Mathematica Balkanica

Mathematical Society of South-Eastern Europe
A quarterly published by
the Bulgarian Academy of Sciences – National Committee for Mathematics

New Series Vol.11, 1997, Fasc. 1-2

Theoretical Problems of Data Structures Modeling

Adriana A. Georgieva

Presented by Bl. Sendov

The problem for creating of an uniform database formal description on data modeling is especially topical today and remains a long-term purpose in the development of database design tools and their applications. Here we describe the basic concepts of an approach, which is based on the tools of the linear algebra for data structure modeling. The theoretical basis of the conceptual data model are determined and certain necessary equivalences within transition modeling at data representation are proved in aspect of ANSI/SPARC database standard. The possibilities for implementation of this algebraic approach for hierarchical data structures formalization and design are represented in details.

1. Introduction

Since about 1970 the main focus in database research has been turned to solving the problems of the conceptual modeling in the database systems. With the appearance of the Codd's relational approach [1],[2] many issues of the conceptual database scheme receive extensive solutions. As a result, very important studies on the conceptual modeling, using different methods [3],[4],[6] begin to treat them formally.

In this paper we represent an approach for modeling in natural data structures in which the entity-relationships are subordinated to some determined hierarchy. In this aspect the relational model is an object of research and development with a view to use its advantages according to higher level formailism, mathematical precision and real possibilities for application to modeling of the wide-spread hierarchical data structures. The main purpose of the research is creating of a new algebraic approach for database modeling and development of its possible applications. The concepts of the abstract linear algebra - the theory of modules, are used as a modeling tool and data transformations are represented with ordinary algebraic operations.

2. A formal description

Our purpose is to provide more adequate representation and effective use of algebraic tools for formal description and data organization. To this end we offer an algebraic structure of finite modules, which use only the operations of the linear algebra. In this algebraic data representation every object with its attributes is described as a finite sequence (of length n) of numbers [5],[7].

The relational model of databases can be represented as a pair $< S, R >$, where: 1) S is the set of attribute-domains pairs $S = \{A_1, A_2, \ldots, A_n\}$, where each $A = [c, D]$, such that c is a character string (the name of the attribute) and D is a set of atomic elements (the attribute values); 2) R is a set of relations defined over the subset of S, usually denoted by $R^{(n)}(a_1, a_2, \ldots, a_n)$, where $a_i \in D_i$ for $i = 1, \ldots, n$.

Therefore, from relational view the database is a collection or relations, where every $R^{(n)}(a_1, a_2, \ldots, a_n)$ is a set of n-tuples for some fixed n, where n defines the arity of the relation. This means that if M represents the data from the real world and KR represents the conceptual scheme of the relational model, there exists a bijective mapping $f_1 : M \mapsto KR$, which correlates to every existing relationship between objects of M in one-to-one manner relation $R^{(n)}(a_1, a_2, \ldots, a_n)$ from KR.

On the other hand, we can correlate a finite sequence of length n: $(a_{r1}, a_{r2}, \ldots, a_{rn})$ to each relation $R^{(n)}$, where $\{a_{ri}\}, i = 1, 2, \ldots, n$ is the set of the code values of the correspondent attributes of the object $0_r \in M$. Then, if we denote by KA the conceptual scheme of one model, called further algebraic, there exists a bijective mapping $f_2 : M \mapsto KA$, which correlates to every element from M in one-to-one manner an unique element from KA described by the finite sequence $(a_{r1}, a_{r2}, \ldots, a_{rn})$.

Proposition 2.1 *There exists a bijective mapping $f_3 : KR \mapsto KA$, which correlates to every really existing relationship between n different attributes from KR in one-to-one manner an unique finite sequence (of length n) from KA. Then, $a_{ri} \in D \subset Z$ (here Z is a set of integers), where $r = 1, \ldots, m$ and $i = 1, \ldots, n$.*

In this case the domains $D_1, D_2, \ldots, D_n$ of the object 0_r, considered like sets of all the natural numbers $A_{ri} \in N \subset Z$, corresponding to the code values of the possible n attributes of all the m objects from the real user's area.

