データベース・システム 82－9
（1991. 3. 11）

深い入れ子代数による非正規型関係のクラス分け

北川博之* 国井利泰** 大保信夫*
* 筑波大学電子・情報工学系
** 東京大学理学部情報科学科

入れ子関係（非正規型関係）を対象とした入れ子代数は、これまで何種類もが提案されているが、大きく浅い入れ子代数と深い入れ子代数に分類できる。前者が入れ子関係の最も外側のレベルのみを操作対象とするのに対し、後者は内部の入れ子構造を直接操作可能である。これまでに、浅い入れ子代数のNESTとFLAT演算子により定義された、いくつかの重要な入れ子関係のサブクラスが知られている。本稿では、深い入れ子代数のNEST/FLAT演算子の諸性質から、浅い入れ子代数の代わりに深い入れ子代数をその定義に用いても、各サブクラス自身は変化しないことを示す。

Classification of Nested Relations under Deeply Nested Algebra

Hiroyuki KITAGAWA*, Tosiyasu L. KUNII** and Nobuo OHBO*
* Institute of Information Sciences and Electronics, University of Tsukuba
** Department of Information Science, Faculty of Science, The University of Tokyo

Nested relational algebras are classified into the shallowly nested algebra and the deeply nested algebra. The former restricts application of operations to the outermost level of nested relations, while the latter allows direct manipulation of internal table structures. Several interesting subclasses of nested relations were identified under the shallowly nested NEST and FLAT operations. In this paper, we study classification of nested relations under the deeply nested NEST/FLAT, and prove that each subclass defined under the deeply nested NEST/FLAT is equal to its counterpart defined under the shallowly nested NEST/FLAT.

1. Introduction[†1]

A considerable amount of research effort has been devoted to the study of nested relations since the late 1970's. The first study on the design of nested relations was done by Makinouchi in 1977 [17]. The authors proposed the *nested table data model (NTD)* as an underlying construct for office form handling in 1979 [10] and later in [11, 12, 13, 14]. In addition to extensions of the standard relational algebra operations, two algebraic operations, *NEST* and *FLAT*, were defined and their basic properties were studied in 1980 [11]. Jaeschke and Schek presented a similar study on relations which include set values in 1982 [8]. Fischer and Thomas formally defined a full set of operations for nested relational algebra in 1983 [5]. Later, various formulations of nested relational algebra have been proposed by several researchers [1, 2, 3, 4, 7, 9, 18, 19, 20]. Some of them restrict application of operations to the outermost level of nested relations [5, 7, 8, 19], while others allow direct application of operations to internal table structures [1, 3, 4, 9, 11, 14, 20][†2]. In this paper, we generically refer to an algebra with the former property as a *shallowly nested algebra*, and one with the latter property as a *deeply nested algebra.*

The nested relational algebra by Fischer and Thomas [5] is a well known instance of the shallowly nested algebra. Most theoretical studies on nested relations have been based on the shallowly nested algebra because of its logical simplicity. However, under the shallowly nested algebra, sequences of NEST and FLAT (also referred to as UNNEST) are required to manipulate internal table structures. Moreover, because of irreversibility of FLAT [8, 11], technique such as "tagging" is sometimes mandatory to prevent information loss. Under the deeply nested algebra, manipulation of nested relations can be expressed more succinctly, since algebraic operations are directly applicable to internal table structures without sequences of NEST and FLAT. To name some examples, algebra of the nested table data model (NTD) [14, 15], Jaeschke's nonrecursive algebra [9], algebra of Deshpande and Larson [4], and Colby's recursive algebra [3] are instances of the deeply nested algebra.

