MATHEMATICA — REVUE D'ANALYSE NUMÉRUQUE
ET DE THÉORIE DE L'APPROXIMATION
L'ANALYSE NUMÉRIQUE ET LA THÉORIE DE L'APPROXIMATION
Tome 10, Nº 1, 1981, pp. 33—47

INFORMATION STORAGE AND RETRIEVAL SYSTEMS WITH INEXACT QUERIES

by

C. JALOBEANU
(Cluj-Napoca)

1. Introduction

This paper is concerned with a mathematical approach to some problems being involved in information storage and retrieval systems, when the retrieval request is not precise. An imprecise query can be either incomplete or have some errors.

Studies on mathematical foundation of information storage and retrieval system have been first published in the papers [1—4].

The origines of the present studied systems could be found in the need to organize a data base for a mass-spectrometry laboratory. The data base containing the mass-spectra of chemical substances could be used for substance identification The problem is like this: by a physico-chemical analysis, a function $I = f(m)$ defined on a real bounded and closed interval, is put in correspondence with a given substance. The function with a particular shape, has a constant value on the hole interval excepting some neighbourhoods of the points m_i. In m_i the function has a local maximum $I_i = f(m_i)$. The pairs (m_i, I_i), $i = \overline{1, n}$, $m_i \in \mathbb{N}$, $I_i \in \mathbb{R}$, characterize the analysed substance. A mass spectra is a sequence of pairs (m_i, I_i), $i = \overline{1, n}$. The values m_i correspond to the fragments masses of the substance and the values I_i represent the corresponding ionic currents.

Theoretically, to a certain mass-spectrum corresponds a unique substance and this fact allows the identification of substances using a sufficiently large catalogue with mass spectra. Until now, there have been catalogued approximately 200,000 mass spectra. Practicaly, the measured mass spectrum depends on the condition of the analysis and on the characteristics of the measure device. In this way, the mass spectrum obtained in an analysis does not coincide with any catalogued spectra and so difficulties in its identification are risen. If we consider the catalogue stored in a computing system, the information communicated to the system as a query, i.e. the measured mass-spectrum, does not coincide exactly with any information stored in the files. The differences appear to I_i, and they lead to differences in the order of the sequence (m_i, I_i), $i = \overline{1, n}$.

This problem leads us to consider a special type of information storage and retrieval systems namely hierarchical systems, which allows the processing of inexact queries. We have studied two more problems on queries: the problem of error detection and the problem of error recovery. By error recovery we mean here to find a set of correct queries, which can „approximate" the erroneous query.

Our study is based on the mathematical model introduced by M. Marek and Z. Pawlak and some other results due to W. Lipski and M. Jaegermann.

2. Information Storage and Retrieval Systems in Marek — Pawlak Sense

Let X be a set of objects (like books, document sor chemical substances), wich must be identified mostly by their description. For description there is used a finite set of criterions, which will be referred to as the set of attributes. Let I be the set of attributes. An object has, in respect with an attribute, one descriptor. For example, if we consider the documents as objects, an attribute of them could be their issue data. A corresponding descriptor can be 1980. The set of descriptors will be denoted by symbols from a set A considered as an alphabet. One considers an equivalence relation R_I on the set of descriptors, aR_Ib iff a and b are descriptors for the same attribute. The relation R_I generates a partition $\{A_i\}_{i \in I}$ of A into families of equivalence classes: $A = \bigcup_i A_i$ and if $i \neq j$ then $A_i \cap A_j = \emptyset$. It is denoted by $\mathcal{L}_A$ the description language corresponding to the alphabet A. $\mathcal{L}_A$ is a sort of intermediate language between propositional and predicate calculi.

By an information storage and retrival system it is to be understood a quadruple consisting of a set of objects X together with the set of descriptors A, the set of attributes I and a function U which associates a subset of X to each descriptor from A. Thus each object from X may be described in the system by a vector of descriptors from A, exhausting all posible attrtibutes from I.

DEFINITION 2.1. *An information storage and retrieval system (i.s.r. system) is a quadruple*

$$\mathcal{J} = (X, A, R_I, U)$$

where X is the set of objects, the carrier of the $\mathcal{J}$; A is the set of descrip-, tors and R_I is an equivalence in A of finite index. U maps A into $\mathcal{P}(X)$ $(U : A \to \mathcal{P}(X))$ and satisfies the following two conditions:

(1) *if aR_Ib and $a \neq b$, then $U(a) \cap U(b) = \emptyset$*

(2) *$\bigcup\{U(b) | bR\, a\} = X$ for each $a \in A$.*

DEFINITION 2.2. *Let $\mathcal{J} = (X, A, R_I, U)$ be an i.s.r. system and $x \in X$, then*

(a) *an information on x in $\mathcal{J}$ is a function $f_x : I \to A$ such that for all $i \in I$, $f_x(i) \in A$ and $x \in U(f_x(i))$;*

(b) *a description of x in $\mathcal{J}$ is $t_x = \prod_{i \in I} f_x(i)$.*

Using the above defined system the authors introduce in [1] the notion of describable set of objects. Since not all subsets of X are describable, as a general fact, there has been investigated the structure of the family of the describable set. There has also been investigated some dynamical aspects of the system, the case of adding or removal of some atributes and/or descriptors from the system. There has been introduced some algebraic operations on i.s.r. systems.

In the next sections we will introduce a system with a hierarchical structure of attributes. This structure allows the system to verify the correctness of the queries and to point up the errors.

In the subsequent we shall consider the language of our hierarchical system as a subset of the semigroup generated by the set of descriptors.