Each set D_i can be mathematically modeled by a factor set $Z_{\alpha i} = \{0, 1, \ldots, \alpha_{i-1}\}$ of the remainders q $(0 \leq q \leq \alpha_i)$ from division of every $n \in Z$ to α_i, where $\alpha_i \in Z$ is the number of all the possible values $a_{ri} \in D_i$, i.e. α_i is a dimension of D_i. As a result, each D_i can be described by a ring $(Z_{\alpha i}, \oplus, \otimes)$ corresponding to the operations addition mod α_i : $\oplus$ and multiplication mod α_i : $\otimes$, i.e. a direct sum and a direct product for natural numbers. Here these operations have formal meanings only, for the formation of all the sets of code values of the attributes, but they are used indirectly in the further physical database design.

It is evident that for each $a_{ri} \in D_1, a_{r2} \in D_2, \ldots, a_{ri} \in D_i, \ldots, a_{rn} \in D_n$ for the object 0_r we can construct the corresponding modules:

$$(Z_{\alpha 1}, \oplus, \otimes), (Z_{\alpha 2}, \oplus, \otimes), \ldots, (Z_{\alpha i}, \oplus, \otimes), \ldots, (Z_{\alpha n}, \oplus, \otimes).$$

Every mappping $\rho_i : Z \mapsto Z_{\alpha i}$, which maps any natural number upon its remainder mod α_i, i.e. forms the code, corresponding to the attribute's value, is surjective. This means that in algebraic aspect every set of code values D_i can be described by the module A_i over the ring $Z_{\alpha i}$ ($Z_{\alpha i}$ - module).

In this way, we can define the set KA, the conceptual scheme of really existing data structure M, as an algebraic structure, denoted by $A = \{A_1, A_2, \ldots, A_n\}$, a family of $Z_{\alpha i}$ – modules A_i.

Corollary 2.1. *In general, this formal description used for database conceptual modeling establishes the bijectives f_1, f_2 and f_3 in the following manner:*

object
O_r

f_1 f_2

$R^{(n)}(a_1, a_2, \ldots, a_n)$ $(a_{r1}, a_{r2}, \ldots, a_{rn})$

relation f_3 sequence of length n

Furthermore, this formal description gives possibilities for the transition modeling "conceptual-internal" for a data organization by algebraic operations only.

3. An algebraic model of hierarchical data structure

Based on this formal data representation, we can define an algebraic model of hierachical data structure with specific peculiarities, depending on the requirements of the given user's (external) scheme. It is characterized by the following quantities:

– n: a number of the hierarchical levels;

– $\alpha_1, \alpha_2, \ldots, \alpha_n$: the number of the elements on each level i for $i = 1, \ldots, n; \alpha_i \in Z$ and moreover, $\alpha_1 < \alpha_2 < \ldots < \alpha_n$;

– $c_0, c_1, c_2, \ldots, c_{n-1}$: a number of the subordinate elements of every element from level i into level $i+1$ (the "children" of every element from one level into the next); $c_i \in Z$ (in general $c_0 = 1$, but it is possible also $c_0 > 1$). In the present research it is assumed that for any hierarchical level i: c_i is an average number of the subordinate elements c_{ij} to the element j from level i into level $i+1$. In some cases, one can assume that $c_{ij} = \max(c_{i1}, c_{i2}, \ldots, c_{i\alpha i})$. Then, $\alpha_i = \alpha_{i-1}.c_{i-1} = \alpha_1.c_0.c_1.c_2 \ldots c_{i-1}$ for each $i = 2, \ldots, n$.

Proposition 2.2 *The set $A = \{A_1, A_2, \ldots, A_n\}$, as an algebraic model of hierarchical data structure with n levels, is partially ordered by the inclusion:* $A_1 \subset A_2 \subset \ldots \subset A_i \subset \ldots \subset A_n$.