In this paper, we study subclasses of nested relations defined under the deeply nested NEST and FLAT, which can directly create and remove internal table structures, respectively. Van Gucht and Fischer identified a number of interesting subclasses of nested relations under the shallowly nested NEST and FLAT, which can only manipulate the outermost subtable structures [6, 21]. They include the "Normalization Lossless Structures," the "Nested Relations,"[†3] the "Permutable Nested Relations," and the "Hierarchical Structures." A nested relation T is a Normalization Lossless Structure, iff $T = \omega^*(\mu^*(T))$ for some sequence μ^* of flat operations such that $\mu^*(T)$ is a flat relation and for some sequence ω^* of nest and flat operations. Here, "flat" and "nest" stand for the shallowly nested FLAT and NEST, respectively. When we replace ω^* with some sequence ν^* of nest operations, we get the definition of Nested Relations. Furthermore, T is a Permutable Nested Relation, iff $T = \nu^*(\mu^*(T))$ for some sequence μ^* of flat operations and any sequence ν^* of nest operations such that T and $\nu^*(\mu^*(T))$ have an identical schema.

We introduce several subclasses of nested relations in analogy with the above subclasses but under the deeply nested NEST/FLAT. We then show that each of these subclasses is equal to its counterpart defined under the shallowly nested NEST/FLAT. The former definition based on the deeply nested NEST/FLAT is more intuitively understandable, while the latter that is based on the shallowly nested NEST/FLAT lends itself better to theoretical analysis. Our study is based on the deeply nested NEST and FLAT, provided by the nested table data model (NTD) [14, 15]. The study in this paper also clarifies some interesting properties of the deeply nested NEST and FLAT operations.

The remaining part of the paper is organized

†1 This paper is a revised edition of [16].

†2 In some nested relational algebras, internal tables can be manipulated in a very restricted way.

†3 The term "nested relation" was used to refer to instances of a specific subclass of nested relations in the definition by Van Gucht and Fischer. To avoid the confusion, we use the capitalized initial letters.

as follows. Section 2 introduces the deeply nested NEST and FLAT, and clarifies their basic properties. Section 3 discusses sequences of NEST and FLAT. In Section 4, we define three subclasses of nested relations, in analogy with the Normalization Lossless Structures, the Nested Relations, and the Permutable Nested Relations, under the deeply nested NEST/FLAT. Then, we show that each subclass is equal to its counterpart originally defined under the shallowly nested NEST/FLAT. In Section 5, we introduce hierarchical nest operation, HNEST, to study another subclass: the Hierarchical Structures. Section 6 is the conclusion.

2. Deeply Nested NEST and FLAT

2. 1. Basic Definitions

As we previously mentioned, we use the *nested table data model (NTD)* as a basis of our study. In NTD, nested relations are referred to as *nested tables (NTs)*. NTD provides *nested table operations (NT operations)* for algebraic manipulation of NTs. NT operations form a typical instance of the deeply nested algebra. Here, we give definitions of NTs and NEST and FLAT operations.

A *nested table (NT)* T is defined as the following triple:

$$T = (NN, NS, NO),$$

where NN is an *NT name*, NS is an *NT schema*, and NO is an *NT occurrence*. An NT schema NS is a set of *group schemas* which meet the *tree condition* given later. A group schema GS_i in NS is an expression of the following form:

$$G_i<C_1, \ldots, C_{n_i}> \qquad (n_i \geq 1),$$

where G_i is a name designating the *group*, and C_j ($1 \leq j \leq n_i$) is a name designating a *component* of G_i. Here, group names G_i are different from each other within an NT schema NS, and so are $C_1, \ldots, C_{n_i}$ within a group schema GS_i. If a group G_k appears as a component of G_i, G_k is called a *child* of G_i, and G_i is called a *parent* of G_k. The sets of *descendants* and *ancestors* of G_i are also defined in an obvious way and denoted by $dg(G_i)$ and $ag(G_i)$, respectively. The set of child groups of G_i is denoted by $cg(G_i)$. The other components of G_i are called *fields* and denoted by $cf(G_i)$. The set of components of G_i, namely $cf(G_i) \cup cg(G_i)$, is denoted by $cc(G_i)$, and the sets $dg(G_i) \cup \{G_i\}$ and $ag(G_i) \cup \{G_i\}$ are denoted by $dg+(G_i)$ and $ag+(G_i)$, respectively. Group schemas in NS must satisfy the following *tree condition* :

(a) There exists one group called the *root*, which has no parent.

(b) Every group other than the root has just one parent and is a descendant of the root.

