), \tag{5.10}$$

т. е. соотношение (5.1) доказано в случае $i=p$.

Соотношение (5.2) для схем B2 и B3 доказывается непосредственной подстановкой в (4.6) вместо s величины $\varrho_p^{\lambda_p}s$, где s-достаточно мало, с использованием (5.10). Законность указанных действий гарантируется равномерностью по ϱ_p соотношения (4.6).

Докажем, что (5.7) имеет место при $i=p$ и для схем A и B1. Из формул (2.4) при $s^+\to 0$ следует, что

$$s-v_{p+1}(s)=a_{p+1}\beta_{p+1}\,\mu_{p+1}(s)-a_{p+1}\alpha_{p+1}^{(\beta)}\mu_{p+1}^{\gamma_{p+1}}(s)\big(1+\varphi_{(p+1)}(\mu_{p+1}(s))\big). \tag{5.11}$$

Будем определять $v_{p+1}(s)$ по последней формуле и при переменных загрузках. Ввиду того, что функция $\varphi_{(p+1)}(s)$ не зависит от параметров $a_p,\dots,a_r$, при $\varrho\downarrow 0$ имеем

$$\varphi_{(p+1)}\big(\mu_{p+1}(\varrho_p^{\lambda_p}s)\big)=\varphi_{(p+1)}\big(\varrho_p^{\lambda_p-1}\Lambda(s)(1+0(1))\big)=0_{\varrho_p}(1).$$

Тогда при $\varrho_p\downarrow 0$ из соотношения (5.11) следует, что

$$\varrho_p^{\lambda_p}s-v_{p+1}(\varrho_p^{\lambda_p}s)=\varrho_p^{\lambda_p}(1-c_{p+1})\Lambda(s)\big(1+0(1)\big)-$$
$$-\varrho_p^{(\lambda_p-1)\gamma_{(p+1)}+1}(1-c_{p+1})\,\alpha_{p+1}^{(\beta)}\beta_{p+1\,1}^{-1}\Lambda^{\gamma_{(p+1)}}(s)\big(1+0(1)\big).$$

Сравним порядки обоих слагаемых в правой части полученного соотношения:

$$\frac{\varrho_p^{(\lambda_p-1)\gamma_{(p+1)}+1}(1-c_{p+1})}{\varrho_p^{\lambda_p}(1-c_{p+1})}=\varrho_p^{(\lambda_p-1)(\gamma_{(p+1)}-1)}\to 0.$$

Следовательно, формула (5.2) для схем А и В1 при $i=p$ верна.

Допустим, что утверждения теоремы 2 верны при $k=\overline{p+1, i-1}$ и докажем их справедливость при $k=i$.

Сначала получим разложение для функции $y_i(s)$.

а) Пусть $\bar{c}_i=0$. Тогда из предположения индукции при $\varrho\downarrow 0$

$$v_i(\varrho_{i-1}^{\lambda_p} s) = \varrho_{i-1}^{\lambda_p}[s-(1-c_i)\Lambda(s)](1+0(1)) =$$
$$= \varrho_{i-1}^{\lambda_p}[c_i s+(1-c_i)\cdot \overline{K}_p \Lambda^{\gamma_p}(s)](1+0(1)).$$

Пользуясь равномерностью по достаточно малым s при $\varrho\downarrow 0$, подставим вместо s выражение $c_i^{\lambda_p-1} s$.

Воспользовавшись легко доказуемым соотношением

$$\lim_{s\downarrow 0} s^{-1}\Lambda(s) = 1, \tag{5.12}$$

получаем, что (ср. с (5.6))

$$v_i(\varrho_{i-1}\,\varrho_i^{\lambda_p-1}s) = \varrho_i^{\lambda_p}(s+\overline{K}_p s^{\gamma_p})\cdot(1+0(1)), \tag{5.13}$$

откуда, аналогично получению соотношения (5.9), при $\varrho\downarrow 0$ имеем

$$y_i(\varrho_i^{\lambda_p} s) = \varrho_{i-1}\varrho_i^{\lambda_p-1}\Lambda(s)(1+0(1)). \tag{5.14}$$

б) Рассмотрим теперь случай $\bar{c}_i\neq 0$. Из предположения индукции при $\varrho\downarrow 0$ имеем:

$$v_i(\varrho_{i-1}^{\lambda_p} s) = \varrho_i^{\lambda_p}\tau(1+0(1)), \tag{5.15}$$

где τ определяется из равенства

$$c_i^{\lambda_p}\tau = s-(1-c_i)\Lambda(\tau).$$

Покажем, что

$$\Lambda(s)= c_i^{\lambda_p-1}\Lambda(\tau). \tag{5.16}$$

Действительно,

$$c_i^{\lambda_p}\cdot\tau = \Lambda(s)+\overline{K}_p\Lambda^{\gamma_p}(s)-(1-c_i)\Lambda(s) = = c_i\Lambda(s)+\overline{K}_p\Lambda^{\gamma_p}(s),$$

откуда следует, что

$$\tau = \frac{\Lambda(s)}{c_i^{\lambda_p-1}}+\overline{K}_p\left[\frac{\Lambda(s)}{c_i^{\lambda_p-1}}\right]^{\gamma_p}. \tag{5.17}$$

Поскольку

$$z = \Lambda(z+\overline{K}_p z^{\gamma_p}), \tag{5.18}$$

то из соотношения (5.17) имеем

$$\Lambda(\tau) = \Lambda\left(\frac{\Lambda(s)}{c_i^{\lambda_p-1}}+\overline{K}_p\left(\frac{\Lambda(s)}{c_i^{\lambda_p-1}}\right)^{\gamma_p}\right) = \frac{\Lambda(s)}{c_i^{\lambda_p-1}},$$

что и доказывает справедливость (5.16).

Теперь из предположения индукции и соотношения (2.2) при условиях критической загрузки имеем

$$y_i(v_i(\varrho_{i-1}^{\lambda_p}s)) - v_i(\varrho_{i-1}^{\lambda_p}s) = \varrho_{i-1}^{\lambda_p}s - v_i(\varrho_{i-1}^{\lambda_p}s) = \varrho_{i-1}^{\lambda_p}(1-c_i)\Lambda(s)(1+0(1)). \quad (5.19)$$

С другой стороны, из (5.14) следует, что

$$y_i(v_i(\varrho_{i-1}^{\lambda_p}s)) - v_i(\varrho_{i-1}^{\lambda_p}s) = y_i(\varrho_i^{\lambda_p}\tau(1+0(1))) - \varrho_i^{\lambda_p}\tau(1+0(1)).$$

Используя соотношения (5.19) и (5.16), выводим следующее асимптотическое равенство:

$$y_i(\varrho_i^{\lambda_p}\tau(1+0(1))) - \varrho_i^{\lambda_p}\tau(1+0(1)) = \varrho_{i-1}\varrho_i^{\lambda_p-1}(1-c_i)\Lambda(\tau)\cdot(1+0(1)),$$

откуда следует формула (5.3) при $\bar{c}_i \neq 0$.

Таким образом, при $\bar{c}_i = 0$ разложение $y_i(\varrho_i^{\lambda_p}s)$ при $\varrho \downarrow 0$ определяется по формуле (5.14), а при $\bar{c}_i \neq 0$-по (5.3). Но, поскольку при $\bar{c}_i \neq 0$

$$\frac{\varrho_i^{\lambda_p}}{\varrho_{i-1}\varrho_i^{\lambda_p-1}} = c_i \to 0,$$

то оба результата можно записать в форме (5.3).

Для доказательства (5.1) также рассмотрим два случая.

а) $\bar{c}_i = 0$. Тогда, исходя из предположения индукции и соотношений (2.1), (5.14) и (5.12), выводим

$$\mu_{i+1}(\varrho_i^{\lambda_p}s) = \mu_i(y_i(\varrho_i^{\lambda_p}s)) = \mu_i(\varrho_{i-1}^{\lambda_p}c_i^{\lambda_p-1}\Lambda(s)(1+0(1))) =$$
$$= \varrho_{i-1}^{\lambda_p-1}\Lambda(c_i^{\lambda_p-1}\Lambda(s)(1+0(1))(1+0(1))) = \varrho_i^{\lambda_p-1}\Lambda(s)(1+0(1)).$$

б) $\bar{c}_i \neq 0$. Покажем вначале, что

$$c_i^{\lambda_p-1}\Lambda(s) = \Lambda(c_i^{\lambda_p}s + c_i^{\lambda_p-1}(1-c_i)\Lambda(s)). \quad (5.20)$$

Действительно, воспользовавшись равенствами (5.18) и (5.8), выводим:

$$c_i^{\lambda_p-1}\Lambda(s) = \Lambda(c_i^{\lambda_p-1}\Lambda(s) + \bar{K}_p c_i^{\lambda_p}\Lambda^{\gamma_p}(s)) =$$
$$= \Lambda(c_i^{\lambda_p}(\Lambda(s) + \bar{K}_p\Lambda\,\gamma_p(s)) + c_i^{\lambda_p-1}(1-c_i)\Lambda(s)) = \Lambda(c_i^{\lambda_p}s + c_i^{\lambda_p-1}(1-c_i)\Lambda(s)),$$

что доказывает справедливость (5.20).

Поэтому, исходя из (2.1), (5.1) и (5.20), при $\varrho \downarrow 0$ выводим:

$$\mu_{i+1}(\varrho_i^{\lambda_p}s) = \mu_i(\varrho_{i-1}^{\lambda_p}(c_i^{\lambda_p}s + c_i^{\lambda_p-1}(1-c_i)\Lambda(s)(1+0(1)))) =$$
$$= \varrho_{i-1}^{\lambda_p-1}c_i^{\lambda_p-1}\Lambda(s)(1+0(1)) = \varrho_i^{\lambda_p-1}\Lambda(s)(1+0(1)).$$

Таким образом, мы получили асимптотическое разложение (5.1).

Соотношение (5.2) для $i = \overline{p+1, r}$ доказывается аналогично случаю $i=p$ с использованием разложения (5.1).

Теорема 2 доказана.

6°. Рассмотрим следующее уравнение:

$$z+z^{\gamma_p} = s. \tag{6.1}$$

Аналогично (5.4) нетрудно показать, что решением уравнения (6.1) является функция $\Delta(s)$, удовлетворяющая начальному условию $\Delta(0)=0$.

Заметим, что в случае $\gamma_p=2$ (см. [5])

$$\Delta(s) = \sqrt{\frac{1}{4}+s}-\frac{1}{2}.$$

Покажем, что функции $\Lambda(s)$ и $\Delta(s)$ связаны соотношением:

$$\Delta(s) = \bar{K}_p^{\lambda_p-1}\Lambda(s\cdot\bar{K}_p^{-(\lambda_p-1)}). \tag{6.2}$$

Действительно, подставив в (5.8) вместо s величину $s\bar{K}_p^{-(\lambda_p-1)}$, имеем

$$\bar{K}_p^{\lambda_p-1}\Lambda\left(\frac{s}{\bar{K}_p^{\lambda_p-1}}\right)+\left[\bar{K}_p^{\lambda_p-1}\Lambda\left(\frac{s}{\bar{K}_p^{\lambda_p-1}}\right)\right]^{\gamma_p} = s,$$

откуда, пользуясь единственностью решения уравнения (6.1), приходим к (6.2).

Положим $(i=\overline{p, r})$:

$$A_i(s) = \begin{cases} 1, & \text{если} \quad \bar{c}_i = 1, \\ \left(1+\delta_i^{\gamma_p-1}+\chi(\bar{c}_{i+1})\Delta(q_{i+1})\dfrac{\Delta^{\gamma_p-1}(q_{i+1})-\delta_i^{\gamma_p-1}}{\Delta(q_{i+1})-\delta_i}\right)^{-1}, & \text{если} \quad \bar{c}_i = 0, \\ \bar{c}_i\dfrac{\Delta(\delta_i)-\bar{c}_i^{\lambda_p-1}\chi(\bar{c}_{i+1})\Delta(q_{i+1})}{\Delta(\delta_i)-\Delta(q_i)}, & \text{если} \quad 0<\bar{c}_i<1, \end{cases} \tag{6.3}$$

где δ_i и q_i $(i=\overline{p, r+1})$ определяются рекуррентным образом:

$$\delta_{r+1} = q_{r+1} = 0,$$

$$\delta_i = \begin{cases} s_i+\chi(\bar{c}_{i+1})\Delta(\delta_{i+1}), & \text{если} \quad \bar{c}_i = 0, \\ \bar{c}^{\lambda_p-1}\{s_i+\chi(\bar{c}_{i+1})[\bar{c}_i\delta_{i+1}+(1-\bar{c}_i)\Delta(\delta_{i+1})]\}, & \text{если} \quad \bar{c}_i \neq 0, \end{cases} \tag{6.4}$$

$$q_i = \begin{cases} \chi(\bar{c}_{i+1})q_{i+1}-\delta_i(1+\delta_i^{\gamma_p-1}), & \text{если} \quad \bar{c}_i = 0, \\ (1-\bar{c}_i)\Delta(\delta_i)+\chi(\bar{c}_{i+1})\bar{c}_i^{\lambda_p}q_{i+1}, & \text{если} \quad \bar{c}_i \neq 0. \end{cases} \tag{6.5}$$

Легко заметить, что функция $A_i(s)$ при $p_u \leqq i < p_{u+1}$ зависит только от вектора $s_u^{K_u}=(s_{p_u}, \ldots, s_{p_{u+1}-1})$, где $K_u=p_{u+1}-p_u-\delta_{uo}$.

Для удобства дальнейших выкладок введем обозначения:

$$\tilde{\delta}_i = \delta_i\cdot\bar{K}_p^{-(\lambda_p-1)},\ \tilde{q}_i = q_i\cdot\bar{K}_p^{-(\lambda_p-1)},$$

$$s_i^* = s_i\cdot\{\chi(p-i)+R_i[1-\chi(p-i)]\}, \tag{6.6}$$

где $R_i = \varrho_{i-1}\varrho_i^{\lambda_p-1}\cdot\bar{K}_p^{-(\lambda_p-1)}$

$$s^* = (s_1^*, \ldots, s_r^*),\ \eta_k^*(s) = \eta_k(s_k^*, \ldots, s_r^*),\ \theta_k^*(s) = \theta_k(s_k^*, \ldots, s_r^*).$$

Лемма 5. При $\varrho\downarrow 0$ справедливы следующие асимптотические соотношения $(i=\overline{p, r})$:

$$\eta_i^* \stackrel{\text{def}}{=} \eta_i^*(s) = \left\{\varrho_i^{\lambda_p-1}(1-\chi(\bar{c}_i))+\varrho_{i-1}^{\lambda_p-1}\chi(\bar{c}_i)\right\}\varrho_{i-1}\cdot\tilde{\delta}_i(1+0(1)), \qquad (6.7)'$$

$$\theta_i^* \stackrel{\text{def}}{=} \theta_i^*(s)=(\eta_i^*+\varrho_i^{\lambda_p}\tilde{q}_i)(1-\chi(\bar{c}_i))+\varrho_{i-1}^{\lambda_p}\tilde{q}_i\chi(\bar{c}_i)+0(\varrho_i^{\lambda_p}). \qquad (6.8)$$

Доказательство. Покажем вначале справедливость соотношения (6.7), используя математическую индукцию по k.

При $k=r$ и $\varrho\downarrow 0$ имеем

$$\eta_r^* = s_r^* = \varrho_{r-1}\varrho_r^{\lambda_p-1}\overline{K}_p^{-(\lambda_p-1)}s = c_r^{\lambda_p-1}\varrho_{r-1}^{\lambda_p}\overline{K}_p^{-(\lambda_p-1)}s,$$

что в силу определения $\tilde{\delta}_r$ эквивалентно (6.7).

Положим, что (6.7) выполнено для всех $k=\overline{i+1, r-1}$ и докажем его справедливость при $k=i$ $(i=\overline{p, r-1})$.

а) Пусть $\bar{c}_{i+1}=0$. По предположению индукции при $\varrho\downarrow 0$

$$\eta_{i+1}^* = \varrho_i\varrho_{i+1}^{\lambda_p-1}\tilde{\delta}_{i+1}(1+0(1)).$$

Тогда, с учетом соотношений (5.14) и (5.12), выводим

$$y_i(\eta_i^*) = y_i(\varrho_i^{\lambda_p}c_{i+1}^{\lambda_p-1}\tilde{\delta}_{i+1}(1+0(1))) =$$
$$= \varrho_{i-1}\varrho_i^{\lambda_p-1}c_{i+1}^{\lambda_p-1}\tilde{\delta}_{i+1}(1+0(1)) = 0(\varrho_{i-1}\varrho_i^{\lambda_p-1}),$$

откуда и из определения η_i следует, что

$$\eta_i^* = \varrho_{i-1}\varrho_i^{\lambda_p-1}K_p^{-(\lambda_p-1)}s(1+0(1)) = \varrho_{i-1}\varrho_i^{\lambda_p-1}\tilde{\delta}_i(1+0(1)).$$

б) Если $\bar{c}_{i+1}\neq 0$, то в силу предположения индукции при $\varrho\downarrow 0$

$$\eta_{i+1}^* = \varrho_i^{\lambda_p}\tilde{\delta}_{i+1}(1+0(1)),$$

откуда и из разложения (5.3) получаем

$$y_i(\eta_{i+1}^*) = y_i(\varrho_i^{\lambda_p}\tilde{\delta}_{i+1}(1+0(1))) =$$
$$= \varrho_{i-1}\varrho_i^{\lambda_p-1}[(1-\bar{c}_i)\Lambda(\tilde{\delta}_{i+1})+\bar{c}_i\tilde{\delta}_{i+1}](1+0(1)).$$

Последнее соотношение вкупе с определением η_i, (6.4) и (6.6) дает требуемое соотношение (6.7).

Перейдем теперь к доказательству соотношения (6.8), которое также проведем индукцией по k.

Очевидно, что при $k=r+1$ (6.8) имеет место. Положим, что оно справедливо при $k=\overline{r, i+1}$ и покажем, его справедливость при $k=i$.

а) Пусть $\bar{c}_i \neq 0$. Тогда из соотношений (6.7), (5.2) и определения q_i следует, что в условиях критической загрузки

$$\theta_i^* = \eta_i^* - v_i(\eta_i^*) + \chi(\bar{c}_{i+1}) \cdot \varrho_{i-1}^{\lambda_p - 1} c_i^{\lambda_p} \tilde{q}_{i+1}(1 + 0(1)) =$$
$$= \varrho_{i-1}^{\lambda_p}[(1 - c_i) \Lambda(\tilde{\delta}_i) + \chi(\bar{c}_{i+1}) \cdot c_i^{\lambda_p} \tilde{q}_{i+1}] \cdot (1 + 0(1))$$

откуда и следует (6.8).

б) Если же $\bar{c}_i = 0$, то из индуктивного предположения и соотношения (5.13) при $\varrho \downarrow 0$ получаем

$$\theta_i^* = \eta_i^* + \varrho_i^{\lambda_p}[\chi(\bar{c}_{i+1}) \tilde{q}_{i+1} - \tilde{\delta}_i(1 + \bar{K}_p \tilde{\delta}_i^{\gamma_p - 1})](1 + 0(1)),$$

что при $\bar{c}_i = 0$ совпадает с (6.8). Лемма 5 доказана полностью.