Every object from level n in this formal representation by the finite sequence $(a_1, a_2, \ldots, a_n)$ can be expressed as a linear combination:

$$(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} a_i.e_i = a_1.e_1 + a_2.e_2 + \ldots + a_n.e_n$$

of the generators (base elements) for the structure A:

$$e_1 = (1,0,\ldots,0),\ e_2 = (0,1,0,\ldots,0),\ \ldots\ , e_n = (0,0,\ldots,0,1).$$

Obviously, each e_i expresses the participation of an element from level i of the hierarchy in this representation. From the other hand, $(a_1, a_2, \ldots, a_n) = \sum_{i=1}^{n} b_i.h_i$, where:

– $b_i = a_i$ for $i = 1$ and $b_i = a_i/(c_1, c_2, \ldots, c_{i-1})$ for $i = 2, \ldots, n$;

– $h_i = (0, 0, \ldots, c_1.c_2 \ldots c_{i-1}, \ldots, 0)$ and $h_1 = e_1$

are sequences of length n, called caharacteristic elements, because they reflect the corresponding number of the subordinate elements for each hierarchical level.

In connection of this algebraic modeling of the transition "conceptual-internal", it is defined a mapping $\Phi : A \longmapsto P$, which correlates to every finite sequence $(a_1, a_2, \ldots, a_n)^r \in A$ in one-to-one manner a fixed integer $p_n^r \in P$. In this way, p_n^r uniquely defines the place of the object O_n^r from level n in the real structure M, i.e. its address in the physical database design P. It transforms the characteristic elements in the following way:

$$\begin{aligned}
\Phi(h_1) &= c_0 = 1 \\
\Phi(h_2) &= c_0.c_1 = c_1 \\
\Phi(h_3) &= c_0.c_1.c_2 = c_1.c_2 \\
&\ldots\ldots\ldots \\
\Phi(h_n) &= c_0.c_1.c_2\ldots c_{n-1} = c_1.c_2\ldots c_{n-1}.
\end{aligned}$$

In other words, $\Phi : Z^n \mapsto Z$ and we have the matrix:

$$C_\Phi = [1 \quad c_1 \quad c_1.c_2 \quad \ldots \quad c_1.c_2\ldots c_{n-1}], \quad \text{when } c_0 = 1.$$

For further formal and correct description of data transition "conceptual-internal", necessary for the physical data representation, we prove the following proposition.

Proposition 2.3. *The mapping $\Phi : A \mapsto P$, modeling the transition "conceptual-internal" for hierarchical data structures, is:*

A) bijective;

B) linear mapping.

Proof. A) The proof is based on the algorithm for the physical address determination of each element r from level k in common hierarchy, so that to every sequence of length k the ampping Φ correlates in one-to-one manner an unique integer p_k^r in the following way:

$$\begin{aligned}
p_k^r &= \sum_{i=1}^{k-1} \alpha_i + a_k^I = \alpha_1 + \alpha_2 + \ldots + \alpha_{k-1} + a_k^I \\
&= \alpha_1\Phi(h_1) + \alpha_1\Phi(h_2) + \ldots + \alpha_1\Phi(h_{k-1}) + a_k^I = \alpha_1 \sum_{i=1}^{n} \Phi(h_i) + a_k^I,
\end{aligned} \tag{1}$$

where

$$a_k^I = \{\ldots\{(a_1 - 1).c_1 + (a_2 - 1)\}.c_2 + \ldots + (a_{k-1} - 1)\}.c_{k-1} + a_k - 1 \tag{2}$$

is the serial number of the code value a_k in the hierarchical level k. The computation by formula (2) is based on the formal description of the sets of attribute code values as modules.