Figure 1 shows a sample NT. The NT schema of this NT consists of the following group schemas:

$$G_1<F_1, F_2, G_2>, \quad G_2<F_3, F_4>.$$

The functions cg, cf, dg, and ag are defined as follows:

$$cg(G_1) = \{G_2\}, \qquad cg(G_2) = \phi,$$
$$cf(G_1) = \{F_1, F_2\}, \qquad cf(G_2) = \{F_3, F_4\},$$
$$dg(G_1) = \{G_2\}, \qquad dg(G_2) = \phi,$$
$$ag(G_1) = \phi, \qquad ag(G_2) = \{G_1\}.$$

G_1			
F_1	F_2	G_2	
		F_3	F_4
X	Y	X	X
		X	Y
		Y	Y
X	Y	Y	X
Z	Y	Y	X

Figure 1. Nested Table

Every field and group has a *domain* of data *occurrences*.

(a) The domain of a field F, denoted by dom(F), is defined as a set of atomic data items.

(b) The domain of a group with group schema $G_i<C_1, \ldots, C_{n_i}>$, denoted by $dom(G_i)$, is defined as follows:

$$dom(G_i) = 2^{dom(C_1) \times \ldots \times dom(C_{n_i})}.$$

Here, we denote with 2^A the powerset of a set A and with $A_1 \times \ldots \times A_n$ the Cartesian product of sets $A_1, ..., A_n$. Elements in $dom(C_1) \times ... \times dom(C_{n_i})$ are called *clusters*. The clusters are called *G_i clusters* to explicitly specify that they can appear in occurrences of G_i.

An NT occurrence NO is an occurrence of the root group G_R. If an NT schema is composed of only one group schema, the NT is called a *flat NT*. Flat NTs are obviously equivalent to relations in the relational model. Given a group $G_i<C_1, ..., C_{n_i}>$ and a G_i cluster t, the data occurrence for component C_j in t is denote by $t[C_j]$. This notation is also used for a subset of components $C \subseteq \{C_1, ..., C_{n_i}\}$.

Primitive NT operations consist of NEST, FLAT, PROJECTION, SELECTION, PRODUCT, UNION, and DIFFERENCE. Definitions of NEST and FLAT are given below. The others are natural extensions of primitive operations of the standard relational algebra. Their formal definitions are given in [14].

Definition 1 : Given a group G_i and $X \subseteq cc(G_i)$ $(X \neq \phi)$, the *NEST* operation $N[G_i, G_j<X>]$ creates as a child of G_i a new group G_j consisting of components X. Here, $G_j<X>$ is a new group schema. Every occurrence O of group G_i is replaced by the following O':

$$O' = \{(t[cc(G_i) - X], FN(t)) \mid t \in O\},$$

where

$$FN(t) = \{u[X] \mid u \in O \wedge u[cc(G_i) - X] = t[cc(G_i) - X]\}.$$

In case G_i is the root, the NEST is referred to as an *outer NEST*. ▭

Note that G_i can be any group in the NT schema in the deeply nested NEST. An example of the NEST operation is given in Figure 2. The shallowly nested algebra only allows outer NEST operations.

Definition 2: Given a group G_i other than the root, the *FLAT* operation $F[G_i]$ removes group G_i, and components $cc(G_i)$ are converted into components of the parent of G_i. Let G_j be the parent of G_i with group schema $G_j<X>$. Then, every occurrence O of group G_j is replaced by the following O':

$$O' = \{(t[X - G_i], u) \mid t \in O \wedge u \in t[G_i]\}.$$

In case G_j is the root (in other words, G_i is a child of the root), the FLAT is referred to as an *outer FLAT*. ▭

Figure 2 includes an example of the FLAT operation. The shallowly nested algebra only allows outer FLAT operations.