Лемма 6. При условиях критической загрузки для всех схем А и В имеют место асимптотические соотношения ($i = \overline{p, r}$):

$$\mu_i(\eta_i^*) = \varrho_i^{\lambda_p - 1} \tilde{\delta}_i (1 - \chi(\bar{c}_i)) + \varrho_{i-1}^{\lambda_p - 1} \Lambda(\tilde{\delta}_i) \chi(\bar{c}_i) + 0(\varrho_i^{\lambda_p - 1}), \tag{6.9}$$

$$\theta_{i+1}^* - v_i(\theta_i^*) = \varrho_i^{\lambda_p} \tilde{q}_i (1 - \chi(\bar{c}_i)) + \varrho_{i-1}^{\lambda_p}(1 - \bar{c}_i)[\Lambda(\tilde{q}_i) - \Lambda(\tilde{\delta}_i)] \chi(\bar{c}_i) + 0(\varrho_i^{\lambda_p}). \tag{6.10}$$

Доказательство леммы 6 проводится аналогично доказательству предыдущей леммы.

Лемма 7. Пусть для схемы В1 при $\varrho \downarrow 0$ выполнено условие

$$\varrho_{p_u - 1}^{-1} \sum_{j = p+1}^{\varrho_{p_u} - 1} a_j \varrho_{p_u}^{\frac{\gamma_{(j)} - 1}{\gamma_p - 1}} \to 0 \quad (u = \overline{2, m}). \tag{6.11}$$

Тогда

$$\mu_i(\theta_i^*) = \begin{cases} \mu_i(\eta_i^*) + c_i \varrho_i^{\lambda_p - 1} \tilde{q}_i (1 + 0(1)), & \text{при } i = p_u (u = \overline{1, m}), \\ \varrho_{i-1}^{\lambda_p} \Lambda(\tilde{q}_i)(1 + 0(1)), & \text{если } \bar{c}_i \neq 0; \end{cases} \tag{6.12}$$

$$\mu_{i+1}(\theta_{i+1}^*) - \mu_i(\theta_i^*) = \varrho_i^{\lambda_p - 1}[\Lambda(\tilde{q}_{i+1}) \chi(\bar{c}_{i+1}) - \delta_i](1 - \chi(\bar{c}_i)) +$$
$$+ \varrho_{i-1}^{\lambda_p - 1}[c_i^{\lambda_p - 1} \Lambda(\tilde{q}_{i+1}) \chi(\bar{c}_{i+1}) - \Lambda(\tilde{q}_i)] \chi(\bar{c}_i) + 0(\varrho_i^{\lambda_p - 1}). \tag{6.13}$$

Доказательство. Второе из соотношений (6.12) легко доказать, если воспользоваться (6.8) и (6.9).

Пусть теперь $\bar{c}_i = 0$. Ясно, что

$$\mu_i(\theta_i^*) - \mu_i(\eta_i^*) = \mu_i'(\eta_i^*(1 + 0(1)))(\theta_i^* - \eta_i^*).$$

Из соотношения (2.3) дифференцированием по s выводим

$$\mu_i'(s) = \left\{1 - \sum_{j=1}^{i-1} a_j \beta_j'(\mu_i(s))\right\}^{-1}. \tag{6.14}$$

Покажем, что при $s^+ \to 0$ ($j = \overline{1, r}$)

$$\beta_{j1} + \beta_j'(s) = \alpha_j^{(\beta)} \gamma_{(j)} s^{\gamma_{(j)} - 1}(1 + 0(1)). \tag{6.15}$$

Действительно, интегрированием по частям можно показать, что

$$\beta_j'(s) = -\frac{1-\beta(s)}{s} + \frac{1}{s}\int_0^\infty u\bar{B}_j(us^{-1})e^{-u}\,du. \tag{6.16}$$

По тауберовым теоремам [6] из разложения (3.1) получаем, что для $t \geqq 0$, $j=\overline{1, r}$

$$\bar{B}_j(t) = 1 - B_j(t) = A_j t^{-\gamma_{(j)}}\left(1+\bar{\varphi}_j(t)\right) \tag{6.17}$$

где $\bar{\varphi}_j(t) \to 0$ при $t \to \infty$, а

$$A_j = \alpha_j^{(\beta)} \frac{\Gamma(\gamma_{(j)}-1)}{\Gamma(2-\gamma_{(j)})}.$$

Из (6.17) получаем, что

$$s^{-1}\int_0^\infty ue^{-u}\bar{B}_j(us^{-1})\,du = s^{\gamma_{(j)}-1}A_j\int_0^\infty u^{1-\gamma_{(j)}}e^{-u}\left(1+\bar{\varphi}_j(us^{-1})\right)du,$$

Ясно, что при $u>0$

$$u^{1-\gamma_{(j)}}e^{-u}\left(1+\bar{\varphi}_j(us^{-1})\right) \to u^{1-\gamma_{(j)}}e^{-u},$$

откуда в силу теоремы Лебега при $s^+ \to 0$

$$\int_0^\infty u^{1-\gamma_{(j)}}e^{-u}\left(1+\bar{\varphi}_j(us^{-1})\right) = \Gamma(2-\gamma_{(j)})\left(1+0_s(1)\right),$$

что вкупе с (6.16) дает требуемое соотношение (6.15).

Тогда из соотношения (6.14) выводим:

$$\mu_i'\left(\eta_i^*(1+0(1))\right) = \left\{\varrho_{i-1} + \sum_{j=1}^{i-1} a_j\alpha_j^{(\beta)}\gamma_{(j)}\varrho_i^{(\lambda_p-1)(\gamma_{(j)}-1)}\left(1+0(1)\right)\right\}^{-1}. \tag{6.18}$$

Ясно, что при

$$\varrho_{i-1}^{-1}\sum_{j=1}^{p} a_j\alpha_j^{(\beta)}\gamma_{(j)}\varrho_i^{(\lambda_p-1)(\gamma_{(j)}-1)} \to 0,$$

$$\varrho_{i-1}^{-1}\sum_{j=p_{u-1}^{+1}}^{p_u-1} a_j\alpha_j^{(\beta)}\gamma_{(j)}\varrho_i^{(\lambda_p-1)(\gamma_{(j)}-1)} \to 0,$$

откуда и из (6.18), (6.12) следует первое из соотношений (6.12).

Асимптотическое разложение (6.13) легко получается из (6.12) и (6.9).

Лемма 8. Если $\varrho \downarrow 0$, то $(i=\overline{p, r})$

$$c_i\left[\chi(\bar{c}_i) + \frac{\varrho_{i-1}-\varrho_i}{a_i}\cdot E_i(s^*)\right] \sim A_i(s), \tag{6.19}$$

$$T_i(s^*) \sim E(s^*)\frac{\varrho_{i-1}-\varrho_i}{a_i}, \tag{6.20}$$

$$N_i(s^*) \sim M_i(s^*) \sim \frac{\varrho_{i-1}-\varrho_i}{a_i}. \tag{6.21}$$

Доказательство. Покажем вначале справедливость (6.19).

а) Пусть $\bar{c}_i=0$. Из (2.7), (6.10) и (6.13) при $\varrho\downarrow 0$ находим

$$c_i\cdot\frac{\varrho_{i-1}-\varrho_i}{a_i}\cdot E_i(s^*)\sim\frac{\chi(\bar{c}_{i+1})\Lambda(\tilde{q}_{i+1})-\tilde{\delta}_i}{\tilde{q}_i},$$

откуда, используя (6.2)—(6.6), приходим к отношению эквивалентности (6.19).

б) Если $\bar{c}_i\neq 0$, то при $\varrho\downarrow 0$ имеем

$$c_i\left[1+\frac{\varrho_{i-1}-\varrho_i}{a_i}\cdot E_i(s^*)\right]\sim\bar{c}_i\,\frac{\Lambda(\tilde{\delta}_i)-c_i^{\lambda_p-1}\Lambda(\tilde{q}_{i+1})\chi(\bar{c}_{i+1})}{\Lambda(\tilde{\delta}_i)-\Lambda(\tilde{q}_i)},$$

откуда и из (6.2), (6.3) следует (6.19).

Остается заметить, что при $\bar{c}_i=1$ в силу равенства (см. (6.5)) $\tilde{q}_i=\chi(\bar{c}_{i+1})\tilde{q}_{i+1}$ следует $A_i(s)=1$. Тем самым доказательство (6.19) завершено.

Отношение эквивалентности (6.20) легко вытекает из (2.10), (6.12) и (6.10), поскольку

$$\frac{\theta^*_{i+1}-\nu_i(\theta^*_i)}{\varrho_{i-1}-\varrho_i}\sim\mu_i(\theta^*_i)-\mu_i(\eta^*_i).$$

Аналогично доказывается и отношение эквивалентности (6.21).

7°. Введем в рассмотрение системы $\vec{M}_p|\vec{G}_p|1|\infty$ (схема А) и $\vec{M}_{p-1}|\vec{G}_{p-1}|1|\infty$ (схемы В), в которых параметры первых $p-1$ потоков фиксированы и совпадают с соответствующими значениями параметров первых $p-1$ потоков в исходных системах $\vec{M}_r|\vec{G}_r|1|\infty$.

Длительности обслуживания в каждой введенной системе независимы в совокупности и не зависят от входных потоков. Ф. р. длительности обслуживания i-вызова $(i=\overline{1,p})-B_i(t)$, $B_i(+0)=0$.

Поскольку во введенных системах п. Л.—С. $\omega(s^{(p)})$ и $\omega(s^{(p-1)})$ находятся на основании теоремы 1, то справедлива.

Лемма 9. Для введенной системы (схема А) $\vec{M}_p|\vec{G}_p|1|\infty$

$$\lim_{\varrho\downarrow 0}\omega(s^{(p-1)},0)=\frac{\varrho_{p-1}}{\beta_{p1}}\Psi_p(s^{(p-1)},0), \tag{7.1}$$

где $\Psi_p(s^{(p-1)},0)$ находится из теоремы 1 и не зависит от $\bar{a}_p=\dfrac{\varrho_{p-1}}{\beta_{p1}}$.

Доказательство. Прежде всего заметим, что в силу определения η_p и θ_p имеем $\eta_p=s_p$, $\theta_{p+1}=0$, а при $s_p\downarrow 0$ из соотношения (3.4) выводим

$$\theta_p=s_p-\nu_p(s_p)=s_p(1-c_p)+0(s_p).$$

Тогда в силу асимптотических соотношений (3.2) и (3.4) при $\varrho\downarrow 0$ имеем

$$E_p(s^{(p)})=a_p\lim_{s_p\downarrow 0}\frac{\mu_p(\theta_p)}{\nu_p(\theta_p)}=a_p\lim_{s_p\downarrow 0}\frac{\varrho_{p-1}^{-1}s_p(1-c_p)+0(s_p)}{s_pc_p(1-c_p)+0(s_p)}=\frac{a_p}{\varrho_p}. \tag{7.2}$$

Поскольку η_i $(i=\overline{1, p+1})$ не зависят от a_p и θ_i при $s_p=0$ не зависят от a_p, то при $s_p=0$ от a_p не зависят и функции $H_i(s)$, $V_j^{\{i\}}(s)$ $(j=\overline{1, p-1};\ i=\overline{1, p})$, а следовательно и $\Psi_i(s^{(p-1)}, 0)$.

Подставляя (7.2) в (2.15) и принимая во внимание конечность функции $\Psi_i(s^{(p)})$ при фиксированном $s^{(p)}>0^{(p)}$ из (2.14) получаем

$$\lim_{\varrho_p\downarrow 0}\omega(s^{(p-1)}, 0) = \lim_{\varrho_p\downarrow 0}\varrho_p\left\{1+\sum_{i=1}^{p}\Psi_i(s^{(p-1)}, 0)E_i(s^{(p-1)}, 0)\right\} = \bar{a}_p\Psi_p(s^{(p-1)}, 0),$$

что и доказывает лемму 9.

Для упрощения формулировки основного результата введем следующие обозначения:

$$Q_0(x^{(0)}) = 1,$$

$$Q_0(x^{(p-1)}) = \alpha(\overline{W}(x^{(p-1)}, \infty), W(x^{(p-1)}))\quad (p=\overline{2, r}),$$

где

$$\alpha(x, y)=\begin{cases}x, & \text{для схемы A},\\ y, & \text{для схем B},\end{cases}$$

а $\overline{W}(x^{(p-1)}, \infty)$ имеет своим п. Л.—С. функцию $\lim\limits_{\varrho_p\downarrow 0}\omega(s^{(p-1)}, 0)$.

Имеет место следующая

Теорема 3. Пусть имеют место разложения (3.1) и для схемы B1 выполнено условие (6.11). Тогда в условиях критической загрузки

$$\lim_{\varrho\downarrow 0}P\{w_i<x_i,\ w_jR_j<x_j(i=\overline{1, p-1};\ j=\overline{p, r})\} = \prod_{u=0}^{m}Q_u(x_u^{k_u})$$

где $k_u=p_{u+1}-p_u-\delta_{u0}^{u=0}$, $x_u^{(ku)}=(x_{p_u+\delta_{u0}}, \dots, x_{p_{u+1}-1}^{-1})$, а функции $Q_u(x_u^{ku})$ $(u=\overline{1, m})$ определяются своими п. Л.—С.

$$\prod_{k=p_u}^{p_{u+1}^{-1}}A_k(s_u^{k_u}).$$

Доказательство. Доказательство вначале проведем для схемы А.

Из соотношения (2.16), как и для введенной системы $\vec{M}_p|\vec{G}_p|1|\infty$ рекуррентно вычисляются

$$\lim_{\varrho\downarrow 0}\Psi_p(s^*) = \Psi_p(s^{(p-1)}, 0).$$

Обозначим $D_i(s)=\beta_{i1}^{-1}\Psi_i(s)$ $(i=\overline{p, r})$. Тогда из (2.9) имеем, что при $\varrho\downarrow 0$ $(k\geqq p;\ i=\overline{k+1, r})$

$$\beta_{i1}^{-1}V_k^{(i)}(s^*)\sim E_k(s^*). \tag{7.3}$$

В силу леммы 8 из (2.16) нетрудно получить, что $(p_u<i\leqq p_{u+1};\ u=\overline{1, m})$

$$\overline{D}_i(s)=\lim\varrho_{p_u}D_i(s^*)=\bar{c}_{p_ui-1}\varrho_{p-1}\overline{D}_p(s)\prod_{k=p}^{i-1}A_k, \tag{7.4}$$

где $\overline{D}_p(s)=\beta_{p1}^{-1}\Psi_p(s^{(p-1)}, 0)$.

Действительно, при $i=p$ соотношение (7.4) имеет место. Положим, что оно верно для $\bar{D}_n$ $(n=\overline{p,i})$ и докажем его справедливость для $\bar{D}_{i+1}(s)$.

а) Пусть $i<p_{u+1}$. Из (2.16) и (6.19) с учетом (7.3) имеем

$$\bar{D}_{i+1}(s)=\lim_{\varrho\downarrow 0}\varrho_{p_u}\left[\sum_{k=p_u}^{i-1}D_k(s^*)E_k(s^*)\beta_{k1}+D_i(s^*)E_i(s^*)\beta_{i1}\right]=$$

$$=\lim_{\varrho\downarrow 0}\varrho_{p_u}D_i(s^*)\left[1+\frac{\varrho_{i-1}-\varrho_i}{a_i}E_i(s^*)\right]=\bar{c}_{p_u i}\varrho_{p-1}\bar{D}_p(s)\prod_{k=p}^{i}A_k(s).$$

б) При $i=p_{u+1}$, учитывая, что $\bar{c}_{ii}=1$, из (2.16), (6.19) и (7.3) получаем

$$\bar{D}_{i+1}(s)=\lim_{\varrho\downarrow 0}\varrho_{p_{u+1}}D_i(s^*)E_i(s^*)\beta_{i1}=c_{i-1p_u}\bar{D}_i(s)A_i(s),$$

что вкупе с предположением индукции для $\bar{D}_i(s)$ дает (7.4).

Далее, из (2.15) при $p_m<r$ выводим

$$\bar{\Phi}(s)\stackrel{\text{def}}{=}\lim_{\varrho\downarrow 0}\varrho_r\Phi(s^*)=$$

$$\lim_{\varrho\downarrow 0}c_{rp_m}\varrho_{p_m}\left(\sum_{i=p_m}^{r-1}D_i(s^*)E_i(s^*)\beta_{i1}+D_r(s^*)\cdot E_r(s^*)\cdot\beta_{r1}\right)=\varrho_{p-1}\bar{D}_p(s)\cdot\prod_{i=p}^{r}A_i(s).$$

Если же $\bar{c}_r=0$, то согласно (6.19) и (7.4) имеем

$$\bar{\Phi}(s)=\lim\varrho_rD_r(s^*)\cdot E_r(s^*)\cdot\beta_{r1}=$$

$$=\bar{c}_{r-1p_{m-1}}\bar{D}_r(s)\cdot A_r(s)=\varrho_{p-1}\bar{D}_p(s)\prod_{i=p}^{r}A_i(s),$$

что и доказывает теорему для схемы А.

Перейдем к доказательству теоремы 3 в случае схем Bν ($\nu=1, 2, 3$). Легко видеть, что функции

$$\bar{\Phi}_i(s)=\lim_{\varrho\downarrow 0}\Phi_i(s^*)\quad(i=\overline{1,p-1})$$

вычисляются рекуррентно из (2.17) и (2.18), как и для введенной системы $\vec{M}_{p-1}|\vec{G}_{p-1}|1|\infty$.

Схема B1. В силу определения функций $T_i(s)$ $(i=\overline{1,r})$ и отношений эквивалентности (6.20), (6.19) легко показать, что

$$\lim_{\varrho\downarrow 0}\omega(s^*)=\varrho_{p-1}\prod_{i=1}^{p-1}\left(1+T_i(s^*)\right)\cdot\lim_{\varrho\downarrow 0}\prod_{i=p}^{r}c_i\left(1+T_i(s^*)\right)=$$

$$=\varrho_{p-1}\left(1+\bar{\Phi}_{p-1}(s)\right)\cdot\prod_{i=p}^{r}A_i(s).$$

Схемы Bν ($\nu=2, 3$). Нетрудно заметить, что при $\varrho\downarrow 0$, с учетом (2.18), (6.21) и (6.19), имеют место асимптотические соотношения:

$$\varrho_r(1+\Phi_r(s^*)) = \varrho_r\{1+\Phi_{r-1}(s^*)+E_r(s^*)[\Phi_{r-1}(s^*)\cdot M_r(s^*)+N_r(s^*)]\} \sim$$

$$\sim \varrho_r[1+\Phi_{r-1}(s^*)]\cdot\left[1+\frac{\varrho_{r-1}-\varrho_r}{a_r}\cdot E_r(s^*)\right] = \ldots =$$

$$= \varrho_{p-1}[1+\Phi_{p-1}(s^*)]\cdot \prod_{i=p}^{r} c_i\left[1+\frac{\varrho_{i-1}-\varrho_i}{a_i}E_i(s^*)\right] \sim \varrho_{p-1}[1+\bar{\Phi}_{p-1}(s)]\cdot\prod_{i=p}^{r} A_i(s),$$

что и доказывает теорему 3.

Приведем без доказательств два следствия из теоремы 3.

Следствие 1. При $\gamma_r=2$ из утверждения теоремы 3 вытекает основной результат работы [5].

Следствие 2. Пусть выполнены условия (3.1) и (6.11), а $m=r$. Тогда существует предел

$$\lim P\{w_1 < x_1, w_jR_j < x_j(j=\overline{2,r})\} = P\{w_1<x_1\}\prod_{k=2}^{r} R_{\bar{c}_k}(x_k)$$

где

а) при $\bar{c}_k=0$

$$R_0(x) = \begin{cases} 1-e^{-x}, \gamma_p=2 \\ \int_0^\infty G_{(\gamma_p-1)/2}(xt^{(\gamma_p-1)/2})\,d\,(1-e^{-t}), \quad 1<\gamma_p<2 \end{cases}$$

б) при $\bar{c}_k>0$

$$\int_0^\infty e^{-sx}\,d\,R_{\bar{c}_k}(x) = \frac{\bar{c}_k}{\bar{c}_k+\Delta^{\gamma_p-1}(s)}.$$

Здесь $G_\alpha(x)$ $(0<\alpha<1)$ есть стандартный положительный устойчивый закон с показателем α.