3. The Semigroup of Descriptors

Let A be the set of descriptors. The finite set of symbols A will be used as an alphabet. Let A^* be the semigroup with unity generated by the alphabet A, with the concatenation operation, verifying the condition:

(1) for all $a, b, c \in A$ $a(bc) = (ab)c$

(2) there exists T, the unity symbol, such that $Ta = aT = a$, for all $a \in A$.

Let us consider a symbol F, such that (3) $Fa = aF = F$, for all $a \in A$. The symbol F will be called the error symbol.

The set $A^* \cup \{F\}$ forms a semigroup with unity and zero elements. In $A^* \cup \{F\}$ we will consider the well known partial ordering of words:

DEFINITION 3.1. *The words $m_1, m_2 \in A^*$ are in the relation $m_1 \leqslant m_2$ iff there exists a pair $a, b \in A^*$ such that $m_2 = am_1b$ and at least one from $a, b \neq T$.*

The words m_1, m_2 are incomparable, $m_1 \mathrel{+\!\!\!\!\|} m_2$, iff neither $m_1 \leqslant m_2$ nor $m_2 \leqslant m_1$. Obviously, $a \leqslant F$ and $a \geqslant T$ for all $a \in A$.

In this way $A^* \cup \{F\}$ is a partial ordered semigroup in respect with the relation $\leqslant$. The element F is a maximal element and T a minimal element. It comes immediately that if $m_1 \leqslant m_2$ and $a \in A^* \cup \{F\}$ then $m_1 \leqslant m_2a$ and $m_1 \leqslant am_2$. Moreover, $a^2 \geqslant a$, for all $a \in A$.

4. Hierarchical Information Storage and Retrieval Systems.

Let A be a set of descriptors and Y the set of attributes. We will consider a partial application.

$$\delta : Y \times A \to Y \text{ with } Y \setminus \mathrm{Imag}_1\delta \neq \emptyset.$$

The natural extension of δ to $Y \times A^*$ is $\delta(\delta(z, a), b) = \delta^*(z, ab)$, for $z \in Y$ and $\delta^*(z, T) = z$ for all $z \in Y$. We will note $\delta^*(y,m) = \delta(y,m)$, $m \in A^*$.

DEFINITION 4.1. *It will be called hierarchical information storage and retrieval system, a system : $S = (X, Y, A, \delta, U)$, where X is a finite set of objects, Y is a finite set of attributes, A is the set of descriptors, the partial application*

$\delta : Y \times (A^* \setminus T) \to Y$ is an injection and

$U : Y \times A \to \mathcal{P}(X)$ is a partial application, such that the following conditions to be fulfiled :

(1) $U(y, a) = \bigcup_{b \in A} U(\delta(y, a), b)$

(2) if *$a \neq b$, then $U(y, a) \cap U(y, b) = \emptyset$*

(3) *$U(y, a) = \emptyset$. if and only if there exists a $z \in Y$ such that $\delta(y, a) = z$.*

Coming back to our example of the spectra, the set of objects X will be the set of catalogued chemical substances.

It could be considered the chemical substance described only by the sequence of fragments masses m_i, $i = \overline{1, n}$,. The sequence is ordered in respect with the corresponding ionic curents. In this case the set of descriptors A will be the set of natural numbers, but we must note that in a description of an object the order of the descriptors will be essential.

Based on the extension of δ, the extension of U to the set $Y \times \overline{A^*}$ could be done. It comes about, from the axiom (1) that

$U(\delta(y, a), b) = U(y, ab)$, for all $a, b \in A$. Indeed, $U(\delta(y, a), b) = \bigcup_{c \in A} U(\delta(y, a), b), c) = \bigcup_{c \in A} U(\delta(y, ab), c) = U(y, ab)$. Let us note that $aF = Fa = F$ implies $U(y, F) = U(y, aF) \subseteq U(y, a)$ for all $a \in A$. For that we can put $U(y, F) = \emptyset$. On the other hand, from $aT = Ta = T$, and from $U(y, a) \supseteq U(y, ab)$ for all $a, b \in A$, results $U(y, T) \supseteq U(y, Ta)$. That means $U(y, T) = X_y$, where X_y is the set of objects having the attribute y and the descriptors ranging over the set of all possible descriptors of y, noted by A_y.

Hence $U(y, T) = \bigcup_{a \in A_y} U(y, a) = X_y$.

We can proof : If $m_1 \nparallel m_2$ and $m_1, m_2 \in A^* \cup \{F\}$, then $U(y, m_1) \cap U(y, m_2) = \emptyset$

Indeed, let us suppose that $m_1 = mam'$ and $m_2 = mbm''$, then

$U(y, mam') = U(\delta(y, m), am') = U(z, am')$

$U(y, mbm'') = U(\delta(y, m), bm'') = U(z, bm'')$

On the other hand, from the axiom (1), it follows that $U(z, am') \subset U(z,a)$ and $U(z, bm'') \subset U(z, b)$.

But as $a \neq b$, the axiom (2) implies

$U(z, a) \cap U(z, b) = \emptyset$, and moreover,

$U(z, am') \cap U(z, bm'') = \emptyset$ and so,

$U(y, m_1) \cap U(y, m_2) = \emptyset$.

The function δ, defined in that manner, allows us to structure the set of attributes Y. Let us denote by Y_0 the set of initial atrtibutes $Y_0 = Y \setminus \text{Imag}\ \delta$. The hierarchical next set is

$$Y_1 = \{\delta(y, a) | y \in Y_0,\ a \in A\}.$$

Similarly

$$Y_n = \{\delta(\delta(\ldots \delta(y, a_1)\, a_2), \ldots, a_n) = \delta(y, a_1 a_2 \ldots a_n) \text{ when } y \in Y_0,\ a_1 a_2 \ldots a_n \in A^*\}.$$

If we assume that

$$Y = \bigcup_{i=1}^{n} Y_i$$

with n a finite integer, then the system has a hierarchical structure with no more than n levels.