G_1						
		G_2			G_3	
F_1	F_2	F_3	F_4	F_5	F_6	F_7
X	Y	X	Y	X	X	Y
		X	Z	Y		
		Z	X	Y		
Y	Z	X	Z	Y	Y	Z
		Z	X	Z	Z	X
		Z	Y	X		

$N[G_2, G_4<F_4, F_5>]$ ↓ ↑ $F[G_4]$

G_1						
		G_2			G_3	
			G_4			
F_1	F_2	F_3	F_4	F_5	F_6	F_7
X	Y	X	Y	X	X	Y
			Z	Y		
		Z	X	Y		
Y	Z	X	Z	Y	Y	Z
		Z	X	Z	Z	X
			Y	X		

Figure 2. NEST and FLAT

2.2. Reversibility and Commutativity of NEST and FLAT

Reversibility and commutativity of NEST and FLAT are essential for the discussion in the remaining part of the paper. Reversibility and commutativity of outer NEST and outer FLAT was studied by some other researchers. Here, we consider properties of NEST and

FLAT in the deeply nested algebra. Some of the following propositions and their proofs are given in our previous work [14, 15].

Proposition 1 (Reversibility of NEST) : For NEST $N[G_i, G_j<C_j>]$ applicable to an NT T, let $T' = N[G_i, G_j<C_j>](T)$. Then, $T = F[G_j](T')$. ▭

Definition 3 : Let G be a group of an NT T, $X \subsetneq cc(G)$, and $Y \subsetneq cc(G)$. Given an occurrence O of G, X *functionally determines* Y in O, if $t[X] = u[X]$ implies $t[Y] = u[Y]$ for every pair of G clusters $t \in O$ and $u \in O$. If X functionally determines Y in every occurrence of G, *functional dependency* $X \rightarrow Y$ holds in T. ▭

Proposition 2 (Reversibility of FLAT) : For FLAT $F[G_i]$ applicable to an NT T, let G_j be the parent of G_i, $C_i = cc(G_i)$ $C_j = cc(G_j)$, and $T' = F[G_i](T)$. If and only if functional dependency $C_j - G_i \rightarrow G_i$ holds in T, $T = N[G_j, G_i<C_i>](T')$. ▭

Proposition 3 (Commutativity of FLATs) : Let G_i and G_j be distinct groups other than the root in an NT T. Then, $F[G_j]F[G_i](T) = F[G_i]F[G_j](T)$[†4]. ▭

Proposition 3 assures that two FLAT operations are always commutative, whatever hierarchical levels they are applied at. On the contrary, NEST does not have this property. To discuss commutativity of NEST operations, we introduce the concept of *weak multivalued dependency* originally identified by Jaeschke and Schek in [8]. Let G be a group, and O be an occurrence of G. The projection of O over $X \subsetneq cc(G)$, $\{t[X] \mid t \in O\}$, is denoted by $O[X]$. The projection of O over $Y \subsetneq cc(G)$ with an X-value x, $\{t[Y] \mid t \in O \wedge t[X] = x\}$, is denoted by $O_x[Y]$. Similarly, the projection of O over Y with an X-value x and a Z-value z ($Z \subsetneq cc(G)$) is denoted by $O_{xz}[Y]$.

Definition 4 : Let G be a group of an NT T, $X \subsetneq cc(G)$, $Y \subsetneq cc(G)$, and $Z = cc(G) - X - Y$. Given an occurrence O of G, X *weakly multidetermines* Y in O, if $O_{xz}[Y] \cap O_{xz'}[Y] \neq \phi$ implies $O_{xz}[Y] = O_{xz'}[Y]$ for every X-value x and Z-values z and z'. If X weakly multidetermines Y in every occurrence of G, *weak multivalued dependency* $X \text{—(w)→→} Y$ holds in T. ▭

Proposition 4 (Commutativity of NESTs) : Let G_k and G_m be groups of an NT T, $X_i \subsetneq cc(G_k)$, $X_j \subsetneq cc(G_m)$, $X_i \neq \phi$, $X_j \neq \phi$, and

$$(G'_k, X'_i, G'_m, X'_j) = \begin{cases} (G_k, (X_i - X_j) \cup \{G_j\}, G_i, X_j) & (\text{if } G_k = G_m \text{ and } X_j \subsetneq X_i) \\ (G_j, X_i, G_k, (X_j - X_i) \cup \{G_i\}) & (\text{if } G_k = G_m \text{ and } X_i \subsetneq X_j) \\ (G_k, X_i, G_m, X_j) & (\text{otherwise}). \end{cases}$$