Multidimensional limit theorems for the waiting time in priority queues of $\vec{M}_r|\vec{G}_r|1|\infty$ type

E. A. Danielian and R. N. Chitchian

Let $w_k\,(k=\overline{1,r})$ be the customer's stationary waiting time of the k-th flow in the system $\vec{M}_r|\vec{G}_r|1|\infty$ with head-of-the-line and preemptive priority disciplines.

Assuming the existence of a finite variance for the distribution function of each priority class customers service time, the class of all possible limiting distributions of the vector $(w_1, \ldots, w_r)$ is described in heavy traffic and under some conditions.

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР
АКАДЕМИИ НАУК АРМЯНСКОЙ ССР
УЛ. П. СЕВАКА, 1
375044 ЕРЕВАН—44, СССР

Литература

[1] Клейнрок, Л., Вычислительные системы с очередями, *Мир, М*, 1979.
[2] Гнеденко, Б. В. и др., Приоритетные системы обслуживания, *МГУ*, 1973.
[3] Даниелян, Э. А., Анализ виртуального времени ожидания в некоторых приоритетных системах массового обслуживания, *ДАН Уз. СССР* (в печати), 1980.
[4] Даниелян, Э. А., Г. А. Попов, Об одной предельной теореме для приоритетных систем при единичной загрузке, *ДАН Арм. ССР*, том 70, №I. 1980.
[5] Даниелян, Э. А., Н. С. Земляной, Класс предельных распределений совместного стационарного распределения времен ожидания некоторых приоритетных систем $\vec{M}_r|\vec{G}_r|1|\infty$ в условиях критической загрузки, *ДАН Арм. ССР*, том 70, №1, 1980.
[6] Феллер, В., Введение в теорию вероятностей и ее приложения, том 2, *Мир, М*, 1967.

(Поступило 5-ого декабря 1980 г.)

Обобщенные преобразования Хаара и автоматические системы проверки качества печатных плат

С. С. Агаян, А. К. Матевосян

§ 1. Введение и постановка задачи

Одной из многочисленных областей применения распознавания образов является проверка качества печатных плат, которая может быть основана на сравнении результатов измерения параметров проверяемой платы и эталонного фотообразца. Эта задача приводит к необходимости разработки автоматических систем проверки качества плат. Основу таких систем должны составить автоматические системы распознавания образов.

Прежде чем приступить к синтезу искомой системы распознавания образов необходимо решить следующие задачи: а) выделения признаков, б) сжатия данных.

Ту и Гонсалес в работе [1] сводят эти задачи к задаче аппроксимации посредством разложения по некоторой системе функций. Задача разложения приводит, в свою очередь, к выполнению преобразования Фурье по выбранной базисной системе, т. е. к выполнению над исходными векторами f матричных преобразовании вида $F=\mathbf{A}\cdot f$ (прямое преобразование) и $f=\overline{\mathbf{A}}\cdot F$ (обратное преобразование), где $\mathbf{A}$-матрица, соответствующая базисной системе, $\overline{\mathbf{A}}$-обратная к ней матрица. Целесообразно выбирать критерием качества аппроксимации среднеквадратичную ошибку [1].

Таким образом, возникает задача выбора такой системы базисных функций, которая минимизирует среднеквадратичную ошибку, сосредотачивая большую часть энергии в первых n слагаемых разложения. Как указано Уинтцем (P. Wintz) [2] решение этой задачи аналогично решению задачи определения преобразования, обеспечивающего получение некоррелированных компонент. Этим условиям удовлетворяет преобразование Хотеллинга [3], которое приводит к некоррелированным компонентам и минимизирует среднеквадратичную ошибку, сосредотачивая максимум энергии в первых n компонентах [2]. Выполнение преобразования Хотеллинга для изображения, состоящего из $N\times N$, N-любое, точек требует выполнения N^4 операций умножения [2]. Наиболее близко по своим декоррелирующим свойствам к преобразованию Хотеллинга стоит ортогональное преобразование Фурье [4]. Преимущество преобразования Фурье по отношению к преобразованию Хотеллинга состоит в существовании быстрого алгоритма выполнения преобразования за $N^2 \log_2 N^2$ операций комплексного умножения. Этим преобразованиям несколько уступают по критерию среднеквадратичной ошибки ортогональные преобразования Уолша (Walsh) [5], Хаара (Haar) [5—6] и наклонное (slant) [7]. Наилучшим из них по своим декоррелирующим свойствам является slant — преобразование, наихудшим — Хаара. Преобразования slant, Уолша и Хаара требуют выполнения $4N(N-2)$, $N=2^k$ операций умножения, $2N^2 \log_2 N$, $4N(N-1)$, $N=2^k$, операций сложения и вычитания соответственно. Таким образом, преобразование Хаара наиболее привлекательно с точки зрения аппаратурной реализации [4], скорости вычисления, хотя и не совсем хорошо декоррелирует данные, что, по-видимому, обусловлено большой разреженностью матриц Хаара. Недостатком последних трех преобразований является и то, что порядки преобразуемых массивов N должны равняться степени двойки.

Таким образом, при выборе базисной системы имеется две альтернативы: выбрать систему, обеспечивающую высокое качество декорреляции, но при этом потерять в скорости и времени обработки информации либо повысить скорость обработки за счет ухудшения качества. Анализ приведенных фактов натолкнул авторов на идею обобщения преобразования Хаара с двух различных позиций. В настоящей работе построены гибридные ортогональные системы Фурье—Хаара, slant—Хаара, Адамара—Хаара, которые по своим скоростным и декоррелирующим качествам являются промежуточными относительно систем Хаара и Фурье, slant, Адамара соответственно: лучше декоррелируют данные, чем классическое преобразование Хаара, и в то же время обладают достаточно быстрыми алгоритмами, близкими по скорости выполнения к преобразованию Хаара и очень просты с точки зрения реализации.

§ 2. Обобщенные матрицы и функции Хаара первого типа

2.1. Определение обобщенных матриц и функций Хаара первого типа. В 1910 году Хааром [8] была описана система $\{X(k, x)\}_{k=0}^{\infty}$ ортогональных на отрезке $[0, 1]$ функций, обладающая следующими свойствами: любая непрерывная на отрезке $[0, 1]$ функция $f(x)$ разлагается в равномерно сходящийся ряд по функциям системы:

$$f(x) = \sum_{k=0}^{\infty} X(k, x) c_k$$

где $c_k = \int_0^1 f(t) X(k, t)\, dt$, а $X(0, x) \equiv 1$

$$X(2^m+j, x) = \begin{cases} \sqrt{2^m}, & \text{для} \quad x \in \left[\dfrac{j}{2^m}, \dfrac{j+1/2}{2^m}\right) \\ -\sqrt{2^m}, & \text{для} \quad x \in \left[\dfrac{j+1/2}{2^m}, \dfrac{j+1}{2^m}\right) \\ 0, & \text{в остальных точках} \end{cases} \tag{2.1}$$

$$m = 0, 1, \ldots, \quad j = 0, 1, \ldots, 2^{m-1}.$$

Система функций Хаара состоит из подсистем. Подсистема номер m содержит 2^m функций Хаара $X(2^m+j, x)$ $j=0, 1, \dots, 2^m-1$. Функции Хаара m-ой подсистемы постоянны на полуинтервалах, получающихся от деления отрезка [0, 1] на 2^{m+1} равных частей.

На рис. 1 изображены первые 8 функций Хаара.

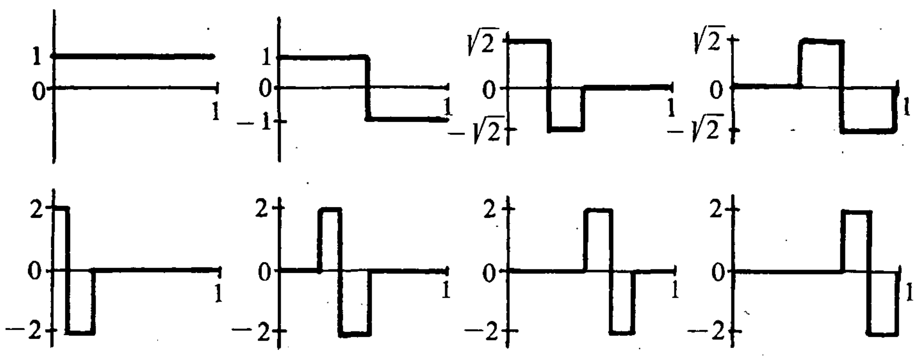


Рис. 1

Как указано в [8], можно строить системы функций, аналогичные $\{X(k, x)\}_{k=0}^{\infty}$, деля отрезок [0, 1], а затем и каждый из получающихся отрезков на две неравные части, лишь бы множество точек деления было бы всюду плотным на [0, 1]. Такие системы называются системами типа Хаара.

Матрица Хаара $\mathbf{HA}_{2^n}$ является квадратной матрицей порядка 2^n, элементы которой равны $\pm\sqrt{2^m}$ или 0 (m-фиксировано в строке), в каждой строке происходит одна смена знака и строки ортогональны, т. е. выполняется

$$\mathbf{HA}_{2^n} \cdot \mathbf{HA}_{2^n}^T = 2^n \mathbf{I}_{2^n} \tag{2.2}$$

где $\mathbf{I}_N$-единичная матрице порядка N.

Матрицы Хаара порядка $N=2^n$ получаются равномерной выборкой функций Хаара. Приведем матрицу Хаара порядка 8.

$$\mathbf{HA}_{2^3} = \begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{Vmatrix}$$

Вопросы, связанные с построением ортогональных матриц в более общем случае, соответствующем выбору функций типа Хаара, не исследовались.

Для матриц Хаара $\mathbf{HA}_{2^n}$ со стороны Файно [9] получены рекуррентные соотношения, связывающие подматрицы матриц $\mathbf{HA}_{2^n}$, $n=1, 2, \dots$. Мы приведем несколько иное рекуррентное построение $\mathbf{HA}_{2^n}$, позволяющее обобщить матрицы Хаара:

$$\mathbf{HA}_2 = \begin{Vmatrix} 1 & 1 \\ 1 & -1 \end{Vmatrix}.$$

$$\mathbf{HA}_{2^n} = \begin{Vmatrix} \mathbf{HA}_{2^{n-1}} \otimes [1, 1] \\ \sqrt{2^{n-1}}\, \mathbf{I}_{2^{n-1}} \otimes [1, -1] \end{Vmatrix} \quad n = 2, 3, \ldots . \tag{2.3}$$

Зададим следующую рекуррентную формулу построения квадратных матриц $\mathbf{A}_{k^n}$ порядка k^n, обобщающую (2.3):

$$\mathbf{A}_{k^n} = \begin{Vmatrix} \mathbf{A}_{k^{n-1}} \otimes e_k \\ \sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}'_k \end{Vmatrix} \tag{2.4}$$

где e_k-вектор-строка из k единиц; $\mathbf{A}'_k$ матрица составленная из последних $(k-1)$ строк матрицы $\mathbf{A}_k$, $\otimes$-кронекерово произведение [13]. Выясним, при каких условиях матрицы $\mathbf{A}_{k^n}$ будут ортогональными.

Справедлива следующая

Теорема. Пусть $\mathbf{A}_k$-квадратная матрица порядка k, удовлетворяющая условиям

$$\mathbf{A}_k \mathbf{A}_k^T = k\mathbf{I}_k, \quad e_k \mathbf{A}_k'^T = O_{k-1} \tag{2.5}$$

(O_k-вектор строка из k нулей) тогда формула (2.4) порождает ортогональные матрицы порядка k^n.

Доказательство. Проведем индукцией по n. Для $n=1$ ортогональность $\mathbf{A}_k$ имеем из условия (25). Предположим справедливость утверждения теоремы для $n-1$, т. е.

$$\mathbf{A}_{k^{n-1}} \cdot \mathbf{A}_{k^{n-1}}^T = k^{n-1} \mathbf{I}_{k^{n-1}}.$$

Докажем ортогональность матрицы $\mathbf{A}_{k^n}$.

$$\mathbf{A}_{k^n} \cdot \mathbf{A}_{k^n}^T = \begin{Vmatrix} R_1 & R_2 \\ R_2^T & R_3 \end{Vmatrix}$$

где

$$R_1 = (\mathbf{A}_{k^{n-1}} \otimes e_k) \cdot (\mathbf{A}_{k^{n-1}} \otimes e_k)^T,$$

$$R_2 = (\mathbf{A}_{k^{n-1}} \otimes e_k) \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}'_n\right)^T,$$

$$R_3 = \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}'_k\right) \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}'_k\right)^T.$$

Из свойств кронекерского произведения, условий (2.5) и предположения ин дукции имеем:

$$R_1 = (\mathbf{A}_{k^{n-1}} \otimes e_k)(\mathbf{A}_{k^{n-1}}^T \otimes e_k^T) = (\mathbf{A}_{k^{n-1}} \mathbf{A}_{k^{n-1}}) \otimes (e_k e_k^T) = kk^{n-1} \mathbf{I}_{k^{n-1}} = k^n \mathbf{I}_{k^{n-1}},$$

$$R_2 = (\mathbf{A}_{k^{n-1}} \otimes e_k) \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}_k'^T\right) =$$

$$= \sqrt{k^{n-1}}\, (\mathbf{A}_{k^{n-1}} I_{k^{n-1}}) \otimes (e_k \mathbf{A}_k'^T) = \sqrt{k^{n-1}}\, \mathbf{A}_{k^{n-1}} \otimes O_{k-1} = O_{k^{n-1}}^T \otimes O_{(k-1)k^{n-1}},$$

$$R_3 = \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}'_k\right) \cdot \left(\sqrt{k^{n-1}}\, \mathbf{I}_{k^{n-1}} \otimes \mathbf{A}_k'^T\right) =$$

$$= k^{n-1} \mathbf{I}_{k^{n-1}} \otimes (\mathbf{A}'_k \mathbf{A}_k'^T) = k^{n-1} \mathbf{I}_{k^{n-1}} \otimes k\mathbf{I}_{k-1} = k^n \mathbf{I}_{(k-1)k^{n-1}}.$$

Таким образом

$$\mathbf{A}_{k^n}\mathbf{A}_{k^n}^T = \begin{Vmatrix} k^n I_{k^n-1} & O_{k^n-1}^T \otimes O_{(k-1)k^n-1} \\ O_{k^n-1} \otimes O_{(k-1)k^n-1}^T & k^n \mathbf{I}_{(k-1)k^n-1} \end{Vmatrix} = k^n \mathbf{I}_{k^n}.$$

Теорема доказана.

Условию (2.5) удовлетворяют

1. матрицы slant, Уолша и Хаара, если $k=2^m$,
2. нормализованные матрицы Адамара [13], если $k=4t$,
3. матрицы Фурье, если k-произвольное натуральное число.

Подставляя указанные матрицы в качестве исходной $\mathbf{A}_k$ в формулу (2.4) будем получать ортогональные матрицы, названные нами гибридными матрицами Хаара-slant, Уолша—Хаара, Адамара—Хаара и Фурье—Хаара, поскольку они по структуре близки к матрицам Хаара, но в то же время содержат в себе функции названных систем.

Отметим также, что система Фурье—Хаара является комплексной системой, что дает некоторые преимущества при решении определенных задач, связанных с обработкой комплексных данных.

В некоторых задачах обработки информации предъявляются высокие требования к скорости проведения преобразования. Если же k такое, что мы вынуждены обратиться к матрицам Фурье, это снижает скорость преобразования по сравнению, например, с преобразованием Уолша—Хаара для $k=2^m$.

Мы приведем такой метод построения матрицы $\mathbf{A}_k$ удовлетворяющей (2. 5), k-любое, которое позволяет получить алгоритмы, близкие по скорости к преобразованию Хаара.

Лемма. Для любого натурального числа k можно построить квадратную матрицу $\mathbf{A}_k$ порядка k с элементами $\lambda_i a_{ij}$, a_{ij}-целые числа, удовлетворяющую (2.5).

Доказательство. Пусть $k=2^p+2^q$, $p>q$. Положим

$$\mathbf{A}_k = \{\overline{a_{ij}}\} = \begin{Vmatrix} & e_k & \\ e_{2^p} & & ce_{2^q} \\ \mathbf{HA}'_{2^p} & & 0 \\ 0 & & \mathbf{HA}'_{2^q} \end{Vmatrix} \tag{2.6}$$

где $c=-\frac{2^p}{2^q}$ -целое число.

Поскольку любое число k можно представить в виде

$$k = 2^p \cdot \varepsilon_p + 2^{p-1}\varepsilon_{p-1} + \ldots + \varepsilon_0,$$

где $\varepsilon_i=\begin{cases}1\\0\end{cases}$ и $\varepsilon_p=1$, то аналогичное (2.6) построение можно провести для любого k.

Положим $\mathbf{A}_k=\{\lambda_i\overline{a_{ij}}\}$, где нормировочные множители λ_i выбираются из условия

$$\sum_{j=1}^{k} \lambda_i^2 \overline{a_{ij}}^2 = k, \quad i = 1, 2, \ldots, k.$$

Легко проверяется, что матрица $\mathbf{A}_k$ удовлетворяет (2.5). Лемма доказана.

Приведем в качестве примера A_3 и A_{3^2}:

$$A_3=\begin{Vmatrix} 1 & 1 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -\sqrt{2} \\ \frac{\sqrt{6}}{2} & -\frac{\sqrt{6}}{2} & 0 \end{Vmatrix}$$

$$A_{3^2}=\begin{Vmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & -\sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & -\frac{\sqrt{6}}{2} & -\frac{\sqrt{6}}{2} & -\frac{\sqrt{6}}{2} & 0 & 0 & 0 \\ \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & -\sqrt{6} & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{\sqrt{18}}{2} & -\frac{\sqrt{18}}{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & -\sqrt{6} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\sqrt{18}}{2} & -\frac{\sqrt{18}}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{6}}{2} & \frac{\sqrt{6}}{2} & -\sqrt{6} \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{\sqrt{18}}{2} & -\frac{\sqrt{18}}{2} & 0 \end{Vmatrix}.$$

Вернемся к формуле (2.4). Полученные по (2.4) ортогональные матрицы назовем обобщенными матрицами Хаара первого типа и будем обозначать GI_N. Структура GI_N максимально приближена к структуре HA_{2^n}. GI_{k^n} состоит из n подматриц. При переходе от GI_{k^n} к $GI_{k^{n+1}}$ добавляется n-ая подматрица, содержащая $(k-1)k^n$ строк.

Основываясь на построенных матрицах Хаара, введем системы ортогональных на отрезке [0, 1] функций. Обозначим через $g(p, q)$ элементы матрицы $GI_{k^{m+1}}$.

Определим функции

$$XI(k^m+j, x) = g(k^m+j, p), \quad \text{если} \quad x \in \left[\frac{p}{k^{m+1}}, \ \frac{p+1}{k^{m+1}}\right)$$

$$j = 0, 1, \ldots, (k-1)k^m-1, \quad m = 0, 1, \ldots .$$

Введенную систему назовем обобщенной системой Хаара первого типа. Первые шесть функций, полученных на основе GI_{3^n} матриц приведены на рис. 2.