Therefore, for each object 0_k^r from level k:

$$\Phi\left((a_1, a_2, \ldots, a_k)^r\right) = p_k^r \in P.$$

On the other hand, each $p_k^r \in P$ is an image of only one sequence of length $k : (a_1, a_2, \ldots, a_k)^r \in A$, i.e. there exists the mapping $\Phi^{-1} : P \mapsto A$, which is modeling the oposite transition "internal-conceptual". This mapping gives a possibility to compute every component of the sequence $(a_1, a_2, \ldots, a_n)^r$: every $a_i \in Z_{ci}$, i.e. to define every code value of the corresponding attribute (elements of hierarchical level) of the considered object in the following order:

$$a_k^I = p_k^r - \alpha_1 . \sum_{i=1}^{k-1} \Phi(h_i) \quad (a_k \text{ will be obtained by formula (2)});$$

$$a_{k-1} = [p_k^r / c_k] + 1 \longrightarrow p_{k-1}^r = \alpha_1 . \sum_{i=1}^{k-2} \Phi(h_{i-1}) + a_{k-1};$$

$$(3) \qquad a_{k-2} = [p_{k-1}^r / c_{k-1}] + 1 \longrightarrow p_{k-2}^r = \alpha_1 . \sum_{i=1}^{k-3} \Phi(h_{i-2}) + a_{k-2};$$

....

$$a_2 = [p_3^r / c_3] + 1 \longrightarrow p_2^r = \alpha_1 . \sum_{i=1}^{1} \Phi(h_2) + a_2;$$

$$a_1 = [p_2^r / c_2] + 1.$$

The integer part from the devision, arising from the definition of the components $a_i \in Z_{ci}$ is denoted by $[\cdot]$ in formula (3).

In other words, $\Phi^{-1}(\Phi((a_1, a_2, \ldots, a_k)^r)) = (a_1, a_2, \ldots, a_k)^r$. i.e. Φ is bijective.

B) Obviously,

$$\text{i)} \quad \Phi((a_1, a_2, \ldots, a_k)^r + (a_1, a_2, \ldots, a_k)^q) = p_k^r + p_k^q$$
$$= \quad \Phi((a_1, a_2, \ldots, a_k)^r) + \Phi((a_1, a_2, \ldots, a_k)^q),$$

$$\text{ii)} \quad \Phi(\lambda(a_1, a_2, \ldots, a_k)^r) = \lambda . p_k^r = \lambda . \Phi((a_1, a_2, \ldots, a_k)^r).$$

because p_k^r and $p_k^q \in Z$ and λ is any integer number, $\lambda \in Z_{ci}$, which presents the displacement of the address in the physical database design - the object place in the hierarchical structure.

Therefore, the mapping $\Phi : A \mapsto P$ is a linear mapping. ■

In this way, the data transition "conceptual-internal" is modeled by means of a bijective linear mapping (function) for data transformation in both directions of this representation. This function permits exact and simple calculation of the physical address p_k^r of each object 0_k^r from the real data structure by formula (1) and on contrary, a determination of the code values of all the attributes $\{a_{ri}\}$ of object 0_r by formula (3). As a result, all the basic operations connected with the data creating and support on the physical (internal) level and output data processing can be realized by ordinary linear calculus.

So formally, for every object from the real hierarchical data structure, a definite meaning can be obtained, on conceptual and physical views in dependence of the purposes and character of the user application with the following representation:

	f_2		Φ	
0_r		$(a_1, a_2, \ldots, a_k)^r$		P_r
object from hierach.structure		sequence from algebr.structure A		address from phys.structure P

Furthermore, one can illustrate briefly the possibilities of this algebraic representation of the binary realtionships by the simpler elements of the algebraic structure A by pairs of code values of elements which correspond to any two levels k and l:

(a_k, a_l), where $a_k \in Z_{ck}$, $a_l \in Z_{cl}$ for each $k, l = 1, \ldots, n \quad (k < l)$.