Then, $N[G'_m, G_j<X'_j>]N[G_k, G_i<X_i>](T) = N[G'_k, G_i<X'_i>]\ N[G_m, G_j<X_j>](T)$, iff

(a) $G_k = G_m$, $X_i \cap X_j = \phi$, and weak multivalued dependency $cc[G_k] - X_i - X_j \text{—(w)→→} X_i$ holds in T,

(b) $G_k = G_m$ and $X_i \subsetneq X_j$,

(c) $G_k = G_m$ and $X_j \subsetneq X_i$, or

(d) $G_k \neq G_m$. ▭

Proposition 4 assures that two NEST operations are commutative, if the new groups do not share the parent. Otherwise, a certain weak multivalued dependency is required to hold.

NEST and FLAT do not generally commute. An example is shown in Figure 3. In this case, $F[G_2]N[G_1, G_3<F_3, F_4>](T) \neq N[G_1, G_3<F_3, F_4>]F[G_2](T)$. The following proposition gives a sufficient condition for commutativity of NEST and FLAT.

Proposition 5 (Commutativity of NEST and FLAT) : Let G_k and G_m be distinct groups of an NT T, G_m be other than the root, $X \subsetneq cc(G_k)$, $X \neq \phi$, $G_m \notin \bigcup_{G \in cg(G_k)-X} dg+(G)$, and

$$X' = \begin{cases} (X - \{G_m\}) \cup cc(G_m) & (\text{if } G_m \in X) \\ X & (\text{otherwise}). \end{cases}$$

†4 $F[G_i]F[G_j](T)$ means $F[G_i](F[G_j](T))$.

Then, $F[G_m]N[G_k, G_i<X>](T) = N[G_k, G_i<X'>]F[G_m](T)$. ▭

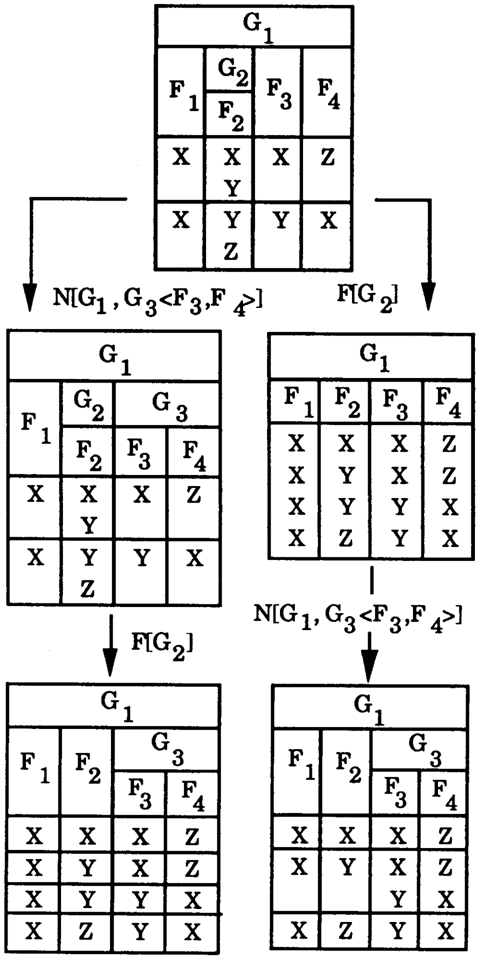

Figure 3. Incommutativity of NEST and FLAT

3. Nesting and Flattening

Sequences of two NEST and/or FLAT operations were discussed in Section 2. In this section, we consider more general sequences of NEST and/or FLAT to derive a complicated NT from a flat NT and vice versa.

Definition 5 : Given an NT T with n+1 groups, we define, as a *flattening* for T (denoted by F*), a sequence of n FLATs which transforms T into a flat NT. A flattening consisting only of outer FLATs is called an *outer flattening* and denoted by μ*. ▭

From Proposition 3, we obtain the following corollary.

Corollary 1 : Given an NT T, FT = F*(T) is same for any flattening F* for T. ▭

Given two NT schemas NS_1 and NS_2, if NS_1 and NS_2 are obtainable from each other with some sequences of NEST and/or FLAT, NS_1 and NS_2 are said to be *NF-translatable*.