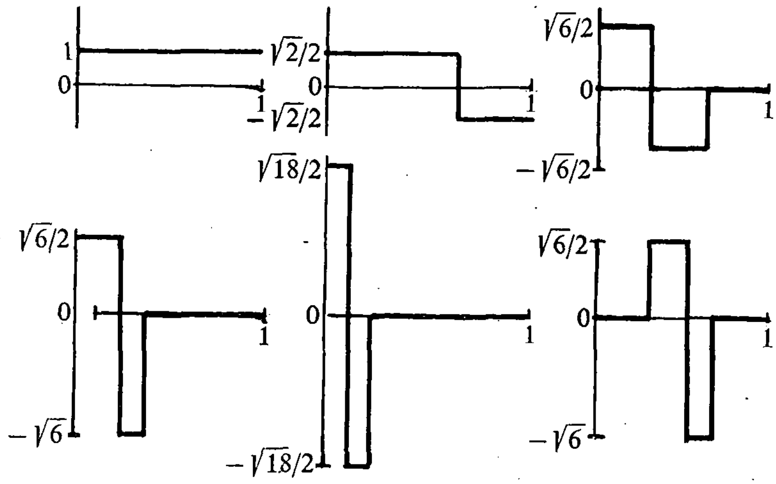


Рис. 2

Отметим, что при выборе $\mathbf{A}_k$ по (2.6) функции $XI(k^m+j, x)$ являются функциями типа Хаара, а соответствующие им матрицы — ортогональными матрицами типа Хаара.

2.2. Свойства обобщенных матриц и функций Хаара 1 типа.

1. Обобщенные матрицы Хаара первого типа ортогональны, т. е. удовлетворяют условию:

$$\mathbf{GI}_N \cdot \mathbf{GI}_N^T = N\mathbf{I}_N.$$

2. Обобщенные функции Хаара первого типа ортогональны на отрезке [0, 1], т. е. удовлетворяют

$$\int_0^1 XI(k^m+j, x) XI(k^n+i, x)\, dx = \begin{cases} 1, & \text{если} \quad m = n,\ j = i \\ 0, & \text{если} \quad m \neq n \quad \text{или} \quad j \neq i. \end{cases}$$

3. Если выбрать $\mathbf{A}_k$ по (2.6), то $\mathbf{GI}_{k^n}$ переходят в матрицы типа Хаара [8], а соответствующие им функции являются мартингалом [10].

4. Элементы гибридных матриц Уолша—Хаара, Адамара—Хаара и матрицы типа Хаара имеют вид $\lambda_i a_{ij}$, где a_{ij}-целые числа, λ_i-нормировочный множитель.

5. Матрицы $\mathbf{GI}_{k^n}$ при $k=2$ совпадают с матрицей $\mathbf{HA}_{2^n}$.

6. Матрицы $\mathbf{GI}_{k^n}$ при $k=2^m$ и выборе $\mathbf{A}_k$ матрицами Хаара связаны матричными соотношениями с матрицами Адамара $\mathbf{HD}_{2^n}$

$$\mathbf{HD}_{2^{mn}} = \begin{Vmatrix} 1 & 0 \\ 0 & p' \end{Vmatrix} \cdot GI_{2^{mn}}$$

где

$$p' = \operatorname{diag}\left\{p, \frac{1}{\sqrt{2^m}} \mathbf{HD}_{2^m} \otimes p, \ldots, \frac{1}{\sqrt{2^{m(n-1)}}} \cdot \mathbf{DH}_{2^{m(n-1)}} \otimes p\right\}$$

а

$$p = \operatorname{diag}\left\{1, \frac{1}{\sqrt{2}}\mathbf{HD}_2, \ldots, \frac{1}{\sqrt{2^{m-1}}}\cdot\mathbf{HD}_{2^{m-1}}\right\}$$

которое при $n=1$ или $m=1$ переходит в матричное соотношение между матрицами Адамара и классическими матрицами Хаара, полученное Файно [9].

7. Пусть $f(x)$-интегрируемая на отрезке [0, 1] функция, а $\varphi_n(x)$ перенумерованные функции $\mathbf{GI}_{k^n}$ системы. Составим ряд

$$\sum_{n=0}^{\infty} c_n\varphi_n(x),$$

где

$$c_n = \int_0^1 f(t)\varphi_n(t)\,dt.$$

Через $S_m(f, x)$ обозначим частичную сумму ряда

$$S_m(f, x) = \sum_{n=0}^{m} c_n\varphi_n(x).$$

Справедливы

Теорема. Пусть функция $f(x)$ интегрируема на отрезке [0, 1] тогда:

1. $S_{m_n}(f, x)$ сходится $f(x)$ почти во всех точках [0, 1], $m_n=1+(k-1)n$,
2. в точке x_0 непрерывности $f(x)$ $S_{m_n}(f, x)$ сходится к $f(x_0)$, $m_n=$ $=1+(k-1)n$,
3. если $f(x)$ непрерывна на отрезке [0, 1], то $S_{m_n}(f, x)$ сходится к $f(x)$ равномерно по x.

Теорема. Если $f(x)\in \mathrm{Lip}\,(\alpha, L)$, то

$$|f(x)-S_{m_n}(f, x)| \leqq \frac{L}{\alpha+1}\left(\frac{1}{m_n}\right)^{\alpha}$$

для всех

$$x\in[0, 1], \quad m_n = k^n.$$

2.3. Алгоритмы быстрых преобразований по $\mathbf{GI}_{k^n}$ матрицам. В настоящем пункте будут приведены алгоритмы выполнения прямого

$$F = \mathbf{GI}_{k^n}\cdot f^T \tag{2.7}$$

и обратного

$$f = \frac{1}{k^n}\mathbf{GI}_{k^n}^T\cdot F^T \tag{2.8}$$

преобразований, где f-исходный вектор порядка k^n, F-преобразование вектора f.

Алгоритм прямого преобразования. Приведем два быстрых алгоритма прямого преобразования (2.7).

1. Используя свойства кронекерского произведения и на основании (2.4) можно получить, что

$$GI_{k^n}=\begin{Vmatrix}A_k\cdot(I_k\otimes e_{k^{n-1}})\\ \sqrt{k}(I_k\otimes A_k')\cdot(I_{k^2}\otimes e_{k^{n-2}})\\ \vdots\\ \sqrt{k^{n-1}}\,I_{k^{n-1}}\otimes A_k'\end{Vmatrix}=\begin{Vmatrix}A_k & 0 & \cdots & 0\\ 0 & I_k\otimes A_k' & \cdots & 0\\ \vdots & \vdots & & \vdots\\ 0 & 0 & & I_{k^{n-1}}\otimes A_k'\end{Vmatrix}\cdot\begin{Vmatrix}I_k\otimes e_{k^{n-1}}\\ I_{k^2}\otimes e_{k^{n-2}}\sqrt{k}\\ \vdots\\ \sqrt{k^{n-1}}\,I^{k^n}\end{Vmatrix}$$

откуда следует существование быстрого алгоритма для выполнения (2.7).

2. Приведем иное разложение матрицы GI_{k^n} в произведение слабозаполненных матриц, которое и задаст второй алгоритм быстрого выполнения преобразований (2.7).

Теорема. Пусть матрица A_{k^n} построена по (2.4), тогда имеет место разложение:

$$A_{k^n}=R_1R_2\dots R_n \tag{2.9}$$

где

$$R_n=\begin{Vmatrix}I_{k^{n-1}}\otimes e_k\\ \sqrt{k^{n-1}}\,I_{k^{n-1}}\otimes A_k'\end{Vmatrix},\quad R_i=\begin{Vmatrix}I_{k^{i-1}}\otimes e_k & 0\\ \sqrt{k^{i-1}}\,I_{k^{i-1}}\otimes A_k' & 0\\ 0 & I_{k^n-k^i}\end{Vmatrix},\ i=1,2,\dots,n-1. \tag{2.10}$$

Доказательство приводится непосредственной проверкой. Действительно, нижняя часть R_n представляет собой последнюю подматрицу матрицы A_{k^n}. Умножив $R_{n-1}\cdot R_n$, получим

$$R_{n-1}\cdot R_n=\begin{Vmatrix}I_{k^{n-2}}\otimes e_{k^2}\\ \sqrt{k^{n-2}}\,I_{k^{n-2}}\otimes A_k'\otimes e_k\\ \sqrt{k^{n-1}}\,I_{k^{n-1}}\otimes A_k'\end{Vmatrix}$$

т. е. предпоследняя подматрица тоже восстановлена и т. д. Таким образом, справедливость разложения (2.9) доказана.

Теперь преобразование (2.7) сведется к последовательным умножениям матриц R_i на вектора. Если обозначить через $D_0(1)$ и $D_1(1)$ число операций сложения и умножения соответственно, требуемое для выполнения преобразования A_kf^T, то на выполнение этих последовательных умножений потребуется (умножения на $\sqrt{k^{n-i}}$ отложим до обратного преобразования, как было сделано и для классического преобразования Хаара [5]): для R_i — $k^{i-1}D_0(1)$, $i=1,2,\dots,n$ операций сложения и $k^{i-1}D_1(1)$ операций умножения. Всего на выполнение преобразования требуется $D_0(1)\sum_{i=1}^{n}k^{i-1}=$ $=D_0(1)\dfrac{k^n-1}{k-1}$ операций сложения и $D_1(1)\dfrac{k^n-1}{k-1}$ операций умножения.

В частном случае, при $k=2$, алгоритм переходит в известный алгоритм быстрого преобразования Хаара [5, 6].

На рис. 3 приводится граф быстрого преобразования для гибридной матрицы Фурье—Хаара порядка 9. На графе $W=\exp\left\{-\dfrac{2\pi i}{3}\right\}$ ребра графа с коэффициентами означают перенос сигнала, умноженного на соответствующий коэффициент. Коэффициенты, равные 1, не указаны. Коэффициенты представляют собой элементы начальной матрицы фурье

$$A_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & W & W^2 \\ 1 & W^2 & W^4 \end{bmatrix}.$$

Приведем подпрограмму быстрого преобразования FGHR по гибридным матрицам $\mathbf{GI}_{k^n}$. R обозначает преобразование Уолша, Адамара или Хаара. Подпрограмма использует программы быстрых FR преобразований, команды обращения к которым имеют вид CALL FR(K, $F\emptyset$, F). В подпрограммах $F\emptyset$ обозначает исходный массив, F-массив для записи конечного результата, A-рабочий массив.

```
      SUBROUTINE FGHR(K, M, N, FØ, F)
      DIMENSION FØ(N), A(K), F(N)
    1 M=M-1
      M1=K**M
      DO 2 J=1, M1
      DO 3 I=1, K
    3 A(I)=FØ((J-1)*K+1)
      CALL FR(K, A, A)
      FØ(J)=A(1)
      DO 2 I=2, K
    2 F(M1+(K-1)*(J-1)+I-1)=A(I)
      IF(M.EQ.Ø)GO TO 4
      GO TO 1
    4 F(1)=FØ(1)
      RETURN
      END
```

в) Алгоритм обратного преобразования.

Поскольку матрица $\mathbf{GI}_{k^n}$ не симметрична, то необходимо построить алгоритм обратного преобразования (2.8), который будет отличаться от прямого преобразования.

Разложение (2.9) и задает алгоритм и для обратного преобразования. В этом случае матрица преобразования

$$\mathbf{GI}_{k^n}^T = R_n^T \cdot R_{n-1}^T \dots R_1^T. \tag{2.11}$$

Оценим теперь количество операций обратного преобразования. Обозначим через $D_0(1)$ и $D_1(1)$ количество операций сложения и умножения, необходимое на выполнение преобразования $A_k^T f^T$. Тогда на умножение на матрицу R_i^T $i=1, 2, \dots, n$ требуется:

$k^{i-1}D_0(1)$ операций сложения и

$k^{i-1}D_1(1)$ операций умножения.

Таким образом, на выполнение обратного преобразования необходимо

$D_0(1)\dfrac{k^n-1}{k-1}$ операций сложения и

$D_1(1)\dfrac{k^n-1}{k-1}$ операций умножения.

На рис. 4 приведен граф обратного быстрого гибридного преобразования Фурье—Хаара для матрицы порядка 9.

Приведенный алгоритм обратного преобразования обобщенных матриц Хаара при $k=2$ совпадает с алгоритмом быстрого преобразования классических матриц Хаара.

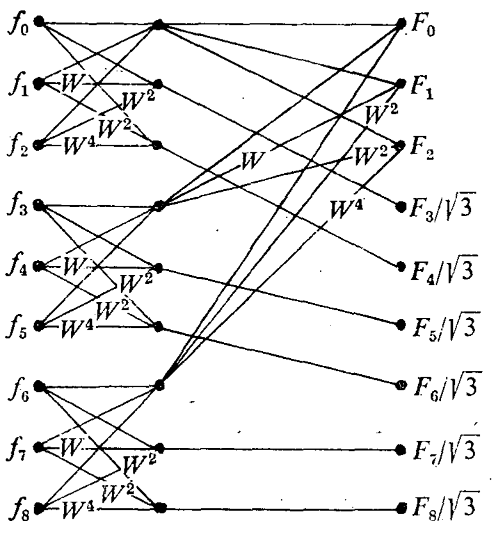


Рис. 3
Граф быстрого гибридного преобразования Фурье—Хаара для матрицы порядка 9

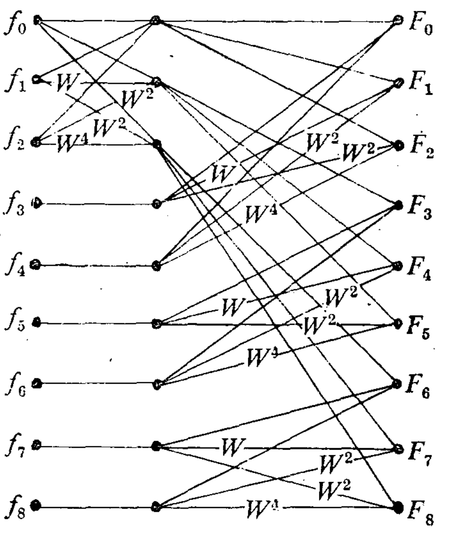


Рис. 4
Граф быстрого обратного гибридного преобразования Фурье—Хаара для матрицы порядка 9

Приведем подпрограмму быстрого обратного гибридного преобразования FINGHR. Как и подпрограмма прямого преобразования, использует подпрограммы для обратных R-преобразований FINR.

```
  SUBROUTINE FINGHR (K, M, N, FØ, F)
  DIMENSION FØ(N), A(K), F(N)
  I1=1
  DO 1 I=1, N
1 F(I)=FØ (I)
  DO 2 I=1, M
  I2=K**(M-I)
  DO 5 J=1, I1
  A(1)=Ø
  DO 4 L=2, K
4 A(L)=F(I1+(K-1)*(J-1)+L-1)
  CALL FINR (K, A, A)
  DO 5 N2=1, K
  A(N2)=A(N2)+F(I2*(N2-1)+1)
  DO 5 L=1, I2
5 F(I2*(N2-1)+L)=A(N2)
2 I1=I1*K
  RETURN
  END
```

Поскольку в подпрограммах FGHR и FINGHR не выполнялись нормировочные умножения на $\sqrt{k^{n-i}}$, приводится подпрограмма GHNORM, которая применяется для соответствующей нормировки массива A длинной $N=K^M$ и должна выполняться после прямого или обратного преобразований.

```
  SUBROUTINE HNORM (K, M, A)
  DIMENSION A(N)
  DO 1 I=1, K
1 A(I)=A(I)/K**M
  DO 2 II=2, M
  I=II-1
  A=1./K**(M-I)
  JMIN=K**I+1
  JMAX=K**II
  DO 2 J=JMIN, JMAX
2 A(J)=A(J)*A
  RETURN
  END
```

§ 3. Обобщенные матрицы и функции Хаара второго типа

3.1. Определение обобщенных матриц и функций Хаара второго типа. Перейдем ко второму обобщению матриц Хаара $\mathbf{HA}_{2^n}$. Обобщенными матрицами Хаара второго типа назовем квадратные ортогональные матрицы, состоящие из 0 и $\pm\sqrt{2^m}$, m-натуральное число, фиксированное в строке, и будем обозначать $\mathbf{GII}_N$.

Определение [12]. Квадратные $(0, \pm 1)$ матрицы X_i, $i=1, 2, \ldots, s$ порядка k назовем s-элементным гиперкаркассом, если они удовлетворяют условиям:

1. $X_i * X_j = 0 \quad i \neq j \quad i, j = 1, 2, \ldots, s$

где $*$-адамарово произведение [13],

2. $\sum_{i=1}^{s} X_i \quad - \quad (-1, +1)$ матрица,

3. $X_i X_j^T = X_j X_i^T$,

4. $\sum_{i=1}^{s} X_i X_i^T = k\mathbf{I}_k$.

Определение [13]. Матрицы **A** и **B** называются специальными, если

$$\mathbf{AB}^T = -\mathbf{BA}^T.$$

Лемма. Матрицы $\mathbf{HA}_{2^n}$ и $S_{2^n}\mathbf{HA}_{2^n}$, где

$$S_{2^n} = \mathbf{I}_{2^{n-1}} \otimes \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad \text{или} \quad S_{2^n} = \mathbf{I}_{2^{n-1}} \otimes \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

являются специальными матрицами Хаара.

В работах [11, 12] приводятся методы построения s-элементных, $s=2, 4$, гиперкаркассов.

Теорема. Пусть $\mathbf{HA}_{01}$ и $\mathbf{HA}_{02}$-специальные матрицы Хаара порядка p_0, а X_1 и X_2 2-элементный гиперкаркас порядка p_1, тогда

$$\mathbf{GII}_{0,i} = \mathbf{HA}_{0,i}, \quad i = 1, 2$$

$$\left.\begin{aligned} \mathbf{GII}_{n,1} &= X_1 \otimes \mathbf{GII}_{n-1,1} + X_2 \otimes \mathbf{GII}_{n-1,2} \\ \mathbf{GII}_{n,2} &= X_1 \otimes \mathbf{GII}_{n-1,2} - X_2 \otimes \mathbf{GII}_{n-1,1} \end{aligned}\right\} n = 1, 2, \tag{3.1}$$

являются специальными обобщенными матрицами Хаара второго типа порядка $p_0 p_1^n$.

Обобщенные матрицы Хаара второго типа порядка $N = p_0 p_1^{\alpha_1} \ldots p_m^{\alpha_m}$ строятся следующим образом. Выбирается матрица Хаара порядка p_0, гиперкаркас $\{X_1, X_2\}$ порядка p_1 и α_1 раз применяются формулы (3.1). Затем построенная $\mathbf{GII}_{n,1}$ порядка $p_0 p_1^{\alpha_1}$ берется в качестве исходной, выбирается гиперкаркасс порядка p_2 и α_2 раз применяется (3.1) и т. д.

Приведем в качестве примера обобщенную матрицу Хаара второго типа порядка 8.

$$\mathbf{GII}_8 = \left\| \begin{array}{cccccccc} + & + & + & + & + & + & - & - \\ + & + & - & - & - & - & - & - \\ \sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} & -\sqrt{2} \\ 0 & 0 & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & \sqrt{2} & 0 & 0 \\ + & + & - & - & + & + & + & + \\ - & - & - & - & + & + & - & - \\ 0 & 0 & \sqrt{2} & -\sqrt{2} & \sqrt{2} & -\sqrt{2} & 0 & 0 \\ -\sqrt{2} & \sqrt{2} & 0 & 0 & 0 & 0 & \sqrt{2} & -\sqrt{2} \end{array} \right\|,$$

Для определения счетной системы функций Хаара второго типа несколько модифицируем построение (3.1). Определим обобщенные матрицы Хаара второго типа $\mathbf{GII}_{n,i}$ $i=1, 2$. Пусть первая строка X_1, (X_1 и X_2 2-элементный гиперкаркасс) состоит из $+1$, а R_n-квадратная матрица порядка n с элементами r_{ij}.

$$r_{ij} = \begin{cases} 1, & \text{если} \quad j = (i-1)p + l(i) \\ 0, & \text{если} \quad j \neq (i-1)p + l(i) \end{cases}$$

$l(i)=k$, если $i \in [(k-1)p_1^{n-1}p_0,\, kp_0p_1^{n-1}]$, $k=1, 2, \ldots, p$.