DEFINITION 4.2. *An attribute $y \in Y$ is called proper if the set $A_y = \{a \in A | U(y, a) \neq \emptyset\}$ is such that $|A_y| \geqslant 2$.* ($|A_y|$ is the cardinal of the set A_y), *and $|U(y, a)| \geqslant 1$.*

DEFINITION 4.3. *An attribute y is a general criterion of classification if $\bigcup_{a \in A_y} U(y, a) = X$*

PROPERTY 4.1. *Let $S = (X, Y, A, \delta, U)$ be a hierarchical i.s.r. system. If Y is a set of proper attributes and $y \in Y$ is a general criterion of classification then y is an initial attribute.*

Proof. Indeed, let us assume that $y \in Y_1$, $\bigcup_{b \in A} U(y, b) = X$ and let $z \in Y_0$, such that $\delta(z, \alpha) = y$ and $|A_z| \geqslant 2$. We have, from the condition 1 definition 4.1, that $U(z, \alpha) = \bigcup_{b \in A} U(\delta(z, \alpha), b) = \bigcup_{b \in A} U(y, b) = X$. But, if $\alpha' \in A_z \setminus \{\alpha\}$, then $U(z, \alpha') = \emptyset$ and hence $|A_z| = 1$, which is in constradiction with the hypothesis.

LEMMA 4.1. *If $S = (X, Y, A, \delta, U)$ is a hierarchical system, then the application δ verifies the property : $\delta(y, m) \neq y$, for all $y \in Y$ and $m \in A^* \setminus \{T\}$.*

Proof. Let us assume that there exist $y \in Y$ and $m \in A^* \setminus \{T\}$ such that $\delta(y, m) = y$. Then $\delta(y, ma) = \delta(\delta(y, m), a) = \delta(y, a)$. The injectivity of δ implies $ma = a$ and thus $m = T$, which is in contradiction with the hypothesis.

We will show in the sequel that the restriction of the function δ in respect with an initial attribute is a labeled rooted tree.

DEFINITION 4.4. *A directed graph is a pair (Y, Γ), where Y is a finite set of nodes and Γ is a relation which associates to a node his succesor.*

DEFINITION 4.5. *A directed graph is said to be a rooted tree if*

(1) *for all $y \in Y$, $(y, y) \notin \Gamma$* (there are no loops)

(2) *if $(y_1, y_3) \in \Gamma$ and $(y_2, y_3) \in \Gamma$, then $y_1 = y_2$, for all $y_1, y_2, y_3 \in Y$* (there are no circuits)

(3) *there exists only one* $y_0 \in Y$ *such that* $\Gamma^{-1}(y_0) = \emptyset$ (y_0 is the root of the tree).

THEOREM 4.1. *Let* $S = (X, Y, A, \delta, U)$ *a hierarchical i.s.r. system. The graph of the restriction of the function* $\delta : Y \times A^* \Rightarrow Y$ *in respect with an initial attribute* $y \in Y_0$ *is a rooted tree labeled with symbols from* A.

Proof. Let us note that $\delta(y, a) = z$ means that $(y, z) \in \Gamma$ and the corresponding arrow has the label a.

1. From the previous lemma follows that $\delta(y, a) \neq y$ for all $a \in A$ and all $y \in Y$, i.e. the graph has no loops.

2. The injectivity of δ implies that the graph has no circuits.

3. If $y \in Y_0$, then for all $z \in Y$ and $a \in A$ $\delta(z, a) \neq y$, and so y is the root of the tree.

CORROLARY 4.1. *The triple* (Y, δ, A) *is a set of rooted trees with the roots from the set* $Y \setminus \mathrm{Imag}\, \delta$.

5. The Lattice of Queries

The extension of the application U leads to the following application. $U : Y \times (A^* \cup \{F\}) \to \mathcal{P}(X)$.

In the set $Y \times (A^* \cup \{F\})$ we introduce the relation $(y, m) \approx (z, n) \Leftrightarrow U(y, m) = U(z, n)$. It is obvious that the relation is an equivalence.

DEFINITION 5.1. *The attributes* $y, z \in Y$ *are comparable if there exist* $w \in Y$ *and* $m, n \in A^*$ *such that* $\delta(w, m) = y$ *and* $\delta(w, n) = z$ *or if there exists* $q \in A^*$ *such that* $\delta(y, q) = z$ *or* $\delta(z, q) = y$.

THEOREM 5.1. *The defined equivalence has the following properties:*

a) $(y, m) \approx (y, p) \Leftrightarrow m = p,\ m, p \in A^* \cup \{F\}$;

b) *for* $y, z \in Y$ *comparable attribute,* $(y, a) \approx (z, ma)$ *iff* $\delta(z, m) = y$;

c) $(y, F) \approx (z, F)$, *for all* $y, z \in Y$.

Proof. a) Let us assume that $m \# p$. Then $U(y, m) \cap U(y, p) = \emptyset$, but from $(y, m) \approx (y, p)$, $U(y, m) = U(y, p)$.

b) From the comparability of y, z follows that there exists $w \in Y$ and $p, q \in A^*$ such that $\delta(w, p) = y$ and $\delta(w, q) = z$. Then

$U(y, a) = U(\delta(w, p), a) = U(w, pa)$ and

$U(z, ma) = U(\delta(w, q), ma) = U(w, qma)$

From $(y, a) \approx (z, ma)$ results that $U(w, pa) = U(w, qma)$ and $pa = qma$, namely $p = qm$. It implies $\delta(w, qm) = \delta(w, p) = y$, and $y = \delta(w, qm) = \delta(\delta(w, q), m) = \delta(z, m)$.