Definition 6 : Given a flat NT FT and an NF-translatable NT schema NS with n+1 groups, we define, as a *nesting* for FT (denoted by N*), a sequence of n NESTs which transforms FT into an NT with the NT schema NS. A nesting consisting only of outer NESTs is called an *outer nesting* and denoted by ν*. ▭

From Proposition 4, we obtain for following corollary.

Corollary 2 : Given a flat NT FT and an NF-translatable NT schema NS, N_1*(FT) = N_2*(FT) does not always hold for different nestings N_1* and N_2* for FT. ▭

As stated in Corollary 2, we cannot arbitrarily change the order of NESTs in a nesting. However, any nesting has an equivalent outer nesting. We get the following proposition from Proposition 4.

Proposition 6 : Given a nesting N* for a flat NT FT, there exists an outer nesting ν* such that N*(FT) = ν*(FT). ▭

We can use a mixed sequence of NEST and FLAT as well as a nesting to derive a complicated NT from a flat NT.

Definition 7: Given a flat NT FT and an NF-translatable NT schema NS, we define, as a *general nesting* for FT (denoted by $\mathbb{N}$*), a mixed sequence of NESTs and FLATs which transforms FT into NT with the NT schema NS. A general nesting consisting only of outer NESTs and outer FLATs is called an *outer general nesting* and denoted by ω*. ▭

A general nesting does not always have an equivalent nesting. For example, the NT shown in Figure 1 can be derived with the general nesting $F[G]N[G_1, G_2<F_3, F_4>]N[G_1, G<F_1, F_2>]$ from the flat NT shown in Figure 4. However, it cannot be obtained with the only applicable nesting $N[G_1, G_2<F_3, F_4>]$. As in the case of nesting, any general nesting has an equivalent outer general nesting.

G_1			
F_1	F_2	F_3	F_4
X	Y	X	X
X	Y	X	Y
X	Y	Y	Y
X	Y	Y	X
Z	Y	Y	X

Figure 4. Flat Nested Table

Proposition 7: Given a general nesting N* for a flat NT FT, there exists an outer general nesting ω* such that N*(FT) = ω*(FT). ▭

This proposition can be derived from Propositions 1, 2, 3, 4, 5.

4. Classification of Nested Tables

Van Gucht and Fischer identified a number of interesting subclasses of nested relations under the shallowly nested NEST and FLAT [21]. Here, we consider similar classification under the deeply nested NEST and FLAT. Definitions of some of the subclasses discussed in the remaining part of the paper were given in our previous work [14, 15]. In this section, we discuss the Normalization Lossless Structure, the Nested Relation, and the Permutable Nested Relation. In Section 5, we consider the Hierarchical Structure.

Definition 8 : An NT T is a *Normalization Lossless Nested Table (NLNT),* if T = N*(F*(T)) for some flattening F* and general nesting N* [15]. ▭

As we previously mentioned, the NT shown in Figure 1 is an example of an NLNT. If we restrict F* and N* to outer flattening μ* and outer general nesting ω* in Definition 8, we get the definition of Normalization Lossless Structures given in [21]. Any Normalization Lossless Structure is an NLNT by the definition. By Corollary 1, any flattening has an equivalent outer flattening. By Proposition 7, any general nesting has an equivalent outer general nesting. Therefore, any NLNT is a Normalization Lossless Structure.

Proposition 8 : The class of Normalization Lossless Nested Tables (NLNTs) is equal to that of Normalization Lossless Structures. ▭

Definition 9 : An NT T is a *Canonical Nested Table (CNT),* if T = N*(F*(T)) for some flattening F* and nesting N* [14, 15]. ▭

By the definition, a CNT is always an NLNT. However, the converse does not hold, as exemplified by the NT shown in Figure 1. If we restrict F* and N* to outer flattening μ* and outer nesting ν* in Definition 9, we get the definition of Nested Relations.

Proposition 9 : The class of Canonical Nested Tables (CNTs) is equal to that of Nested Relations. ▭

Van Gucht and Fischer proposed an efficient algorithm to determine a given nested relation is a Nested Relation [21]. Proposition 9 assures that the same algorithm can be used to identify CNTs.