Положим

$$\begin{aligned} \overline{\mathbf{GII}}_{0,i} &= \mathbf{GII}_{0,i}, \quad i = 1, 2 \\ \mathbf{GII}_{n,1} &= \overline{\mathbf{GII}}_{n-1,1} \otimes X_1 + \overline{\mathbf{GII}}_{n-1,2} \otimes X_2 \\ \mathbf{GII}_{n,2} &= \overline{\mathbf{GII}}_{n-1,2} \otimes X_1 - \overline{\mathbf{GII}}_{n-1,1} \otimes X_2 \\ \overline{\mathbf{GII}}_{n,i} &= R_n \cdot \mathbf{GII}_{n,i} \quad i = 1, 2, \quad n = 1, 2, \ldots \end{aligned} \tag{3.2}$$

Определим функции Хаара второго типа по следующему:

$$XII(m, x) = g(m, n), \quad \text{если} \quad x \in \left[\frac{n-1}{p_0 p_1^k};\ \frac{n}{p_0 p_1^k}\right)$$

где $g(m, n)$-элементы матрицы $\overline{\mathbf{GII}}_{p_0 p_1^k}$ порядка $p_0 p_1^k \geqq m$.

В качестве примера приведем $\mathbf{GII}_8$ и $\overline{\mathbf{GII}}_8$, полученные по (3.2),

$$\mathbf{GII}_8 = \left\| \begin{array}{cccccccc} + & + & + & + & + & + & + & + \\ + & - & + & - & - & + & - & + \\ + & + & + & + & - & - & - & - \\ - & + & - & + & - & + & - & + \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & \sqrt{2} \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ -\sqrt{2} & \sqrt{2} & \sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \end{array} \right\|$$

$$\overline{\mathrm{GII}_8} = \begin{Vmatrix} + & + & + & + & + & + & + & + \\ + & + & + & + & - & - & - & - \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ + & - & + & - & - & + & - & + \\ - & + & - & + & - & + & - & + \\ 0 & 0 & 0 & 0 & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & \sqrt{2} \\ -\sqrt{2} & \sqrt{2} & \sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \end{Vmatrix}$$

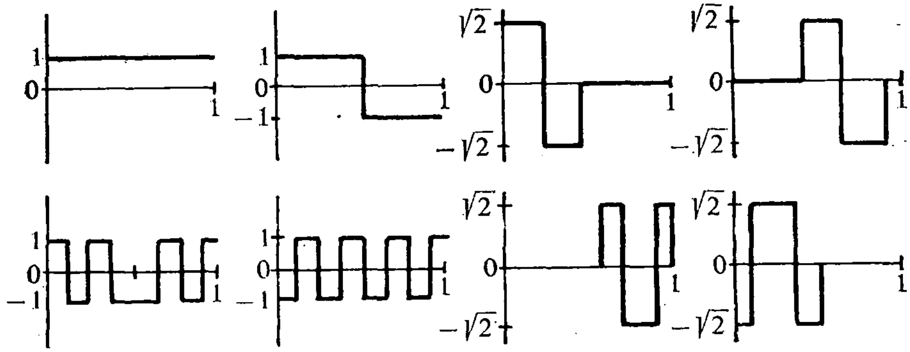


Рис. 5

На рис. 5 приводится функция Хаара второго типа, соответствующие матрице $\overline{\mathrm{GII}_8}$.

Вернемся к построению (3.1). Обобщен ные матрицы Хаара второго типа близки по структуре к матрицам Адамара, построенным в работе [11], строки же матрицы GII_N представляют собой склеенные функции Хаара. Это уже обобщение с другой позиции, в отличие от GI_N, структура которых совпала со структурой матриц Хаара. Естественно ожидать, что GII_N, в силу близости к матрицам Адамара, будут декоррелировать данные лучше классического преобразования Хаара.

3.2. Свойства обобщенных матриц и функций Хаара второго типа.

1. Матрицы Хаара второго типа ортогональны, т. е.

$$\mathrm{GII}_N \cdot \mathrm{GII}_n^T = N\mathrm{I}_N.$$

2. Функции Хаара второго типа $\{XII(m, x)\}_{m=0}^{\infty}$ образуют ортогональную систему функций, т. е.

$$\int_0^1 XII(m, x) \cdot XII(n, x)\, dx = \begin{cases} 1, & \text{если} \quad n = m \\ 0, & \text{если} \quad n \neq m. \end{cases}$$

3. Элементы матрицы GII_N имеют вид $\lambda_i a_{ij}$, где a_{ij} есть $+1$ или -1, а $\lambda_i = \sqrt{2^{m(i)}}$.

4. Относительно рядов Фурье по системе $\{XII(m, x)\}_{m=0}^{\infty}$ справедливы теоремы п. 2.2.7 с последовательностью $m_n = p_0 p_1^n$.

3.3. Алгоритм быстрого преобразования по $\mathbf{GII}_N$ матрицам. Воспользовавшись блочной структурой $\mathbf{GII}_N$ матриц, можно построить быстрые алгоритмы преобразований по этим матрицам. Основная идея таких алгоритмов состоит в том, чтоб, используя рекуррентные формулы (3.1), разбить исходный вектор на более короткие вектора, преобразования которых, будучи скомбинированы определенным образом, дадут преобразование исходной последовательности. Алгоритм быстрого преобразования задается разложением матрицы $\mathbf{GII}_N$ в произведение слабозаполненных матриц. В работе [12] доказана

Теорема. Пусть $\mathbf{HA}_{0,2} = S_{p_0}\mathbf{HA}_{0,1}$, где $S_{p_0} = \mathbf{I}_{p_0/2} \otimes \begin{Vmatrix} 0 & 1 \\ -1 & 0 \end{Vmatrix}$ или $S_{p_0} = \mathbf{I}_{p_0/2} \otimes \begin{Vmatrix} 0 & -1 \\ 1 & 0 \end{Vmatrix}$ и $\mathbf{GII}_{n,i}$ $i=1, 2$ построены по (3.1), тогда $\mathbf{GII}_{n,1}$ представима в виде обычного произведения слабозаполненных матриц:

$$\mathbf{GII}_{n,1} = M_1 M_2 \dots M_{n+1} \tag{3.3}$$

$$M_r = \mathbf{I}_{p_1^{r-1}} \otimes [X_1 \otimes \mathbf{I}_{p_0 p_1^{n-2}} + X_2 \otimes S_{p_0 p_1^{n-2}}] \quad r = 1, 2, \dots, n$$

$$M_{n+1} = \mathbf{I}_{p_1^n} \otimes \mathbf{GII}_{0,1}. \tag{3.4}$$

Обозначим через $D(N)$ количество операций, необходимое на преобразование матрицы типа (3.1). На умножение сомножителя M_r, $r=1, 2, \dots, n$ на некоторый вектор потребуется Np_1 операций сложения или вычитания, поскольку в каждой его строке p_1 отличных от 0 чисел. Тогда

$$D(N) = \sum_{i=1}^{n} Np_1 + nD(p_0)$$

и в предложении, что в матрице $\mathbf{GII}_{0,1}$ все элементы ненулевые

$$D(N) = \sum_{i=1}^{n} Np_1 + Np_0 = N(np_1 + p_0).$$

Если $N = p_0^{\alpha_0} p_1^{\alpha_1} \dots p_n^{\alpha_m}$, то нетрудно получить, что

$$D(N) = N \sum_{i=0}^{m} p_i \alpha_i \leqq N \sum_{i=0}^{m} p_i \log_{p_i} N.$$

Используя разложение (3.3), (3.4) матриц (3.1), можно написать простые программы реализации этих преобразований.

В таблице 1 приведены для сравнения оценки числа операций, необходимых на выполнение введенных и известных преобразований. Оценки приводятся для двумерного случая, т. е. для преобразований над исходной матрицей f вида

$$F = \mathbf{A} f \mathbf{A}^T$$

где $\mathbf{A}$-матрица преобразования.

Таблица 1,

Тип преобразования	Литература	Порядок k исходной матрицы A_k	Порядок N матрицы преобразования	Число операций умножения, $D_1(N)$	Число операций сложения, $D_0(N)$
ХААРА	[6]	–	2^m	–	$2N(2N-2)$
УОЛША	[14]	–	2^m	–	$2N^2\log_2 N$
АДАМАРА	[11]	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	–	$2N^2\sum_{i=0}^{m}p_i\alpha_i$
slant	[7]	–	2^m	$2N(2N-4)$	$2N\left(N\log_2 N+\frac{N}{2}-2\right)$
ФУРЬЕ	[14–15]	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	$4N^2\sum_{i=0}^{m}p_i\alpha_i$	$4N^2\sum_{i=0}^{m}p_i\alpha_i$
Ортогональные (спеиальные)	[12]	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	–	$2N^2\sum_{i=0}^{m}p_i\alpha_i$
УОЛША—ХААРА	–	2^n	2^{mn}	–	$2N\cdot n\log_2 n\sum_{i=0}^{m}p_i\alpha_i$
АДАМАРА—ХААРА	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	$(p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m})^k$	–	$2Nk\frac{N-1}{k-1}\sum_{i=0}^{m}p_i\alpha_i$
slant—ХААРА	–	2^m	2^{mk}	$4N(k-2)\frac{N-1}{k-1}$	$2N\left(k\log_2 k+\frac{k}{2}-2\right)\frac{N-1}{k-1}$
ФУРЬЕ—ХААРА	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	$(p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m})^k$	$2Nk\frac{N-1}{k-1}\sum_{i=0}^{m}p_i\alpha_i$	$2Nk\frac{N-1}{k-1}\sum_{i=0}^{m}p_i\alpha_i$
ХААРА II	–	–	$p_0^{\alpha_0}p_1^{\alpha_1}\ldots p_m^{\alpha_m}$	–	$2N^2\left(\sum_{i=0}^{m}p_i\alpha_i+\frac{2(p_0^{\alpha_0}-1)}{p_0^{\alpha_0}}\right)$

Заключение. В настоящей работе обобщаются с двух различных позиций известные матрицы Хаара (преобразования Хаара): вводится

а) матрицы Хаара первого типа содержащие в себе, как частный случай матрицы Хаара,

б) матрицы Хаара второго типа, близкие по структуре к матрицам Адамара.

Метод построения матриц Хаара первого типа позволяет строить гибридные ортогональные системы: Уолша—Хаара, Адамара—Хаара, Фурье—Хаара.

Построенные гибридные системы содержат с одной стороны лучшие свойства ортогональных преобразований Фурье, Уолша, Адамара, slant и Хаара с другой: так, например, гибридные преобразования

— лучше декоррелируют данные чем преобразования Хаара;

— обладают быстрыми алгоритмами выполнения преобразования близким по скоростям к преобразованию Хаара;

— позволяют фрагментно более качественно (по заранее заданным фрагментам) обрабатывать изображения, чем преобразования Хаара;

— аппаратурно легко реализуемые и могут служить основой конструирования автоматических систем распознавания образов, в частности автоматических систем проверки качества плат.

В работе приведены программы реализации быстрых алгоритмов, а также оценки каличества операций выполнения указанных ортогональных преобразований.

Generalized Haar transforms and automatic quality test of printed circuit boards

S. S. Agaian and A. K. Matevosyan

In the given paper the following matrices are introduced:

a) Haar matrices of the first type, which contain partial case Haar matrices;

b) Haar matrices of the second type that by their structure are closer to Hadamard matrices.

The method of construction of Haar matrices of the first type makes it possible to construct the hybrid orthogonal systems of Walsh—Haar, Hadamard—Haar, Slant—Haar, Foury—Haar.

The constructed hybrid systems have the best properties of orthogonal transforms of Foury, Walsh, Hadamard on the one hand and of Haar ones on the other hand: So, for instance the hybrid transforms

— decode better Haar transforms;

— have rapid algorithms of transforms carryingouts which are close to Haar transform;

— makes it possible fragmently more qualitatively (for initially given fragments) to process pictures than Haar transforms;

— can be easier realized and can be used as a basis for construction of automatic pattern recognition systems particularly automatic systems of printed circuit board quality verification.

ВЫЧИСЛИТЕЛЬНЫЙ ЦЕНТР
АКАДЕМИИ НАУК АРМЯНСКОЙ ССР
УЛ. П. СЕВАКА, I
375 044 ЕРЕВАН—44, СССР

Литература

[1] Tou, J., R. Conzalez, *Pattern recognition principles,* Addison-Wesley Publishing Company, 1974.
[2] Wintz, P., Transform picture coting, *Proc. IEEE,* v. 60, N7, 1972.
[3] Hotelling, H., Analysis of a complex of statistical variables into principal components, *J. Educ. Psychology,* v. 24, 1933.
[4] Рабинер, Л., Б. Гоулд, Теория и применение цифровой обработки сигналов, *Мир,* 1978.
[5] Beauchamp, K. G., *Walsh functions and their applications,* Academic Press, London, 1975.
[6] Kremer, H., Algorithms for the Haar functions and the fast Haar transform, Symposium: Theory and Applications of Walsh functions, Hatfield Polytechnic, England, 1971.
[7] Pratt, W., W. H. Chen, L. Welch, Slant transform image coding, *IEEE Trans. Comm.,* v. COM—22, N8, August, 1974.
[8] Haar, A., Zur Theorie der orthogonalen Funktionensysteme, *Math. Ann.,* v. 69, 1910, pp. 331—371.
[9] Фино, В., Связь между преобразованиями Уолша—Адамара и Хаара, *ТИИЭР,* No 5, 1969.
[10] Луб, Дж. Л., Вероятные процессы, *И. Л.,* 1956.
[11] Агаян, С., А. Матевосян, Быстрое преобразование Адамара, *Труды ВЦ АН Арм. ССР,* No 12, 1980.
[12] Agaian, S. S., Algorithm of Orthogonal matrices fast transform, Proc. of the Fifth European Meeting on Cybernetics and Systems Research, v. 8, B. 1980.
[13] Wallis, D., Street, S. Wallis, *Combinatorics,* Room squares, sum-free set, Hadamard matrices, Lecture Notes in Math., 1972.
[14] Pratt, W. K., J. Kane, H. Andrews, *Proc. IEEE,* v. 57 and 58, 1969.
[15] Andrews, H., *Computer techniques in image processing,* Chap. 5, Academic Press, New York, 1970.

(Поступило 28-ого декабря 1980 г.)

A modelling tool based on mathematical logic T-PROLOG

By I. FUTÓ*, J. SZEREDI* and K. SZENES**

1. Introduction

T-PROLOG is a very high level language and a simulation language at the same time. It is a tool for simulation and modelling purposes equipped with the advantageous facilities of the very high level, logic based languages used in the field of AI. As we shall show, the marriage of two different principles — simulation and logic based problem solving results in a simulation technique new features. We amplified the logic based problem solving with a new — hitherto not considered — facility: the manipulation of time-dependent problems.

The structure of the simulation models written in T-PROLOG can be made dynamic — that is it can be suitably changed during run time — as the run time modification of the model description by means of addition and deletion of assertions/reductors to the program representing the model is allowed.

The language is planned to solve problems needing the cooperation of relatively few but frequently communicating processes. The processes pass and resume control dynamically, the control transfer is determined at run time, it is not determined at the time of writing the program (e.g. SIMULA [6]). The processes are controlled by sophisticated conditions. These are not simple and/or relations but the results of logical deductions of more than one steps.

The compiler of T-PROLOG is written in PROLOG and generates PROLOG programs that always contain a scheduler which modifies appropriately the strategy of the PROLOG interpreter.

2. Preliminaries

In this section an informal introduction to PROLOG is given. The reader familiar with this language may skip it.

2.1 Program elements. A PROLOG program consists of a set of Horn clauses. Horn clauses may be introduced as a follows: Every formula of a first order predicate calculi can be written as a conjunction of clauses

$$C_1 \wedge C_2 \wedge \ldots \wedge C_H .$$

A clause is the disjunction of literals

$$C_i = L_1 \vee L_2 \vee \ldots \vee L_k.$$

A literal is an atomic formula (positive literal) or a negation of an atomic formula (negative literal). An atomic formula is an expression of the form

$$P(t_1, \ldots, t_n),$$

where P is as n-ary predicate symbol and $t_1, \ldots, t_n$ are terms. A term is a variable symbol, a constant symbol, or an expression of the form

$$f(t_1, \ldots, t_r),$$

where f is an r-ary function symbol and $t_1, \ldots, t_r$ are terms. A clause having at most one positive literal is named Horn clause. The usual notation for Horn clauses is

$$B \leftarrow L_1 \wedge L_2 \wedge \ldots \wedge L_k,$$

where both sides of the arrow can be missing.

The various Horn clauses and their notation in T-PROLOG (j stands for the number of the positive literals and k stands for the number of the negative literals)

a) $j=0, k=0$	$\square$	the empty clause,
b) $j=1, k=0$	$B.$	the assertion,
c) $j=0, k>0$	$: A_1, A_2, \ldots, A_k.$	goalsequence,
d) $j=1, k>0$	$B: A_1, A_2, \ldots, A_k.$	rule of inference or reductor.

The assertion in T-PROLOG is expressed by a literal terminated by a point. The assertion means the declaration of a simple fact.

E.g. (1) Beautiful_is (Mary).

The reductor is expressed by a literal sequence in the following way, literal$_1$: literal$_2$, ..., literal$_n$. The reductor serves for the declaration of the "preconditioned" fact, literal$_2$ and ... and literal$_n$ are preconditions of literal$_1$.

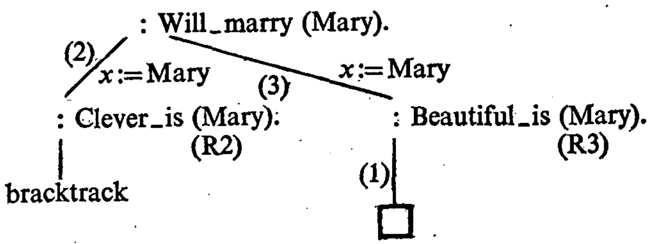
E.g. (2) Will_marry(x): Clever_is(x).
(3) Will_marry(x): Beautiful_is(x).

The meaning of (2) is, that every clever person will marry, while the meaning of (3) is the same, with "beautiful" instead of "clever". PROLOG goal statement is expressed by a literal or a sequence of literals preceeded by a: sign and terminated by a point.

E.g. (4): Will_marry(Mary).
(4)': Will_be_happy(Ann), Will_marry(Ann), Is_beautiful(Ann).

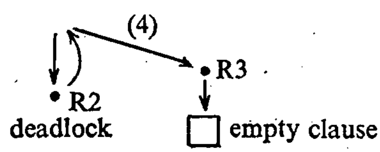
The goal statement expresses a simple question (4) or a complex one — (4)' — to be answered by the system. (Will marry Mary? or Will be happy Ann and Will marry Ann and Is beautiful Ann?).