Conversely, if $\delta(z, m) = y$, then $U(y, a) = U(\delta(z, m), a) = U(z, ma)$, hence $(y, a) \approx (z, ma)$.

c) From $y \in Y$ and $U(y, F) = \emptyset$ results that $(y, F) \approx (z, F)$, for all $y, z \in Y$.

It results immediately the following

COROLLARY 5.1. *If the attributes* y, z *are comparable, then* $(y, a) \approx (z, a)$ *iff* $z = y$.

Let $\mathcal{A}$ be the set of equivalence classes from $Y \times (A^* \cup \{F\})$, in respect with the equivalence relation $\approx$ defined above. The representative elements of equivalence classes are paths in the rooted tree beginning with the corresponding root.

DEFINITION 5.2. *The elements of the set* $\mathcal{A}$ *will be called queries.*

In the sequel we assume that all initial atrtibutes will be general criterion of classification.

In the set $\mathcal{A}$ we could consider a partial ordering:

$$(y, m) \leqslant (y, n) \Leftrightarrow U(y, m) \supseteq U(y, n)$$

THEOREM 5.2. *If* $(y, m) \leqslant (y, n)$, *then there exists* $p \in A^* \setminus \{T\}$ *such that* $n = mp$.

Proof. At first, let us assume that $m \# n$. Then let $m = qm'$ and $n = qn'$, where $m' \# n'$. If we denote by $z = \delta(y, q)$, we have $U(y, m) = U(z, m')$ and $U(y, n) = U(z, n')$. From the extension of axiom 2 definition 4.1. results that $U(z, m') \cap U(z, n') = \emptyset$ and hence $U(y, m) \cap U(y, n) = \emptyset$ which is in contradiction with the hypothesis $U(y, m) \supseteq U(y, n)$.

On the other hand, if we assume that $m = nq$, $q \in A$, then $U(y, m) = U(y, nq) \subseteq U(y, n)$, in contradiction with the hypothesis $(y, m) \leqslant (y. n)$.

THEOREM 5.3. *The relation* $\leqslant$ *is a partial ordering.*

COROLLARY 5.2. *The equivalence class defined by* (y, F) *is a maximal element and* (y, T) *is a minimal element with respect the relation* $\approx$.

Proof. We will show that $(y, T) \leqslant (y, m) \leqslant (y, F)$, for all (y, m).

1) Let us note that for all $y \in Y \setminus \mathrm{Imag}\, \delta$, $U(y, T) = \bigcup\limits_{a \in A} U(y, Ta) = X \supseteq U(y, m)$; and thus $(y, T) \supseteq (y, m)$.

2) From $U(y, F) = \emptyset$, for all $y \in Y$, results that $U(y, m) \supseteq U(y, F)$ and thus $(y, m) \leqslant (y, F)$.

We will put $(y, T) = \underline{0}$ and $(y, F) = \underline{1}$

Let us define the g.l.b., $(y, m) \wedge (y, n)$ as follows

$(y, m) \wedge (z, n) = \underline{0}$ if $y \neq z$ or $m \# n$

$(y, m) \wedge (y, n) = (y, q)$, where $q \begin{cases} n \text{ if } m = nq,\ p \in A^* \\ m \text{ if } n = mp \end{cases}$

Similarly, the l.u.b. is defined by $(y, m) \vee (z, n) = \underline{1}$ if $y \neq z$ or $m \# n$ and $(y, m) \wedge (y, n) = (y, p)$, where $p = \begin{cases} n \text{ if } n = mq,\ q \in A^* \\ m \text{ if } m = nq \end{cases}$

COROLLARY 5.3. *The set* $\mathcal{A}$ *is a distributive latice with elements* $\underline{0}$ *and* $\underline{1}$.

DEFINITION 5.3. *Let* $S = (X, Y, A, \delta, U)$ *be a hierarchical i.s.r. system and* $\mathcal{A}$ *the corresponding set of queries. The set of maximal elements*

$$D = \{(y, m) \in (\mathcal{A} \setminus \{\underline{1}\}) \mid \forall (y, n) \in \mathcal{A},\ (y, m) \geqslant (y, n)\}$$

is called the set of descriptions in S.

DEFINITION 3.6. *The set* $\mathcal{A} \setminus \{\underline{1}\}$ *is called the language of the hierarchical i.s.r. system.*

6. Relationship Between the Hierarchical Systems and the Marek—Pawlak Systems

THEOREM 6.1. *If $S = (X, Y, A, \delta, U)$ is a hierarchical i.s.r. system, then there exists a relation R_y on $Y_0 \times A$ such that the system $\mathcal{J} = (X, Y_0 \times A, R_y, U)$ is a Marek-Pawlak one.*

Proof. Let R_y be the equivalence relation defined by:

$$(y, a) R_y (z, b) \Leftrightarrow z = y$$

If $(y, a) \neq (y, b)$, that is $\delta(y, a) \neq \delta(y, b)$, then from the condition (2) definition 4.1 results $U(y, a) \cap U(y, b) = \emptyset$. Moreover, if $y \in Y_0$, then it is a general criterion of classification and $\bigcup\limits_{a \in A} U(y, a) = X$.
Thus the axioms of Marek-Pawlak system are fulfilled.

THEOREM 6.2. *If $S = (X, Y, A, \delta, U)$ is a hierarchical i.s.r. system and D its corresponding set of descriptions, then exists a Marek-Pawlak system $\mathcal{J} = (X, D, R_I, \bar{U})$ such that the finest partition defined by S on X be the same to that defined by $\mathcal{J}$ on X.*

Proof. Let R_I the equivalence relation defined on D in the following way:

$$(y, m) R_I (z, n) \text{ iff } z = y, \quad z, y \in Y \setminus \text{Imag } \delta.$$

The mapping $\bar{U}$ is the restriction of U to D.
Let us note that if $(y, m), (y, n) \in D$ and $m \nparallel n$, then $U(y, m) \cap U(y, n) = \emptyset$
On the other hand, from the definition of D results that the set $\{U(y,m)\}$, $(y,m) \in D$ is the finest partition of X realized by the system S: $\bigcup\limits_{(y,m) \in D} U(y,m) = X$
That is, $\mathcal{J}$ is a Marek-Pawlak system.