Definition 10 : An NT T is a Permutable Nested Table (PNT), if T = N*(F*(T)) for some flattening F* and any nesting N* such that T and N*(F*(T)) have an identical NT schema. ▭

By the definition, a PNT is always a CNT. However, the converse does not hold because NEST operations do not necessarily commute as discussed in Proposition 4. If we restrict F* and N* to outer flattening μ* and outer nesting ν* in Definition 10, we get the definition of Permutable Nested Relations. Any PNT is a Permutable Nested Relation by the definition and Corollary 1. The converse is proved from Corollary 1 and Proposition 7.

Proposition 10 : The class of Permutable Nested Tables (PNTs) is equal to that of Permutable Nested Relations. ▭

Van Gucht and Fischer also indicated an algorithm to identify Permutable Nested Relations [21]. Proposition 10 assures its applicability to PNTs.

5. Hierarchical Structure

In addition to the subclasses of nested relations mentioned above, Hierarchical Structures were discussed based on the "hierarchical nest" operation in [21]. Some nested relational models consider only Hierarchical Structures as data structures [1, 19]. To discuss Hierarchical Structures under the deeply nested algebra, we have to extend the definition of the hierarchical nest operation in [21].

Definition 11 : Given a group G_i, $F \subseteq cf(Gi)$, and $X \subseteq cc(G_i)$ $(X \neq \phi, F \cap X = \phi)$, the *HNEST* operation $H[G_i<F>, G_j<X>]$ creates as a child of G_i a new group G_j consisting of components X. Here, $G_j<X>$ is a new group schema. By the HNEST operation, every occurrence O of G_i is replaced by the following O':

$$O' = \{(t[cc(G_i) - X], FH(t)) \mid t \in O\},$$

where

$$FH(t) = \{u[X] \mid u \in O \wedge u[F] = t[F]\}.$$

In case G_i is the root, the HNEST is referred to as an *outer HNEST*. ▭

The outer HNEST is equivalent to the hierarchical nest operation introduced in [21]. Figure 5 shows an example of the HNEST operation. We consider sequences of HNESTs to derive a complicated NT from a flat NT.

Definition 12 : Given a flat NT FT and an NF-translatable NT schema NS, we define, as a *hierarchical nesting* for FT (denoted by H*), a sequence of HNESTs
(1) which transforms FT into an NT T with the NT schema NS, and
(2) each HNEST of which has the form $H[G_i<K>, G_j<X>]$ such that

$$K = \{cf(G_k) \mid G_k \in (dg(G_i) \cup \{G_i\}) \wedge G_k \in ag(G_j)\},$$

where functions cf, dg, and ag (defined in Section 2) are evaluated in the context of NS. A hierarchical nesting consisting only of outer HNESTs is called an *outer hierarchical nesting* and denoted by λ^*. ▭

G_1						
F_1	F_2	G_2			G_3	
		F_3	F_4	F_5	F_6	F_7
X	Y	X	Y	X	X	Y
		X	Z	Y		
		Z	X	Y		
Y	Z	X	Z	Y	Y	Z
		Z	X	Z	Z	X
		Z	Y	X		

$H[G_2<F_3>, G_4<F_4>]$

↓

G_1						
F_1	F_2	G_2			G_3	
		F_3	G_4	F_5	F_6	F_7
			F_4			
X	Y	X	Y Z	X	X	Y
		X	Y Z	Y		
		Z	X	Y		
Y	Z	X	Z	Y	Y	Z
		Z	X Y	Z	Z	X
		Z	X Y	X		

Figure 5. HNEST

Definition 13 : An NT T is a *Hierarchical Nested Table (HNT),* if T = H*(FT) for some hierarchical nesting H* for a flat NT FT[†5]. ▭

If we restrict H* to outer hierarchical nesting

†5 HNTs are referred to as Well-classified Nested Tables (WNTs) in [21].

λ^* in Definition 13, we get the definition of Hierarchical Structures. Figure 6 shows an example of a hierarchical nesting and the obtained HNT. To decide whether the class of HNTs is also equal to that of Hierarchical Structures, we consider commutativity of HNESTs.