To illustrate the execution of a PROLOG program we show the diagram of the above simple example. PROLOG uses the LUSH resolution for deriving the goal from the given set of assertions and reductors. The diagram of the program execution is the following:



2.2. The search tree of the deduction. The program execution of 2.1 can be represented by a tree, called search tree. The root of the search tree symbolizes (or is labeled by) the goal statement. When PROLOG tries to match the goal, clause (4), to the assertions/reductors it founds clauses (2) and (3) as being appropriate ones. Having executed the matching got (R2) and (R3), the remainder clauses of (2) and (3), respectively. These remainders we got leaving the first, matching literal from the clauses.

The nodes next to the root of the search tree symbolize (or are labeled by) the remainder of the clauses matching to the root. Proceeding in the proof the theorem prover founds no clauses matching to (R2) so it got into a deadlock. But for (R3) it founds the matching (1). The remainder clause of (R3) will be the next node of the search tree, in this case it is the empty clause. The nodes following those ones next to the root are labeled always by the respective remainder clauses. PROLOG traverses the search tree in a depth-first way. In the case of the example 2.1 this tree is very simple:



3. Running the processes of the simulation model

As one of the fundamental notions of simulation field is process, so we need to incorporate it in our language. Therefore our notion of process has to fulfil the following requirements:

a) The process notion has to cover the same concept people think of reading articles about simulation, co-routine programming etc.

b) The process notion has to originate somehow from the way of deduction used in T-PROLOG. This is necessary because the instructions supporting time considerations exploit the facilities resulting from the derivation method.

In section 2.2 we introduced the search tree of a PROLOG goal statement, that is the search tree of a T-PROLOG simple goal statement. Solving the problem given by a simple goal statement, the T-PROLOG system traverses the search tree determined by the simple goal and the given assertions/reductors. This traversing is terminated when the empty clause appears on one of the branches of the tree. We define the notion of the process as this search tree traversing process.

The simulation model is described by the T-PROLOG program. The model is put into motion when the system starts the processes. The running of the simulation model means the logical deduction process of proving the compound goal statement. This is just the process notion we need for the explanation of the meaning and the effect of the T-PROLOG instructions, and these explanations will fit into the "simulational" way of thinking and also into the logical deduction context. We shall say that the processes themselves traverse the search trees, — we do so because we shall have to explain the steps of the program execution which are actually jumps from one node of the tree to an appropriate other one. (In case of the compound goal this other node won't necessarily belong to the same tree as the starting node of the "jump" does.

In a real simulation model we have obviously more than one process. For the sake of the creation of these processes compound goals are used. To the compound goal of the simulation model description, which is the T-PROLOG program itself, we attach the set of processes corresponding the simple goals comprising this compound one. At the same time to the compound goal we attach the search space of the proof of the compound goal, that is the set of the search trees corresponding to the simple goals.

The run of the processes is synchronised by a built-in scheduler. The user can change in a certain extent the synchronisation strategy of the scheduler. (See at the description of T-PROLOG instructions.) Besides the synchronisation, on a one-processor computer the scheduler helps the system to run the processes quasi-simultaneously. The processes traverse the search space but every process keeps to its own search tree. When a process passes the control to an other one it also means the respective changing of search tree. This means that the proof "goes" breadth-way too, (while going top-down, depth-first on one tree), the order of the choice of the trees depends on the order of the processes in the lists of the scheduler.

The processes communicate in three ways:

a) through the common logical variables evaluated by pattern matching,

b) through the statements expressing sending and receiving messages (so called demon mechanism),

c) by the modification of the simulation model description (they can add/delete clauses to the program and they can even create new processes and delete existing ones).

4. Scheduling the activity of the processes

4.1. Simulation system. Simulation models cannot be described in a system lacking the facilities of handling the simulation system time (in short system time).

In T-PROLOG the preconditions prescribing to a process wether it may be proceeded to the next execution step or not, can be either dependent or independent on time. This means that to every step of a process a time interval can be given in order to determine the duration of the step. If the execution of the step "doesn't take time" then this duration is considered to be zero, otherwise it is given in an integer number of time unit.

Now the meaning of system time is the following: By definition its value is zero at the start of the execution of the T-PROLOG program. If — at the current moment — the value of system time is t_0 and a process having currently the control encounters a precondition referring to time duration of value t_d and the process is able to fulfil the precondition then it proceeds to its next instruction at system time t_0+t_d. (We have only two instructions expressing the fact that a statement is true depending on a time condition. The above explanation of system time however, could be made clearer thinking only of the explicit time condition. The meaning of all the operations on system time will be explained at their description).

4.2. The scheduler. A scheduler is needed not only because only one process can be run at a time and the state of the others has to be recorded but also for system time maintenance, for synchronisation of the — of course — communicating processes and as a tool of realisation of the desired effect of the simulation instructions.

The scheduler maintains two lists — the list of the waiting, and the list of the blocked processes. The elements of the first one are the processes able to run at the current time, and the processes having start time (it can be given in the goal description) or reactivation time (a process can meet a condition prescribing to suspend — hold — its activity for a given time interval) greater than the current system time. Processes of the waiting list are ordered on the bases of their activation or reactivation time. The elements of the second list are the processes which had become blocked because of some — as yet unfulfilled — precondition instructing the process, e.g. to wait for a message from an other one, or for the fulfilment of a condition.

The scheduler works in the following way:

a) Initialisation of the program execution.

The scheduler sets the system time to the minimum of the start times given in the description of the simple goals comprising the compound goal statement of the program. To every process denoted by the simple goal statements the start time — when the process has to begin its activity — and the end time — when the process has to finish its activity — can be given. (See sect. 6.)

b) Starting the program execution.

All of the processes enlist, in the order of their start time, to the waiting list.

c) Continuation of the program execution.

If the waiting list and the blocked list is empty, it means that the deduction was successful and the program terminates.

If there is a process in the waiting list with start time or reactivation time equal to system time then the scheduler starts the first one. If this gets blocked, or has to change its position in the waiting list (having encountered a precondition of suspending effect) then the scheduler starts the next process with appropriate start or reactivation time from the waiting list, if there is any such one at all.

If there is no such a process in the waiting list, or the waiting list has become exhausted, then the scheduler turns to investigate the blocked list. If the reason for staying there dissolves for any of the processes then the scheduler removes all such processes from the blocked list, puts them into the beginning of the waiting list and returns to the start of step 3. If there is no such a process in the blocked list then the scheduler goes to step 4.

d) Incrementing the system time.

If there is a process in the waiting list whose activation time is less than or equal to its end time (the point of time when the proof of the simple goal corresponding to this process has to be finished) then the scheduler sets system time to this activation time and returns to step 3. Otherwise backtracking begins. If the waiting list is empty but the blocked list isn't, then backtracking in time begins too. (In both cases the system goes back to the last previous decision point and choses the "next" alternative. If there is no next alternative even at the first decision point, then the problem cannot be solved under the conditions given in the program.) If both of the lists are empty then we are ready, all the processes have completed their proof.

5. A simple example

The example given here is a complete T-PROLOG program. The problem to be solved: Paul and Annie want to go to a movie at six o'clock. It is half past five now. Paul can go to a movie to see a film, if there are tickets available to a film at a movie, that film is acceptable for Annie and then travels to the movie. Annie can go to a movie to see the same film as Paul, if Paul can buy tickets to the film, the film is a good one and it is acceptable for her and then she travels to the movie. There are films which are good and Paul and Annie have to travel during an interval of time of nonzero length to get to the movie.

The corresponding T-PROLOG program is:

(1) Can_go_to(Paul, movie, film):
    There_is_ticket(movie, film),
    Wait_for(Acceptable_for(Annie, film)),
    Travels(Paul, movie).

(2) Can_go_to(Annie, movie, film):
    Wait(Paul_can_buy_a_ticket(film)),
    Good_film(film),
    Send(Acceptable_for(Annie, film)),
    Travels(Annie, movie).

(3) Paul_can_buy_a_ticket(film):
    Not(*Variable*(film)).

(1)—(3) are rules of interference or reductors.

(4) There_is_a_ticket(Rex, Hair).
(5) There_is_a_ticket(Athena, Star_wars).
(6) Good film (Hair).
(7) Good film(Star_wars).

(4)—(7) are time independent assertions.

(8) Travels(Paul, Rex): *During*(45).
(9) Travels(Paul, Athena): *During*(25).
(10) Travels(Annie, Rex): *During*(20).
(11) Travels(Annie, Athena): *During*(20).

(8)—(11) are time dependent assertions.

(12): *New*(Can_go_to(Paul, movie, film). Nil, Paul)*End* 30,
New(Can_go_to(Annie, movie, film). Nil, Annie)*End* 30.

(12) is a compound goal with goals to be achieved in 30 time units. (That is from 17,30 to 18,00 in our case).

The detailed explanation of the program execution is in Sec. 7.

6. T-PROLOG instructions

We shall not deal with time independent assertions and reductors, or built-in procedures, because they are used the same way as in PROLOG [1, 7].

6.1. Compound goal statement. The compound goal statement is a sequence of simple goals separated by "," and terminated by a ".".

6.2. Simple goal statement. In T-PROLOG, as it was said, there is a corresponding process to every goal. The syntax of the simple goal statement is the following:

:*New*(goalsequence, identifier) $\{Start\ T_1\}\{End\ T_2\}$,

where "goalsequence" is a sequence of literals separated by a ".", terminated by "Nil". "Identifier" serves as identifier of the goal and of the process corresponding to it. T_1, T_2 are points of time. The proof of the simple "goal" has to start at T_1 and has to be finished by T_2. Any or both of the two items enclosed in braces can be missing and in this case the default value of the start time and/or end time is assumed to be zero and 100 000 respectively. (4) and (4)′ of 2.1 can be given in T-PROLOG

:*New*(Will_marry(Mary). Nil,1). and
:*New*(Will_be_happy(Ann). Will_marry(Ann).
Is_beautiful(Ann). Nil,1).

6.3. Built-in procedures for synchronizing the events.

a) Time.

Hold(t)

If a process encounters this precondition in its search tree then stops the traversing for the time interval of length of t time units. The position of the process in the waiting list will be changed according to the value of the given time interval (see scheduler).

b) Logical condition.

Wait(condition), where "condition" is a literal.

If a process encounters this precondition in its search tree then if the "condition" can be proved as a simple PROLOG goal, then the proof takes place and the process continues its traversing. Otherwise, when the "condition" cannot be proved, then the process gets blocked and will be reactivated only when the "condition" is already true. The unprovability of "condition" may occur, because of a statement is missing or if a variable has not got the required value.

Wait(condition, identifier)

Where "condition" is a literal and "identifier" is a process name or a variable.

The user has the possibility to force the T-PROLOG scheduler to pass the control to the process named "identifier" if the "condition" cannot be proved, otherwise it has the same effect as *wait*( ). If "identifier" is a variable then the backtrack strategy is changed. In the previous cases if during backtrack the control reached this node again the backtracking is continued. In this case the next process from the waiting list gets the control, and during further backtracks in the worst case all the processes of the waiting list get the control. This built-in predicate was necessitated because considering certain unit clauses as resources seezed by the processes. The order of the seezing may influence the final solution of the problem.

6.4. Built-in procedures for explicite communication.

Wait_for(message), *Send*(message)

where "message" is a term.

If a process encounters precondition *Wait_for*( ) it will be blocked, and will be reactivated only when another process reaches a *Send*( ) precondition with matching "message" value. At the same time, sending process becomes suspended, the processes waiting for this message will be inserted before the first element of the waiting list and all processes inserted will be reactivated and continue execution till a blocking point, before the sending process, which preserved its relative position in the waiting list, would be active again.

6.5. Time dependent assertions.

a) For expressing the fact that a statement is true at a given time, till a given time, after a given time, before a given time, during a time interval, the suffixes *At*(*T*), *Till*(*T*), *After*(*T*), *Before*(*T*), *From*(T_1, *To*(T_2)) are used.

At the moment of the match of an assertion containing one of the above mentioned suffixes these conditions are evaluated, if their value is false, the T-PROLOG begins to backtrack.

b) For changing the position in the waiting list, according to an explicite time condition, the suffix *During*(T_d) is used.

If a process encounters an assertion suffixed with this suffix, then it passes the control to the next element of the waiting list and enlists to the waiting list to the position corresponding to system time $+T_d$.

6.6. Procedure for proving a sequence of conditions in a "really" parallel way.

Simultaneous(list of conditions)

If a process encounters this precondition — or rather — command, then it becomes blocked. The compiler assumes the list of conditions to be a "new compound goal" and generates a process to every "simple goal" of this compound goal. These processes start to execute in turn and continue to do so until all of their corresponding goals will have been proved one-by-one. However, from the point of view of the model these processes seem to execute parallely, because the reactivation time of the formerly blocked process is set to the completion time of these proofs which is the value of the system time at the blocking plus the maximum of time intervals necessary for the individual proofs.

6.7. Facilities for modifying the search strategy and the scheduler.

a) *Delete*(identifier)

If a process encounters this precondition then continues executing but the scheduler deletes the process with identifier "identifier" from its recording and the T-PROLOG compiler deletes the corresponding simple goal from the compound goal.

b) *Systemstate*($w1, b1, p$)

If a process encounters this precondition then the T-PROLOG compiler gives the waiting list of the scheduler in $w1$, the blocked list in $b1$, and the identifier of the running process in p.

7. Execution of the program of section 5

Initial state:

Waiting processes: 0:Paul:30. 0:Annie:30.Nil (start time:process id:end time)
Blocked processes: Nil
Active process: Paul
System time: 0

```
    (1) | :Can_go_to(Paul, movie, film).
[A]     | :There_is_ticket(movie, film),
        |      Wait_for(Acceptable_for(Annie, film)),
        |           Travels(Paul, movie).
    (4) |      movie:=Rex film:=Hair
        | :Wait_for(Acceptable_for(Annie, Hair)),
        |           Travels(Paul, Rex).
```

The process Paul is suspended waiting for a message.

Waiting processes: 0:Annie:30. Nil
Blocked processes: Paul. Nil
Active process: Annie
System time: 0

```
(2) | :Can_go_to(Annie, Rex, Hair).
    | :Wait (Paul_can_buy_a_ticket(Hair)),
    |      Good_film(Hair),
    |        Send(Acceptable_for(Annie, Hair)),
    |         Travels(Annie, Rex).
    :   (3) | :Paul_can_buy_a_ticket(Hair).
    :       |   film:=Hair
    :       □ :Not(Variable(Hair)).
(6) | :Good_film(Hair),
    |      Send(Acceptable_for(Annie, Hair)),
    |          Travels(Annie, Rex).
    | :Send(Acceptable_for(Annie, Hair)),
    |      Travels(Annie, Rex).
```

Process Paul waiting for the message is reactivated.

Waiting processes: 0:Paul:30. 0:Annie:30. Nil
Blocked processes: Nil
Active process: Paul
System time: 0

```
(8) | :Travels(Paul, Rex).
```

Backtrack

A backtrack occurs because of a time problem. Paul has to get to the movie at 30 (six o'clock) and now he needs for traveling 45 time units (8), thus actual system time $+45=0+45>30$, so Paul's goal can't be achieved this way.

There is another alternative but only at state [A].

```
    | :There_is_ticket(movie, film),
    |      Wait_for(Acceptable_for(Annie, film)),
(5) |          Travels(Paul, movie).
    |      movie:=Athena  film:=Star_wars
    | :Wait_for(Acceptable_for(Annie, Star_wars)),
    |            Travels(Paul, Athena).
```

Process Paul is suspended.

Waiting processes: 0:Annie:30. Nil
Blocked processes: Paul. Nil
Active process: Annie
System time: 0

```
(2) | :Can_go_to(Annie, Athena, Star_wars).
    | :Wait(Paul_can_buy_a_ticket(Star wars)),
    |     Good_film(Star_wars),
    |         Send(Acceptable_for(Annie, Star_wars)),
    |             Travels(Annie, Athena).
    :         (3) | : Paul_can_buy_a_ticket(Star_wars).
    :             |     film: = Star_wars
    :            □: Not(Variable(Star_wars)).
    | :Good_film(Star_wars),
    |         Send(Acceptable_for(Annie, Star_wars)),
    |             Travels(Annie, Athena).
    | :Send(Acceptable_for(Annie, Star_wars)),
    |         Travels(Annie, Athena).
```

Process Paul is reactivated.

Waiting processes: 0: Paul: 30. 0: Annie: 30. Nil
Blocked processes: Nil
Active process: Paul
System time: 0

(9) | : Travels(Paul, Athena).

Processing of Paul is abandoned for 25 time units. (He "travels".)

Waiting processes: 0: Annie: 30.25: Paul: 30. Nil
Blocked processes: Nil
Active process: Annie
System time: 0

(10) | : Travels(Annie, Athena).

Processing of Annie is abandoned for 20 time units. (She "travels".)

Waiting processes: 20: Annie: 30.25: Paul: 30. Nil
Blocked processes: Nil
Active process: Annie
System time: 20

|
□

The goal of process Annie is achieved in 20 time units (17^{50}), she arrived at the movie Athena to see the film Star_wars.

Waiting processes: 25: Paul: 30. Nil
Blocked processes: Nil
Active process: Paul
System time: 25

|
□

The goal of process Paul is achieved in 25 time units (17^{55}), he arrived at the movie Athena to see the film Star_wars.

Waiting processes: Nil
Blocked processes: Nil
Active process: –
System time: 25

The problem is solved.

8. Conclusion

As it was shown in the preceeding sections we provide simulation technique more advanced than the previous methods from the following aspects:

a) the system takes over a part of the problem solving effort from the user who has to concentrate rather on defining the task then on solving it;

b) the system changes automatically and dynamically the model on the basis of logical consequences derived from sophisticated preconditions;

c) a built-in backtrack mechanism permits *backtracking in time* in case of a deadlock or hopeless intermediate situation (logical condition necessary for the continuation of the execution is missing or the current time conditions have become contradictory);

d) T-PROLOG provides for a process communication mechanism that is sophisticated enough owing the fact that the processes communicate through variables evaluated by pattern matching or by modification of the model description (the traditional way of communication is preserved, too: the processes are able to send /receive messages).

All the advantages enumerated above are due to this hitherto unusual approach: building the simulation method on a language founded on the principles of mathematical logic. As far as the traditional simulation facilities of T-PROLOG are concerned:

a) a dynamic simulation approach has been choosen: the processes pass the control to each other dynamically, according to the requirements of the situation and not on the basis of a rigid, preprogrammed resume — detach technique as e.g. SIMULA 67 [6],

b) transaction and event oriented simulation approach are allowed.

Abstract

A logic based simulation language is presented. The language is an extension of PROLOG towards time and process manipulation.

Comparing it to the traditional simulation languages the language has the following advantages due to its logical basis:

— it changes automatically and dynamically the simulation model on the basis of logical consequences derived from sophisticated preconditions,
— the system takes over part of the problem solving effort from the user,
— a built in backtrack mechanism permits backtracking in time in case of a deadlock or the occurence of a hopeless intermediate situation during program execution,
— a more advanced process communication mechanism is presented for the user.

The processes are synchronized by a built-in scheduler. Its strategy can be modified by the user. The realisation of the scheduler, the effect of the simulation instructions and the way of using the logical deduction ensures that:
— the interaction of the processes is dynamic, the processes pass and resume control dynamically, and
— either the event-, or the transaction oriented way of simulation programming is allowed.