DEFINITION 6.1. *Let us consider $S = (X,Y,A,\delta,U)$ a hierarchical system whit n levels. Then we shell call restriction with p levels a hierarchical system $S_p = (X, Y_p, A_p, \delta_p, U_p)$, where Y_p is the set of attributes from the first p levels of S, $A_p \subset A$, and U_p, δ_p are the restrictions of U and δ to $Y_p \times A_p$.*

Let us note that to the sequence of hierarchical systems $S_1, S_2, \ldots, S_n$, restrictions with $1, 2, \ldots, n$ levels of the system S, corresponds a sequence of enclosed Marek-Pawlak systems $\mathcal{J}_1, \mathcal{J}_2, \ldots, \mathcal{J}_n$.
The set of partitions defined by that systems coincide with the partition defined by the original hierarchical system S.

7. Objects Retrieval in Hierarchical i.s.r. Systems

DEFINITION 7.1. *Let $S = (X, Y, A, \delta, U)$ be a hierarchical i.s.r. system. An information on $x \in X$, in respect with an attribute $y \in Y$, is a partial injective application*

$$f_x : Y \to Y \times A \text{ such that}$$

(1) $x \in U(f_x(y))$
(2) $f_x(\delta(y, a)) = f_x(\delta(f_x(y))$, *for all* $y \in \text{Dom} f_x$. (We will suppose that $\text{Dom} f_x \neq \emptyset$ for all $x \in X$). If $x \in X$ has no descriptor for an attribute, we put $f_x(y) = (y, F)$.

THEOREM 7.1. *If $(y, m) \in D$ is a description of $x \in X$, then $U(y, m) = U(f_x(\delta f_x)^{n-1}(y))$, when the system S has n levels.*
Proof. We will prove at first the following

LEMMA 7.1. *If $y \in \text{Dom} f_x$, then uniquely there exists a descriptor $a \in A$ so that $f_x(y) = (y, a)$.*

Indeed, from the axiom (2) and from the injectivity of f_x, results that $\delta(y, a) = \delta(f_x(y))$. But as δ is a partial injective application, it results that $(y, a) = f_x(y)$.

Coming back to the proof of the theorem, let us consider $(y, m) = (y, a_1, \ldots, a_n)$ and let $x \in U(y, m)$ Denoting by $y_1 = \delta(y, a_1), \ldots, y_{n-1} = \delta(y_{n-2}, a_{n-1})$ it follows that:

$$U(y, m) = U(y_{n-1}, a_n) \subset U(y_{n-2}, a_{n-1}) \subset \ldots \subset U(y, a_1).$$

From the axiom 1 results that $U(y, a_1) \cap U(f_x(y)) \ni x$. If $f_x(y) = (y, b)$, then $U(y, a_1) \cap U(y, b) \neq \emptyset$, and from the previous lemma, results that $a_1 = b$ and so $f_x(y) = (y, a_1)$. In the same manner, from $\delta(y, a_1) = y_1 = \delta(f_x(y))$, follows that $f_x(y_1) = (y_1, a_2)$, hence

$$f_x(\delta(f_x(y))) = (y_1, a_2), \ \delta(f_x(\delta(f_x(y)))) = \delta(y_1, a_2) = y_2$$

Finally, we can write

$\delta(f_x \ldots \delta(f_x(y)) \ldots) = y_{n-1}$ and $f_x(y_{n-1}) = (y_{n-1}, a_n)$, which means that $U(f_x(y_{n-1})) = U(y_{n-1}, a_n) = U(y, m) = U(f_x(\delta f_x)^{n-1}(y))$.

DEFINITION 7.2. *A system $S = (X, Y, A, \delta, U)$ is called selective if* max (card $(U(y, m))) = 1$, $(y, m) \in D$

THEOREM 7.2. *Let $S = (X, Y, A, \delta, U)$ be a hierarchical i.s.r. system with n levels, Y a set of proper attribute and card $X = 2^n$. If the restriction of δ to every initial attribute is a dichotomical rooted tree, then the system is selective.*

Proof. As it is known, a dichotomic tree with n levels has 2^n terminal nods.

For all $(y, m) \in D$, $U(y, m) \geqslant 1$, since Y has only proper attributes. But as card $X = 2^n$, there results that $U(y, m) = 1$ for all $(y, m) \in D$. That is, the system is selective.

If $(y, m) \in \mathcal{A}$ is a query, then the system's answer will be the set $U(y, m)$.

DEFINITION 7.3. *It is said that the object $x \in X$ is identified, if there exists for an $y \in Y_0$ a unique description $(y, m) \in D$ such that $U(y, m) = \{x\}$.*

THEOREM 7.3. *If the hierarchical i.s.r. system S is selective, then every object could be identified.*

Proof. Let S be a selective system, then for all $(y, m) \in D, |U(y, m)|=1$. From the definition 7.1. results that for all $x \in X$ there exists an attribute $y \in Y$, such that $x \in U(f_x(y))$. That implies the existence of a unique descriptor, $a \in A$, such that $f_x(y) = (y, a)$. Corresponding to (y, a), we may choose a maximal element (z, m) such that $U(z, m) \subseteq U(y, a) \ni x$. Since the system is selective, the unique element from $U(z, m)$ is x.

DEFINITION 7.4. *A query (y, m) for which $U(y, m) = \emptyset$ is called a δ — erroneus query.*

There could be risen the following problems:

1) To find out whether the query (y, m) is δ-erroneous.