For the transition "conceptual-internal" in this case we can define the mapping $\Phi_{kl} : A^2 \mapsto P^2$. It correlates to every ordered pair $(a_k, a_l) \in A^2 \subset A$, presenting the binary relationships between elements from any couple "$k - l$" hierarchical levels two unique matching numbers to their addresses in the physical data structure: $(p_k, p_l) \in P^2 \subset P$ in the following way:

$$
\begin{aligned}
p_k &= (a_k^I + \lambda_{k-1}).e_k + 0.e_l = a_k^I + \sum_{i=1}^{k-1} \alpha_i \\
&= a_k^I + \alpha_1 . \sum_{i=1}^{k-1} \Phi(h_i), \\
p_l &= (a_l^I + \lambda_{l-1}).e_k + 0.e_k = a_l^I + \sum_{i=1}^{l-1} \alpha_i \\
&= a_l^I + \alpha_1 . \sum_{i=1}^{l-1} \Phi(h_i),
\end{aligned} \tag{4}
$$

where:

a_k^I and a_l^I are obtained by formula (2);

e_k and e_l are generators (base elements) in the structure of all the possible pairs in A^2, which present the relationships between different elements from any couple "$k - l$" hierarchical levels ($k, l = 1, \ldots, n$ and $k < l$);

λ_{j-1} is a natural number characterizing the number of the elements downward in the tree for each level j; λ_j is connected with the images $\Phi(h_i)$ of the lements h_i in the following way:

$$\lambda_{j-1} = \alpha_1 . \sum_{i=1}^{j-1} \Phi(h_i).$$

The linear mapping Φ_{kl} facilitates the modeling in the natural hierarchical data structures and supports efficient systems for their application, because it is not necessary to use sequences of length n from code values of elements, expressing the participation of all the n hierarchical levels (attributes of objects) by the formal description. It permits the following representation of the transition "conceptual-internal":

$$(a_k, a_l) \in A^2 \subset A \quad \stackrel{\Phi_{kl}}{\longrightarrow} \quad (p_k, p_l) \in P^2 \subset P$$

and opposite data transition:

$$(p_k, p_l) \in P^2 \subset P \quad \stackrel{\Phi_{kl}^{-1}}{\longrightarrow} \quad (a_k, a_l) \in A^2 \subset A.$$

As it can be seen, it is enough to calculate only two components (instead of n in the algebraic structure A) which in one-to-one manner defines the realtionships in considered data structure, because of the existing hierarchical rules.

Therefore, the possibility to use the mappings Φ_{kl} and Φ_{kl}^{-1} for data transition modeling increases the effect of the representation of the hierarchical data structures from user's (external) and physical (internal) views of database.

4. Conclusion

In this paper we represent the basic principles of one algebraic approach for formalization and design of wide range of data structures which is enough powerful, especially for data bases with natural hierarchical structures.

The purpose of this algebraic data representation on the conceptual database scheme is to permit the implementation of the tools of linear algebra and to eliminate more difficult operators of the relational algebra (or relational calculus) in various database applications in AI an paralel processing.

References

[1] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, **13**, No 6, 1970.

[2] E. F. Codd. Extending the data base relational model to capture more meaning. *ACM, Trans. on Data Base Systems*, 1979.

[3] J. Minker. Binary relations, matrices and inference development. *Information Systems* **1**, 1978.

[4] Y. E. Lien. Hierarchical schemata for relational databases. *ACM, Trans. on Database Systems*, **6**, 1981.
[5] L. Garding, T. Tambour. *Algebra for Computer Science*, Springer-Verlag, New York, 1988.
[6] D. Maier. *Database Theory*, Computer Science Press, Rockville, Md, 1982.
[7] A. Georgieva. Theoretical research in conceptual database scheme. *Proc. 13th Int. Summer School "Appl-s of Maths in Techniques"-Varna, Sept. 1987*, 120-131.

Technical University of Sofia
Institute for Applied Mathematics & Informatics
Sofia - 1156,
BULGARIA

Received 18.10.1995