* INSTITUTE FOR COORDINATION
OF COMPUTER TECHNIQUES
AKADÉMIA U. 17.
BUDAPEST, HUNGARY
H—1052

** CHEMICAL WORKS
OF RICHTER GEDEON
BUDAPEST, HUANGRY
H—1475

Bibliography

[1] FUTO, I., F. DARVAS, P. SZEREDI, Application of PROLOG to implement QA and DBM systems, *Logic and data bases,* Ed. H. Gallaire, J. Minker, Plenum Press New York, 1978, pp. 347—376.
[2] FUTO, I., J. SZEREDI, PAPLAN a very high level programming language for parallel problem solving, Preprints of Int. Conf. on Robot Control, Bratislava, 30.06—04.07, 1980, pp. 74—76.
[3] GPSS/360 User's Manual, IBM 420—03261.
[4] KIVIAT, P. J., *Simulation Using SIMSCRIPT II.,* Rand Corporation, 1968.
[5] KOWALSKI, R., Predicate logic as a programming language, DCL Memo no. 70, Edinburgh University, Edinburg, 1974.
[6] SIMULA Reference Manual, Control Data 6400/6500/6600 Computer Systems.
[7] PEREIRA, F., L. PEREIRA, D. WARREN, User's guide to DECsystem—10 PROLOG, Dept. of AI Univ. of Edinburgh, 1978.

(Received Dec. 1, 1980)

On fault tolerant L-processors

By S. R. Gross

1. Introduction and motivation

In sequential computers the number of real active elements is very small in contrast to the total number of elements. On the other hand the working speed of these elements has nearly reached the physical limitations (with respect to the today's technologies). Therefore a speed up is no more reasonable by improving the conventional (von Neumann type) architectures and a natural approach to increase the computing power seems to be the principle of parallel processing in computer architecture. Cellular spaces might be considered as models for such parallel computing devices. The implementation of cellular structures renders possible by the advent of LSI-technology, too.

Since cellular spaces consist of many cells which are not necessarily totally reliable, it is important to develop strategies for error correction and error detection. Up to now several articles concentrate on this topic, see e.g. [HANO 75], [NIKO 75], and [WRIG 76]. Here we want to investigate this topic in L-processors, which differ from a cellular space as follows:

- a state is a 2-tupel which consists of a visible (external) component and an invisible (internal) component, a so-called qualifier (i.e. the basic cell is a Mealy-type automaton),
- there are several local maps,
- the transition-functions are centralized in a special control unit and 'shared' by the single cells.

Thus, an L-processor may be considered to be an emulator of a cellular space. For details see Legendi [LEGE 76].

The techniques of error correction in L-processors are similar to those of Nishio and Kobuchi [NIKO 75] where the basic idea is that the work of each original cell is simulated by three cells. In this case we are able to correct single errors prior to each state transition by a majority decision, if we assume that the majority decision element is faultless.

2. Notations and basic definitions

Let N denote the natural numbers, $\mathrm{N}_0=\mathrm{N}\cup\{0\}$ and $\mathbf{Z}$ the set of integers. For $d\in\mathrm{N}$ the d-fold Cartesian product of $\mathbf{Z}$ is denoted by $\mathbf{Z}^d$, i.e. $\mathbf{x}\in\mathbf{Z}^d$ can be written as $\mathbf{x}=(x_1, x_2, \ldots, x_d)$ with $x_i\in\mathbf{Z}$ $(i=1, \ldots, d)$. Especially $\mathbf{0}=(0, 0, \ldots, 0)$. $P_i^{(d)}(x_1, \ldots, x_d):=x_i$ $(i=1, \ldots, d)$ denotes the i-th projection. If X is a set, the cardinality of X is denoted by $|X|$ and the set of subsets of X by $\mathfrak{P}(X)$. $\emptyset$ denotes the empty set.

For $\mathbf{x}, \mathbf{y} \in \mathbf{Z}^d$ we define the operations $\mathbf{x} \pm \mathbf{y} := (x_1 \pm y_1, x_2 \pm y_2, \ldots, x_d \pm y_d)$. Let $X, Y \subseteq \mathbf{Z}^d$, $X \neq \emptyset$, $Y \neq \emptyset$. Then we define the sum and difference by $X \pm Y := \{\mathbf{x} \pm \mathbf{y} / \mathbf{x} \in X \wedge \mathbf{y} \in Y\}$. In the following we write $\mathbf{x} \pm Y$ instead of $\{\mathbf{x}\} \pm Y$. Let $\mathbf{x} \in \mathbf{Z}^d$ and $T = (t_{ij})_{i,j=1,\ldots,d}$ a matrix with $t_{ij} \in \mathbf{Z}$ $(i, j = 1, \ldots, d)$. Then we define the product

$$\mathbf{x} \cdot T := \left(\sum_{i=1}^{d} x_i \cdot t_{i1}, \ldots, \sum_{i=1}^{d} x_i \cdot t_{id} \right)$$

and for $X \subseteq \mathbf{Z}^d$, $X \neq \emptyset$

$$X \cdot T := \{\mathbf{x} \cdot T / \mathbf{x} \in X\}.$$

Definition 2.1. Let $d, n \in \mathbf{N}$. An n-tupel $N = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n)$, $\mathbf{a}_i \in \mathbf{Z}^d$ $(i = 1, \ldots, n)$, $\mathbf{a}_i \neq \mathbf{a}_j$ for $i \neq j$ is called a d-dimensional neighbourhood index.

The (unordered) set $\tilde{N} = \{P_i^{(n)}(N) / i = 1, \ldots, n\}$ is called a neighbourhood template. In general we claim that there exists an i $(1 \leq i \leq n)$ such that $\mathbf{a}_i = \mathbf{0}$. In the following we use 'neighbourhood index' and 'neighbourhood template' synonymously.

Definition 2.2. Let $k \in \mathbf{N}_0$, $d \in \mathbf{N}$. The von Neumann neighbourhood template is defined by $H_k^{(d)} := \{\mathbf{x} / \mathbf{x} \in \mathbf{Z}^d \wedge |\mathbf{x}| \leq k\}$, where $|\mathbf{x}| = \sum_{i=1}^{d} |x_i|$.

We write H_k instead of $H_k^{(d)}$ if there is no doubt of the dimension. $H_1 = \{(0, 0), (-1, 0), (0, 1), (1, 0), (0, -1)\}$ is the most frequently used von Neumann neighbourhood template.

Definition 2.3. Let A, Q be finite nonempty sets and $\tilde{N} = \{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ a d-dimensional neighbourhood template. $f\colon A^n \times Q \to A \times Q$ is called a local transition function.

Definition 2.4. $\mathfrak{A} = (A, Q, d, N, \{f^{(1)}, \ldots, f^{(l)}\})$ is called an L-processor, if

1) A is a finite nonempty set of (external) states,
2) Q is a finite nonempty set of (internal) states, in the following called qualifiers,
3) d is the dimension of the space,
4) $N = (\mathbf{a}_1, \ldots, \mathbf{a}_n)$ is a d-dimensional neighbourhood index,
5) $f^{(j)}\colon A^n \times Q \to A \times Q$ $(j = 1, \ldots, l)$ are local transition functions,
6) There exists a $(z_0, q_0) \in A \times Q$, called the quiescent state of the L-processor, with $f^{(j)}(z_0, \ldots, z_0, q_0) := (z_0, q_0)$ for $j = 1, \ldots, l$.

Definition 2.5. Let $\mathfrak{A}$ be an L-processor. A function $c\colon \mathbf{Z}^d \to A$ is called an external configuration or simply configuration. A function $q\colon \mathbf{Z}^d \to Q$ is called an internal configuration. The set of total (internal and external) configurations is denoted by C.

Definition 2.6. $F\colon C \to C$ is called a global transition function (induced by f), if for all $\mathbf{x} \in \mathbf{Z}^d$ and for all total configurations $(c, q) \in C$ the following holds: $Fc((\mathbf{x}), q(\mathbf{x})) = f(c(\mathbf{x} + \mathbf{a}_1), \ldots, c(\mathbf{x} + \mathbf{a}_n), q(\mathbf{x}))$.

Definition 2.7. The support of a total configuration is defined by $\sup(c, q) = \{\mathbf{x}/\mathbf{x}\in \mathbf{Z}^d \wedge (c(\mathbf{x}), q(\mathbf{x})) \neq (z_0, q_0)\}$ with (z_0, q_0) the quiescent state. The set of all total configurations with finite support is denoted by $\bar{C}$.

If we use L-processors for computations we want a reliable system although the individual cell may be unreliable. A cell is said to misoperate if its next external state differs from the expected one (by application of the local transition function to the states of its neighbours). The reason for such a misoperation may be the permanent breakdown of a cell, an occasional failure caused by noise or something else. Therefore an L-processor is called a real L-processor if some cells may misoperate.

In order to make analysis practicable, we restrict the occurrence of misoperations in the following way:

Definition 2.8. Let K be a finite connected subset of $\mathbf{Z}^d$ containing the origin $\mathbf{0}$. A real L-processor is said to misoperate with K-separated errors if at most one cell of each area $\mathbf{x}+K$, $\mathbf{x}\in\mathbf{Z}^d$, misoperates at each state transition.

'Connected' means connected with respect to the underlying neighbourhood template $\tilde{N}$. In the following we denote this as K-separated error condition (K-se condition).

Definition 2.9. Let $\mathfrak{A}_1 = (A_1\times Q_1, 2, H_1, \{f_1^{(1)}, \dots, f_1^{(k)}\})$ be an L-processor, $\mathfrak{A}_2 = (A_2\times Q_2, 2, N_2, \{f_2^{(1)}, \dots, f_2^{(l)}\})$ a real L-processor and $k_1, k_2\in\mathbf{N}$. We say $\mathfrak{A}_2$ simulates $\mathfrak{A}_1$ in k_2/k_1-time if and only if there exist functions G and H such that the diagram in Fig. 1 is commutative. The index K indicates that the K-se condition holds.

$$\begin{array}{ccc} \bar{C}_1 & \xrightarrow{F_1^{k_1}} & \bar{C}_1 \\ {\scriptstyle G}\downarrow & & \downarrow{\scriptstyle H} \\ C_2 & \xrightarrow{F_K^{k_2}} & \mathfrak{P}(C_2)\setminus\emptyset \end{array}$$

Fig. 1
Simulation of L-processors

The 'realistic' global transition function F_K maps C_2 into the set $\mathfrak{P}(C_2)\setminus\emptyset$ of subsets of C_2, because a configuration may have more successors in consequence of the real behavior.

3. Error correction in realtime with an 21-element neighbourhood template

Now we want to design a real L-processor $\mathfrak{A}_2 = (A_2\times Q_2, 2, N_2, \{f_2^{(1)}, \dots, f_2^{(l)}\})$ which simulates a given L-processor $\mathfrak{A}_1 = (A_1\times Q_1, 2, H_1, \{f_1^{(1)}, \dots, f_1^{(k)}\})$ in realtime, i.e. $k_1 = k_2$ (see Definition 2.9).

The most elementary trick of error correction might be to implement each cell three times and to take the majority result. Therefore we must spread out the

2-dimensional space of integers by a transformation matrix $T=\begin{pmatrix}3 & 0\\ 0 & 1\end{pmatrix}$. A coding unit $M=\{(0,0),(1,0),(2,0)\}$ fills up the gaps. It is easy to show that T and M guarantee a unique and total cover of $\mathbf{Z}^2$. G maps a cell and its neighbours into the following region, see Fig. 2.

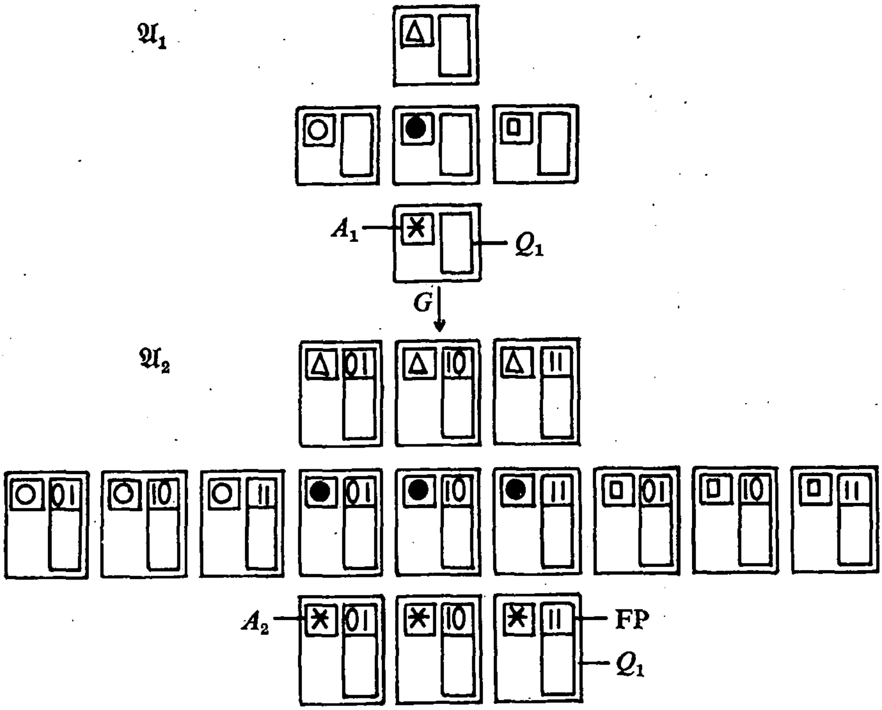


Fig. 2
Function G

Each cell $\mathbf{x}\in\mathbf{Z}^2$ of $\mathfrak{A}_1$ is mapped into the cell $\mathbf{x}\cdot T$ by the transformation matrix T. Now we add the coding unit to it. It is easy to see that we get for each cell in $\mathfrak{A}_1$ exactly three cells in $\mathfrak{A}_2$ in this way. In the following we denote $\mathbf{x}\cdot T+M$ as a block. Each cell of $\mathbf{x}\cdot T+M$ contains a copy of $\mathbf{x}$.

Since we want to carry out the error correction with the majority function, each cell in $\mathfrak{A}_2$ has to know to which block it belongs. This information is stored in the qualifier of the cell by G. The set of states and the qualifier of $\mathfrak{A}_2$ are defined by $A_2=A_1$ and $Q_2=Q_1\times\mathrm{FP}$, where $\mathrm{FP}=\{1,2,3\}$ is a so-called fingerprint. The qualifier of a cell $\mathbf{x}\in\mathbf{Z}^2$ in $\mathfrak{A}_2$ is delivered by $\tilde{q}(\mathbf{x})=(q_1(\mathbf{x}),\mathrm{fp})$.

Now we can define the functions G and H.

1) For any $(c,q)\in\bar{C}_1$ and for any $\mathbf{x}\in\mathbf{Z}^2$

$$G(c(\mathbf{y}),q(\mathbf{y}))=(c(\mathbf{x}),(q(\mathbf{x}),\mathrm{fp})),$$

where $\mathbf{y}\in(\mathbf{x}\cdot T+M)$ and $\mathrm{fp}=1+(y_1 \text{ modulo } 3)$.

2) Function H is locally defined by the triple (M, T, m), where m denotes the majority function. If for any total configuration $(c, \tilde{q})$ of $\mathfrak{A}_2$ there exists a total configuration (c', q') of $\mathfrak{A}_1$ with $c'(\mathbf{x})=m(c(\mathbf{x}\cdot T+M))$ and $q'(\mathbf{x})=m(q_1(\mathbf{x}\cdot T+M))$ for all $\mathbf{x}\in\mathbf{Z}^2$ then $H(c'(\mathbf{y}), q'(\mathbf{y}))=(c'(\mathbf{x}), (q'(\mathbf{x}), \mathrm{fp}))$, where $\mathbf{y}\in(\mathbf{x}\cdot T+M)$ and $\mathrm{fp}=1+(y_1 \text{ modulo } 3)$.

Next we must define the neighbourhood template N_2 of $\mathfrak{A}_2$. Since each cell of $\mathfrak{A}_2$ must have sufficient state information within its neighbourhood to calculate its next state correctly we choose N_2 as the 21-element template sketched in Fig. 3. The hatched cell is the origin of the neighbourhood template.

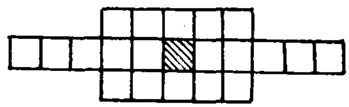


Fig. 3
Neighbourhood template $N_2=(H_1\cdot T+M)-M$

It is easy to see that every cell of the central block of $\mathfrak{A}_2$ in Fig. 2 disposes of the whole information of the 'extended von Neumann neighbourhood template' $H_1\cdot T+M$. We choose the set $K=N_2$, because it must be guaranted that the majority function has at least two correct values. The real L-processor $\mathfrak{A}_2$ may be constructed in such a manner that it has exactly the same number of local transition functions as the L-processor $\mathfrak{A}_1$, i.e. $l:=k$. For any $\mathbf{x}\in\mathbf{Z}^2$ the local transition functions $f_2^{(j)}$ $(j=1, \dots, k)$ are defined by:

$$\big(c'(\mathbf{x}), (q_1'(\mathbf{x}), \mathrm{fp})\big) = f_2^{(j)}\big(c(\mathbf{x}+N_2), (q_1(\mathbf{x}), \mathrm{fp})\big),$$

where $c'(\mathbf{x})$ and $q_1'(\mathbf{x})$ are defined by:

$$\begin{aligned}\big(c'(\mathbf{x}), q_1'(\mathbf{x})\big) = f_1^{(j)}\big(&m(c(\mathbf{x}-(\mathrm{fp}-1, 0)), c(\mathbf{x}-(\mathrm{fp}-2, 0)), c(\mathbf{x}-(\mathrm{fp}-3, 0))),\\ &m(c(\mathbf{x}-(\mathrm{fp}, 0)), c(\mathbf{x}-(\mathrm{fp}+1, 0)), c(\mathbf{x}-(\mathrm{fp}+2,0))),\\ &m(c(\mathbf{x}-(\mathrm{fp}-1, -1)), c(\mathbf{x}-(\mathrm{fp}-2, -1)), c(\mathbf{x}-(\mathrm{fp}-3, -1))),\\ &m(c(\mathbf{x}-(\mathrm{fp}-4, 0)), c(\mathbf{x}-(\mathrm{fp}-5, 0)), c(\mathbf{x}-(\mathrm{fp}-6, 0))),\\ &m(c(\mathbf{x}-(\mathrm{fp}-1, 1)), c(\mathbf{x}-(\mathrm{fp}-2, 1)), c(\mathbf{x}-(\mathrm{fp}-3, 1))), q_1(\mathbf{x})\big).\end{aligned}$$

It is possible to change from $f_2^{(j)}$ to $f_1^{(j)}$, because the template $\mathbf{x}+N_2=\mathbf{x}+K$ contains enough information. The above constructed real L-processor $\mathfrak{A}_2$ simulates the L-processor $\mathfrak{A}_1$ in realtime.

Theorem 1. If we choose $\mathfrak{A}_2$ and K as defined above, then $\mathfrak{A}_2$ corrects any K-separated error.

Proof. The proof is delivered by considering all possibilities of single errors.

1) The state of a cell is incorrect.

In consequence of the K-se condition the states of all other cells of the extended von Neumann neighbourhood template (see Fig. 2) are correct, i.e. the majority function is able to correct the error.

2) The qualifier of a cell is incorrect.

Now the cell works with an incorrect state and an incorrect local transition function in the next step in general. Subsequently the cell always announces an incorrect state in general, i.e. we may consider the cell as permanently broken down. In consequence of the K-se condition all other cells of the extended von Neumann neighbourhood template must work correctly so that they announce correct states in the next step, i.e. the error may be corrected (see 1).

3) The majority function of a cell works incorrectly.