2) To find out the condition when a set of descriptions can be attached to a δ-erroneous query in order to approximate it. Moreover, to establish the condition in which the approximation set can be used to supply an answer of the system to a δ-erroneous query.

The following theorem could be the answer of the first problem:

THEOREM 7.4. *If $S = (Y, X, A, \delta, U)$ is a hierarchical i.s.r. system with n levels, then for all queries $(y, m) \in \mathcal{A}$, the system itself can decide if the query is δ-erroneous, or not.*

Proof. Let $(y, m) = (y, a_1 \ldots a_n)$ be a query. The set Y is finite and to every attribute corresponds a finite set of descriptors. If the query is $(y, a_1 \ldots a_n)$, there is to be verified whether a_1 is a descriptor from A_y, at first. Whether it is, we have to verify whether a_2 is a descriptor from A_{y_1}, when $y_1 = \delta(y, a_1)$, and so on. The number of steps is a finite one and after ranging the hole query, if $a_n \in A_{y_{n-1}}$, then the query is δ-correct. The query is δ-erroneus if at one step, its belonging to, does not take place.

8. Verification of Queries

It will be investigated in the subsequent the possibility of error recovery. There is considered that an error is recovered whether a set of descriptions „approximating" the erroneous query was found. That imply to find out the erroneous descriptor, and to replace it by all the symbols that in the same context leads to elements from D.

We must note that it is possible to be an error in a query (y, m) for which $U(y, m) \neq \emptyset$. That is, the desired element $x \in X$ is not in $U(y, m)$. This type of error will be considered here as a semantical error. In order to find out this type of errors and to localize the δ-errors, we will introduce the concept of „verification attribute". For this reason, we analyse the connection between descriptions with different initial attribute.

Let s and v be two initial proper attribute having the corresponding set of descriptors, respectively A_s and A_v. We assume $A_s \cap A_v = \emptyset$, $\bigcup_{a \in A_s} U(s, a) = X$ and $\bigcup_{b \in A_v} U(v, b) = X$. Let D_v and D_s be the sets of descriptions in respect with v and s, and V and S the set of attributes from the rooted trees with v and s as roots.

The following property holds: For all descriptions $(v, m) \in D_v$, there exists $a \in A_s$ such that $U(s, a) \cap U(v, m) \neq \emptyset$. Indeed, as $\bigcup_{a \in A_s} U(s, a) \cap$ $\cap\, U(v, m) \neq \emptyset$, there exists at least an $a \in A_s$ so that $U(s, a) \cap U(v,m) \neq \emptyset$.

THEOREM 8.1. *Let $s, v \in Y \setminus \mathrm{Imag}\, \delta$ and V be the attributes from the tree with the root v. Then, for all $z \in V$ and $p \in A^*$ such that $\delta(v,p)=z$, there exists $ab \in A_z$ and an $a \in A_s$ such that $U(z, b) \cap U(s, a) \neq \emptyset$.*

Proof. If $\delta(v, p) = z$, then $U(v, p) \neq \emptyset$ and $U(v, p) \supset U(v, pb) = U(z, b)$. Since there is a $k \in A^*$ such that $(v, pbk) \in D_v$, $U(z, b) \supset U(z, pbk)$. From $U(v, pbk) \cap U(s, a) \neq \emptyset$ results $U(z, b) \cap U(s, a) \neq \emptyset$. Let be $s, v \in Y \setminus \mathrm{Imag}\, \delta$ and let be $z, w \in V$.

DEFINITION 8.1. *The set of descriptors*

$$P_z = \{a \in A_s | U(s, a) \cap U(w, b) \neq \emptyset, \text{ when } \delta(w, b) = z\}$$

is to be said the set of consistent descriptors with z

THEOREM 8.2. *If $\delta(w, b) = z$, then $P_w \supseteq P_z$.*

Proof. If $P_z = \{a \in A_s | U(s, a) \cap U(w, b) \neq \emptyset,\ \delta(w, b) = z\}$ and

$$P_w = \{a \in A_s | U(s, a) \cap U(y, c) \neq \emptyset,\ \delta(y, c) = w\}, \text{ then}$$

$P_z = \{a \in A_s | U(s, a) \cap U(\delta(y, c), b) \neq \emptyset\}$. From $U(y, c) \supset U(\delta(y, c)\ b)$ results that if $a \in P_w$ then $a \in P_z$.

DEFINITION 8.2. *An initial proper attribute s is called verification attribute for the rooted tree V if for all $(v, m) \in D_v$, the terminal attribute $z = \delta(v, m)$ has the set of consistent descriptors P_z, such that* $|P_z| = 1$. Let us denote the attribute from the i level of V with V_i and let $A_{v_{i-1}}$ be the descriptors with which V_i are reached to. Then $P_i = \bigcup_{z \in V_i} P_z$ are the corresponding descriptors of the verification attribute. We will define the application:

$$d : V_i \times A_{v_{i-1}} \rightarrow \mathcal{L}(P_i)$$

which attaches to a pair formed by an attribute and his corresponding precedent descriptor, a list of consistent descriptors from A_s

THEOREM 8.3. *Let us consider $y, z \in V$, $d(y, a) = P_y$, $d(z, b) = P_z$. If there exists $w \in V$ such that $\delta(w, a) = y$ and $\delta(w, b) = z$, then $P_y \cup P_z \subseteq P_w$.*

Proof. If

$$P_y = \{\alpha \in A_s | U(s, \alpha) \cap U(w, a) \neq \emptyset,\ \delta(w, a) = y\}$$

$$P_z = \{\alpha \in A_s | U(s, \alpha) \cap U(w, b) \neq \emptyset,\ \delta(w, b) = z\},$$

let be $u \in V$, such that $\delta(u, c) = w$, then

$$P_w = \{\alpha \in A_s | U(s, \alpha) \cap U(u, c) \neq \emptyset,\ \delta(u, c) = w\}.$$

From $U(u, c) = \bigcup_{\beta \in A} U(\delta(u, c), \beta)$ results

$$U(u, c) \supseteq U(w, a) \cap U(w, b)$$

The lists P_s of consistent descriptors in respect with an attribute s supplies additional information which could be viewed as a semantical information. A query with the corresponding descriptor of the verification attribute allows to verify if the found objects, got as a result of searching, have or not the verification descriptor. In this way there could be found errors in queries, in spite of their δ — correctness. Moreover, the verification attribute coud be useful to localize the errors.

Let us consider a function

$$\omega : Y \times A \to Y \times A$$

defined as follows:

$$\omega(y, a) = \begin{cases} (\delta(y, a), a) & \text{if } U(y, a) \neq \emptyset \\ (y, F) & \text{if } U(y, a) = \emptyset \end{cases}$$

The function could be extended to $Y \times A^* \cup \{F\})$. For (y, T) and (y, F) the function is defined by the relations:

$$\omega(y, T) = (y, T)$$

$$\omega(y, F) = (y, a) \text{ if } \delta(z, a) = y$$

At the same time

$$\omega(y_1, a_1a_2) = \begin{cases} (y_3, a_2) \text{ if } U(y_1, a_1a_2) \neq \emptyset \\ (y_2, a_1) \text{ if } U(y_1, a_1a_2) = \emptyset \\ \qquad \text{and } U(y_1, a_1) \neq \emptyset, \end{cases}$$

where $y_2 = \delta(y_1, a_1)$ and $y_3 = \delta(y_2, a_2)$
This extension is based indeed on the definition of ω:

if $U(y_1, a_1a_2) \neq \emptyset$ then $\omega(y_1, a_1a_2) = \delta(\delta(y_1a_1), a_2), a_2) = (y_3, a_2)$
if $U(y_1, a_1a_2) = \emptyset$ and $U(y_1, a_1) \neq \emptyset$, then $\omega(y_1, a_1a_2) = \omega(\delta(y_1, a_1), a_2) = \omega(y_2, F) = (y_2, a_1)$.

Generally,

$$\omega(y_1, a_1 \ldots a_n) = \begin{cases} (y_{k+1}, a_k) \text{ if } U(y_1, a_1 \ldots a_k) \neq \emptyset \\ (y_k, a_{k-1}) \text{ if } U(y_1, a_1 \ldots a_k) = \emptyset \text{ and} \\ U(y_1, a_1 \ldots a_{k-1}) \neq \emptyset \end{cases}$$

It is clear that the function ω wipes out all symbols after an error and supplies an incomplete, but δ-correct query. We will define now a predicate ρ related with a query (y, m) and a corresponding verification descriptor $a \in A_s$:

$$\rho(d(\omega(y, m))) = \begin{cases} 0 \text{ if } a \not\in d\ (\omega(y, m)) \\ 1 \text{ if } a \in d\ (\omega(y, m)). \end{cases}$$

DEFINITION 8.3. *It is said that the query* (y, m) *verifies the criterion* s *for* $a \in A_s$ *if* $\rho(d(\omega(y, m))) = 1$.

9. Error Localization

If a δ — error has been occuring, the ω — function, just defined in the last section, detects the first error and cuts down the query, preserving the δ — correct part. The verification attribute allows us to test whether the query is consistent with the verification descriptors or not. Whether it is not consistent, the last character from the query is wiped and the new shorted query is checked up. The wipping out procedure continues until the error occurence is found.

Let (y_0, a_0a_1) be a query for which $U(y_0, a_0a_1) \neq \emptyset$.
We make the following notation:

$$y_1 = \delta(y_0, a_0) \text{ and } y_2 = \delta(y_0, a_0a_1).$$

What we want to know is to find out whether the query is consistent with the descriptor $a \in A_s$. For that, let as see the predicate:

— if $\rho(d(\omega(y_1, a_1))) = 1$, then the query verifies the s-criterion;

— if $\rho(d(\omega(y_1, a_1))) = 0$, then the last character is wiped out and the pair (y_0, a_0) is checked up. If $\rho(d(\omega(y_0, a_0))) = 1$ then the correct part of the query is the pair (y_0, a_0).

In order to point out an error in a query (y, m) with $a \in A_s$ his verification descriptor, the system works as follows.

1) The query is scanned to establish that $U(y, m) \neq \emptyset$. In the next step the query is verified in respect with the attribute s;

— if $\rho(d(\omega(y, m))) = 1$, the query is correct and the system's answer is the set $U(y, m)$;

— if $\rho(d(\omega(y, m))) = 0$ and $m = m'b$, then the last character is erroneous and the pair (y, m') is checked up. The procedure follows as far as a correct pair is got.

2) When $U(y, m) = 0$, the function ω is applied in order to obtain the greatest pair (v, q) for which $U(v, q) \neq \emptyset$. Then the pair (v, q) is checked up in respect with the verification attribute s.

THEOREM 9.1. *If* $S = (X, X, A, \delta, U)$ *is a hierarchical i.s.r. system, for all queries* $(y, m) \in \mathcal{A}$ *and for all corresponding verification descriptors* $a \in A_s$, *the system may prove if an error occurs in the query and points out the first error.*

Proof. Let $(y_1, a_1 \ldots a_n)$ be a query with $a \in A_s$ his verification descriptor, and $U(y_1, a_1 \ldots a_n) = \emptyset$. Applying the function ω we get.

$$\omega(y_1, a_1 \ldots a_n) = (y_k, a_{k-1}) \text{ if } U(y_{k-1}, a_{k-1}) \neq \emptyset.$$

If $\rho(d(\omega(y_k, a_{k-1}))) = 1$ then the first error was with a_k. If $\rho(d(\omega(y_k, a_{k-1}))) = 0$, then the pairs $(y_{k-1}, a_{k-2}), (y_{k-2}, a_{k-3}), \ldots, (y_{k-i-1}, a_{k-i})$ are tested while a pair for which $\rho(d(\omega(y_{k-i}, a_{k-i+1}))) = 1$ is to be found. In that case the first error is with the character a_{k+i}.