Since the local transition function $f_1^{(j)}$ works with incorrect arguments, in general the cell announces an incorrect state and works with incorrect local transition functions in future. In consequence of the K-se condition all other cells of the extended von Neumann neighbourhood template must work correctly so that 1) or 2) hold in the next step. □

4. Error correction in 5-slow with von Neumann neighbourhood template

Now we present a real L-processor $\mathfrak{A}_2=(A_2\times Q_2, 2, H_1, \{f_2^{(1)}, \dots, f_2^{(l)}\})$ which simulates the L-processor $\mathfrak{A}_1=(A_1\times Q_1, 2, H_1, \{f_1^{(1)}, \dots, f_1^{(k)}\})$ in 5-slow, i.e. $k_2=5\cdot k_1$ (see Definition 2.9). We choose the transformation matrix T and the coding unit M as above. In consequence of the von Neumann neighbourhood template the L-processor $\mathfrak{A}_2$ works in two different steps. First the information of the extended von Neumann neighbourhood template must be compressed into the H_1-neighbourhood template of each cell of $\mathfrak{A}_2$ and second the cells have to change their state. Since every cell of the central block of $\mathfrak{A}_2$ in Fig. 2 must have access to the whole information of the extended von Neumann neighbourhood template to change its state, it is necessary to compress four times. Therefore the set of states of $\mathfrak{A}_2$ is given by $A_2=A_1^9$ and the set of qualifiers by $Q_2=Q_1\times \mathrm{FP}\times \mathrm{MZ}$ where $\mathrm{FP}=\{1, 2, 3\}$ is a fingerprint as above and MZ a modulo-5-counter. Fig. 4 shows the state- and qualifier-register of a cell.

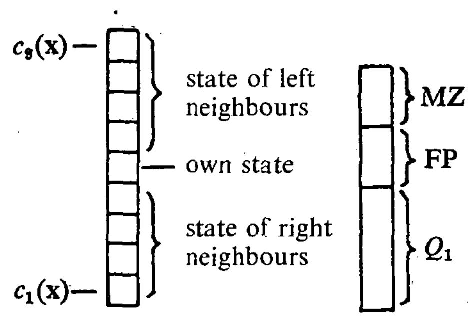


Fig. 4
State- and qualifier-register of a cell

The state of a cell $\mathbf{x}\in \mathbf{Z}^2$ in $\mathfrak{A}_2$ is delivered by $\tilde{c}(\mathbf{x})=(c_9(\mathbf{x}), \dots, c_1(\mathbf{x}))$ and the qualifier by $\tilde{q}(\mathbf{x})=(q_1(\mathbf{x}), \mathrm{fp}, \mathrm{mz})$.

Now we can define the functions G and H.

1) For any $(c, q)\in \bar{C}_1$ and for any $\mathbf{x}\in \mathbf{Z}^2$

$$G(c(\mathbf{y}), q(\mathbf{y})) = ((-, -, -, -, c(\mathbf{x}), -, -, -, -), (q(\mathbf{x}), \mathrm{fp}, \mathrm{mz})),$$

where $y \in (x \cdot T+M)$, fp$=1+(y_1$ modulo 3) and '$-$' denotes any state. Besides mz will be chosen such that it specifies that no information has arrived.

2) Again function H is locally defined by the triple (M, T, m). If for any total configuration $(\tilde{c}, \tilde{q})$ of $\mathfrak{A}_2$ there exists a total configuration (c', q') of $\mathfrak{A}_1$ with $c'(x) = m(c_5(x \cdot T+M))$ and $q'(x) = m(q_1(x \cdot T+M))$ for all $x \in \mathbf{Z}^2$ then

$$H(c'(y), q'(y)) = ((-, -, -, -, c'(x), -, -, -, -), (q'(x), \text{fp}, \text{mz})),$$

where $y \in (x \cdot T+M)$, fp$=1+(y_1$ modulo 3), '$-$' denotes an arbitrary state and mz will be chosen such that it specifies that no information has arrived.

Before a cell $x \in \mathbf{Z}^2$ in $\mathfrak{A}_2$ can change its state, the information of the extended von Neumann neighbourhood template must be compressed. Fig. 5 demonstrates the cells from which x collects its information during the compression.

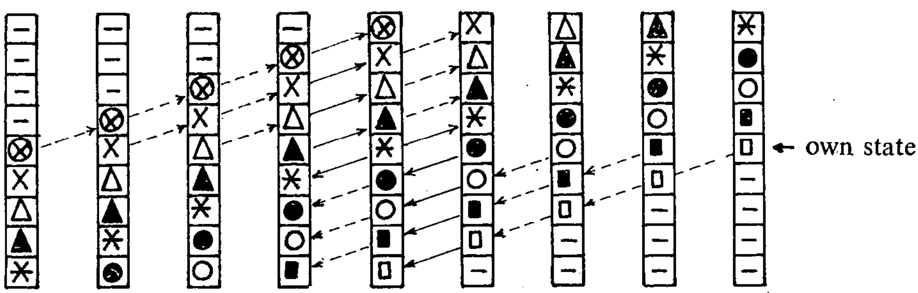


Fig. 5
Compression of information. '$-$' denotes an arbitrary state

Flow of compression. Each cell sends its own state to its left and right neighbours, the states of its left neighbours only to its right one and the states of its right neighbours only to its left one (see cell x) as shown by the arrows in Fig. 5. At the beginning each cell disposes only of its own state. After one cycle a cell disposes of its own state, the state of its left neighbour and the state of its right neighbor in its state register. The remaining six components of the state register contain arbitrary states. After four cycles each cell disposes in its state register of its own state, the states of its four left neighbours and its four right neighbours in this way.

This is the time where each cell in a block of the extended von Neumann neighbourhood template is provided with sufficient information to compute its next state. The compression is realized by hardware or by the following function.

Let $x, y, z \in \mathbf{Z}^2$, $y = x-(1, 0)$ and $z = x+(1, 0)$. Then a function h: $A_2^5 \times Q_2' \to A_2 \times Q_2'$ is defined by

$$h(c(x+H_1), (q, \text{fp}, \text{mz})) = \begin{cases} ((c_8(y), \ldots, c_5(y), c_5(x), c_5(z), \ldots, c_2(z)), \\ (q, \text{fp}, \text{mz} \oplus 1)), \quad \text{if} \quad 0 < \text{mz} \oplus 1 < 4 \\ ((c_8(y), \ldots, c_5(y), c_5(x), c_5(z), \ldots, c_2(z)), \\ (q', \text{fp}, \text{mz} \oplus 1)), \quad \text{if} \quad \text{mz} \oplus 1 = 4 \end{cases}$$

$\oplus$ denotes the add-operation modulo 5.

$Q_2' = (Q_1 \times \{0, 1\}) \times \mathrm{FP} \times \mathrm{MZ}$, i.e. the qualifier of $\mathfrak{A}_1$ is extended by one bit indicating whether $\mathfrak{A}_2$ is in a phase of compression or transformation, i.e. q' differs from q only in this bit. If $\mathrm{mz} \oplus 1 = 0$ holds, $\mathfrak{A}_2$ is in a phase of transformation. If we use the function h for compression the functions G and H must be modified adequate to Q_2'. In this case we define $f_2^{(k+1)} := h$, i.e. $l := k+1$. In the following we consider only cells of the central block of the extended von Neumann neighbourhood template and their direct neighbours to define the local transition functions $f_2^{(j)}$ $(j=1, \dots, k)$. After compression these cells contain the following relevant states, see Fig. 6.

Now we can define the local transition functions $f_2^{(j)}$ $(j=1, \dots, k)$. Let $\mathbf{x} \in \mathbf{Z}^2$

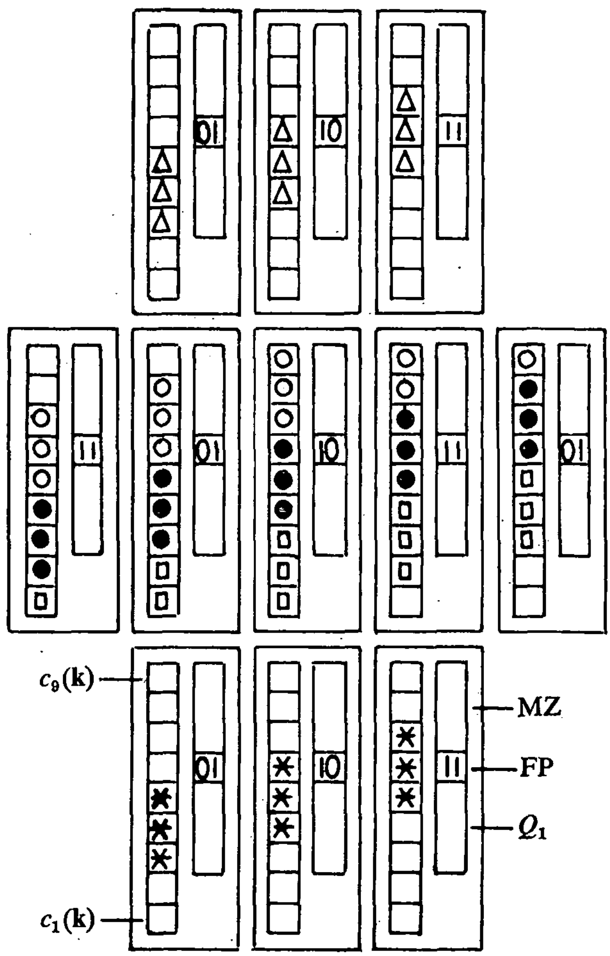


Fig. 6

Relevant states for the central block of the extended von Neumann neighbourhood template after compression and $\mathbf{a}_1=(0,0)$, $\mathbf{a}_2=(-1,0)$, $\mathbf{a}_3=(0,1)$, $\mathbf{a}_4=(1,0)$, $\mathbf{a}_5=(0,-1)$ the five elements of the von Neumann neighbourhood template.

$$\begin{aligned}(c'(\mathbf{x}), q'(\mathbf{x})) &= f_2^{(j)}(c(\mathbf{x}+H_1), (q_1(\mathbf{x}), \mathrm{fp}, \mathrm{mz})) \quad \text{or} \\ &\quad f_2^{(j)}(c(\mathbf{x}+H_1), (\{q_1(\mathbf{x})\}\times\{0,1\}, \mathrm{fp}, \mathrm{mz})) \\ &= (-,-,-,-, c_5'(\mathbf{x}), -,-,-,-), (q_1'(\mathbf{x}), \mathrm{fp}, \mathrm{mz}\oplus 1)) \quad \text{or} \\ &\quad (-,-,-,-, c_5'(\mathbf{x}), -,-,-,-), (\{q_1'(\mathbf{x})\}\times\overline{\{0,1\}}, \mathrm{fp}, \mathrm{mz}\oplus 1)),\end{aligned}$$

where '$-$' denotes any state and '$\oplus$' the add-operation modulo 5 again. $\overline{\{0,1\}}$ means that $f_2^{(j)}$ forms the complement of $\{0,1\}$, if the compression is realized by software. $c_5'(\mathbf{x})$ and $q_1'(\mathbf{x})$ are defined by

$$\begin{aligned}(c_5'(\mathbf{x}), q_1'(\mathbf{x})) = f_1^{(j)}(&m(c_{\mathrm{fp}+2}(\mathbf{x}+\mathbf{a}_1), c_{\mathrm{fp}+3}(\mathbf{x}+\mathbf{a}_1), c_{\mathrm{fp}+4}(\mathbf{x}+\mathbf{a}_1)), \\ &m(c_{\mathrm{fp}+4}(\mathbf{x}+\mathbf{a}_2), c_{\mathrm{fp}+5}(\mathbf{x}+\mathbf{a}_2), c_{\mathrm{fp}+6}(\mathbf{x}+\mathbf{a}_2)), \\ &m(c_{\mathrm{fp}+2}(\mathbf{x}+\mathbf{a}_3), c_{\mathrm{fp}+3}(\mathbf{x}+\mathbf{a}_3), c_{\mathrm{fp}+4}(\mathbf{x}+\mathbf{a}_3)), \\ &m(c_{\mathrm{fp}}(\mathbf{x}+\mathbf{a}_4), \quad c_{\mathrm{fp}+1}(\mathbf{x}+\mathbf{a}_4), c_{\mathrm{fp}+2}(\mathbf{x}+\mathbf{a}_4)), \\ &m(c_{\mathrm{fp}+2}(\mathbf{x}+\mathbf{a}_5), c_{\mathrm{fp}+3}(\mathbf{x}+\mathbf{a}_5), c_{\mathrm{fp}+4}(\mathbf{x}+\mathbf{a}_5)), q_1(\mathbf{x})).\end{aligned}$$

The definition is similar to that in section 3. First we apply the majority function to each component block. These results and the first part of the qualifier of the cell are the values for the local transition function $f_1^{(j)}$, which delivers the new state of component c_5 and the new first part of the qualifier. Again it is possible to change from $f_2^{(j)}$ to $f_1^{(j)}$, because the template $\mathbf{x}+H_1$ contains enough information.

Up to now we did not say anything about the K-se condition which must be modified, if the above real L-processor should correct errors. The modification is necessary because additionally to the error correction there are two further problems.

1) We have a phase of compression.
2) An error correction is not possible concerning the qualifier of a cell.

In the first solution we have permitted that single cells may break down totally. Now such cells will interrupt the information-flow during the compression. Because of that the four right and left neighbours of these cells would hold up to four successive incorrect states in their components, i.e. an error correction is not possible.

Fig. 7 shows the information-flow through one cell during compression. '$\lfloor\ \ \rfloor$' denotes a block. The local transition functions of cells marked by 'X' deliver an incorrect state, because they work with at least one faulty argument. 1) shows the faulty cells if the left cell of a block is broken down, 2) shows these cells if the central cell of a block is broken down, and 3) shows these cells if the right cell of a block is broken down.

Therefore the first modification of the K-se condition consists in that only single components of a cell may be incorrect. It is easy to see that the errors must have a distance of three, i.e. if c_1 is incorrect, c_2 and c_3 must be correct. Since the errors of a cell will be carried off during the compression, the condition that only one cell of each area $\mathbf{x}+K$ ($\mathbf{x}\in\mathbf{Z}^2$) may be incorrect, cannot maintained. The second modification of the K-se condition runs as follows. If a cell $\mathbf{y}\in(\mathbf{x}+K)$ is incorrect all other cells of that region have to work correctly, i.e. they are only

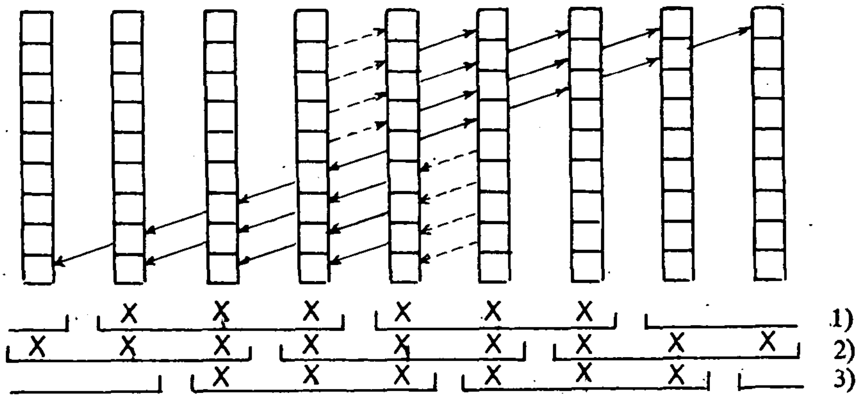


Fig. 7
Information-flow through one cell during compression

allowed to take over the errors of cell y. If the qualifier of a cell is incorrect, in general that cell must be considered as totally broken down. Therefore we must demand that no errors occur in the qualifier of a cell. Only under this new stronger K-se condition the above L-processor may correct errors.

Theorem 2. Let K be the 21-element template of Fig. 3. Then the above L-processor $\mathfrak{A}_2$ corrects any errors if the stronger K-se condition holds.

Since the proof is similar to that of Theorem 1, we do not carry it out here again.

5. Conclusions and outviews

The primary objective of this paper has been to present suggestions for fault tolerant L-processors. The first solution is characterized by the following features:

- $A_2=A_1$ and $Q_2=Q_1\times \mathrm{FP}$, i.e. the set of external states is not increased,
- the real L-processor $\mathfrak{A}_2$ has exactly the same number of local transition functions as the simulated L-processor $\mathfrak{A}_1$,
- $\mathfrak{A}_2$ simulates $\mathfrak{A}_1$ in realtime,
- $\mathfrak{A}_2$ has a relatively large neighbourhood template of 21 elements.

The features of the second solution are:

- $A_2=A_1^9$ and $Q_2=Q_1\times \mathrm{FP}\times \mathrm{MZ}$, i.e. the set of external states of $\mathfrak{A}_2$ is enlarged, too,
- the real L-processor $\mathfrak{A}_2$ has either the same number of local transition functions as the simulated L-processor $\mathfrak{A}_1$ (compression per hardware) or at most one local transition function more than $\mathfrak{A}_1$ (compression per software),
- $\mathfrak{A}_2$ simulates $\mathfrak{A}_1$ in 5-slow,
- $N_2=H_1$,
- very strong K-se condition.

As shown above the second solution is not practicable for error correction, because the simulating L-processor must have nearly no errors. Therefore a fault tolerant L-processor with von Neumann neighbourhood template has to use some different coding.

It is easy to modify the above solutions so that the resulting L-processors may detect errors.

Acknowledgements. I am deeply grateful to my dear colleague Mr. W. O. Höllerer for some suggestions in formal describing L-processors. I would also like to thank Prof. R. Vollmar for many stimulating discussions.

Abstract

This paper treats the problem of designing L-processors with error correction capabilities. First such notions as configuration, real L-processor, K-separated error, and simulation are defined. Then a fault tolerant real L-processor is introduced which simulates a given L-processor in real time. Next a fault tolerant real L-processor with von Neumann neighbourhood index is presented which simulates a given L-processor in 5-slow. In both cases the original L-processors have a von Neumann template. Since an L-processor may be understood as a cellular space with Mealy-type cells, this approach may be considered as a generalization of the work of Nishio and Kobuchi.

LEHRSTUHL D FÜR INFORMATIK
TECHNISCHE UNIVERSITÄT BRAUNSCHWEIG
GAUSSTR. 11
D—3300 BRAUNSCHWEIG

References

[GROS 79] GROSS, S. R., Fehlererkennung und -korrektur in zellularen Automaten, Diplomarbeit, Braunschweig, 1979.

[HANO 75] HARAO, M., S. NOGUCHI, Fault tolerant cellular automata, *J. Comput. System Sci.*, v. 11, 1975, pp. 171—185.

[LEGE 76] LEGENDI, T., Cellprocessors in computer architecture, *Computational Linguistics and Computer Languages*, v. 11, 1976, pp. 147—167.

[LEGE 77] LEGENDI, T., Programming of cellular processors, *Informatik-Berichte*, Nr. 7703, Braunschweig, 1977, pp. 53—66.

[NIKO 75] NISHIO, H., Y. KOBUCHI, Fault tolerant cellular spaces, *J. Comput. System Sci.*, v. 11, 1975, pp. 150—170.

[VOLL 79] VOLLMAR, R., *Algorithmen in Zellularautomaten*, Teubner Verlag, Stuttgart, Bd. 48, 1979.

[WRIG 76] WRIGHT, L. E., Cellular automata with nonworking cells, Ph. D. diss., Rensselaer Polytechnic Institute, Troy, New York, 1976.

(Received April 10, 1981)

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1981.
Megjelent: 1981. december hó
Példányszám: 1025. Terjedelem: 12 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 5601 és az MSZ 5602—55 szabvány szerint
81-3258 — Szegedi Nyomda — F. v.: Dobó József igazgató