10. Errors Recovery

In this section we deal with the case when the query $(y, a_1 \ldots a_n)$ has, in respect with a corresponding descriptor $a \in A_s$, an error at most.

1) We shall assume that a_n is the erroneous character and $U(y_1, a_1 \ldots \ldots a_n) = \emptyset$. By $y_n = \delta(y_1, a_1 \ldots a_{n-1})$ we shall denote the last attribute reached to. Let us build up the set of queries $(y_1, a_1 \ldots a_{n-1}\alpha)$, where $\alpha \in A_y$, ranging over the set of descriptors of y_n.
From this set we select the correct descriptors,

$$R_n = \{\alpha \in A_{y_n} | U(y_1, a_1 \ldots a_{n-1}\alpha) \neq \emptyset \wedge \rho(d(\omega(y_i, \alpha))) = 1\}$$

The set

$$E = \{(y_1, a_1 \ldots a_{n-1}\ \alpha) | \alpha \in R_n\}$$

will be referred to as the approximation set of the initial erroneous query.

2) We shall assume that a_i is the erroneous character. It follows, knowing that there is but one error, that $a_{i+1} \ldots a_n$ is a correct subword. Let us consider the set

$$R_i = \{\alpha \in A_{y_i} | (U(y_1, a_1 \ldots a_{i-1}\alpha) \neq \emptyset \wedge \rho(d(\omega(y_i, \alpha))) = 1\}$$

Then the approximation set for the erroneous query is

$$E = \{(y_1, a_1 \ldots a_{i-1}\alpha a_{i+1} \ldots a_n) | \alpha \in R_i \text{ and }$$
$$U(y_1, a_1 \ldots a_{i-1}\alpha a_{i+1} \ldots a_n) \neq \emptyset \wedge \rho(d(\omega(y_1, a_1 \ldots a_n))) = 1\}$$

3) If we assume that a_1 is erroneous, then

$$R_1 = \{\alpha \in A_{y_1} | U(y_1, \alpha) \neq \emptyset \wedge \rho(d(\omega(y_1, \alpha))) = 1\}$$

and the approximation set could be got as done before, taking into account that $a_2 \ldots a_n$ is a correct subword:

$$E = \{(y_1, \alpha a_2 \ldots a_n) | \alpha \in R_1,\ U(y_1, \alpha a_2 \ldots a_n) \neq \emptyset \wedge$$
$$\wedge \rho(d(\omega(y_1, \alpha \ldots a_n))) = 1\}$$

Hence, if $S = (X, Y, A, \delta, U)$ is a hierarchical i.m.r. system, for all queries $(y, m) \in \mathcal{A}$, containing at most one error, the system's answer is $\bigcup_{(y,p) \in E} U(y, p)$.

DEFINITION 10.1. *It is said that an erroneous query could be corrected if card $E = 1$.*

THEOREMA 10.1. *A query $(y_1, a_1 \ldots a_n)$ with an error with a_n could be corrected if there exists a unique attribute y_{i+1} such that $U(y_{i+1}, a_{i+1} \ldots a_n) \neq \emptyset$.*

Proof. Let R_i be the set of characters which may substitute the error and let us consider the set of attributes that could be reached to from y_i with the descriptors $\alpha \in R_i$. Taking into account that uniquely there is the attribute y_{i+1} such that $U(y_{i+1}, a_{i+1} \ldots a_n) \neq \emptyset$, then let be α^* the descriptor for which $\delta(y_i, \alpha^*) = y_{i+1}$. So the unique approximation query is $(y_1, a_1, \ldots \alpha^* \ldots a_n)$.

REFERENCES

[1] W. Marek, Z. Pawlak, *Information storage and retrieval systems, mathematical foundation.* Theoret. Computer Sc. **1**, 331–354 (1976).

[2] Z. Pawlak, *Mathematical foundation of information retrieval.* C. C. PAS Reports, 101, Warszawa (1973).

[3] E. Wong, T. C. Chiang, *Canonical structure in attribute based file organization.* Comm. A.C.M., 14, 593–597, (1970).

[4] T. Rus, U. Sinn, *An algebraic approach to data organization.* BIT 14, 460–481, (1974).

[5] M. Jaegermann, W. Marek, M. Sobolewski, *Information storage and retrieval systems-mathematical foundation* III. Tree structured-attribute systems, C.C. PAS Reports 214, Warszawa (1975).

[6] M. Jaegermann, *Information storage and retrieval systems-mathematical foundation* IV. Systems with incomplete information. C. C. PAS Reports, 215, Warszawa (1975).

[7] W. Lipski, Jr. W. Marek, *On information storage and retrieval systems.* 215–259, in „Matehematical foundation of Computer Science", ed. A. Mazurkiewicz, Z. Pawlak, Banach Centre Publication, vol. 2. Warszawa (1977).

[8] T. Rus, *Data structures and operating systems.* Edit. Academiei R.S.R., București, John Wiley & Sons, New York (1979).

[9] C. V. Negoiță, *Sisteme de înmagazinare și regăsire a informațiilor.* Edit. Academiei, București (1970).

Received 2. VIII. 1980

Institute of Isotopic and Molecular Technology
3400 Cluj-Napoca, Romania