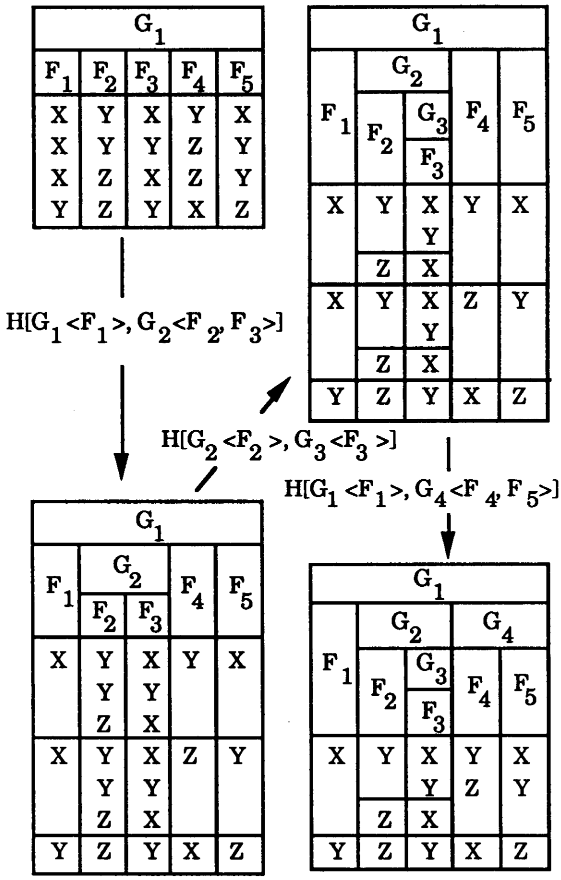

Figure 6. Hierarchical Nesting

Proposition 11 : Let G_k be a group of an NT T, $F \subseteq cf(G_k)$, $X_i \subseteq cc(G_k)$, $X_i \neq \phi$, $F \cap X_i = \phi$, $X_j \subseteq cc(G_k)$, $X_j \neq \phi$, $F \cap X_j = \phi$, and $X_j \cap X_j = \phi$. Then,

$$H[G_k<F>, G_j<X_j>]H[G_k<F>, G_i<X_i>](T)$$
$$= H[G_k<F>, G_i<X_i>]H[G_k<F>, G_j<X_j>](T)$$

▭

Proposition 12 : Let G_k be a group of an NT T, $F_i \subseteq cf(G_k)$, $X_i \subseteq cc(G_k)$, $X_i \neq \phi$, $F_i \cap X_i = \phi$, $F_j \subseteq cf(G_k)$, $F_j \subseteq X_i$, $X_j \subseteq X_i$, $X_j \neq \phi$, and $F_j \cap X_j = \phi$. Then,

$$H[G_i<F_j>, G_j<X_j>]H[G_k<F_i>, G_i<X_i>](T)$$
$$= H[G_k<F_i>, G_i<X_i>]H[G_k<F_iF_j>, G_j<X_j>](T)$$ ▭

Proposition 13 : Let G_k and G_m be distinct groups of an NT T, $F_i \subseteq cf(G_k)$, $X_i \subseteq cc(G_k)$, $X_i \neq \phi$, $F_i \cap X_i = \phi$, $F_j \subseteq cf(G_m)$, $X_j \subseteq cc(G_m)$, $X_j \neq \phi$, and $F_j \cap X_j = \phi$. Then,

$$H[G_m<F_j>, G_j<X_j>]H[G_k<F_i>, G_i<X_i>](T)$$
$$= H[G_k<F_i>, G_i<X_i>]H[G_m<F_j>, G_j<X_j>](T)$$ ▭

Proposition 14 : Given a hierarchical nesting H^* for a flat NT FT, there exists an outer hierarchical nesting λ^* such that $H^*(FT) = \lambda^*(FT)$. ▭

This proposition is derived from Propositions 12 and 13, in a similar way to Proposition 6. Any Hierarchical Structure is an HNT by the definition. The converse is proved from Proposition 14.

Proposition 15 : The class of Hierarchical Nested Tables (HNTs) is equal to the class of Hierarchical Structures. ▭

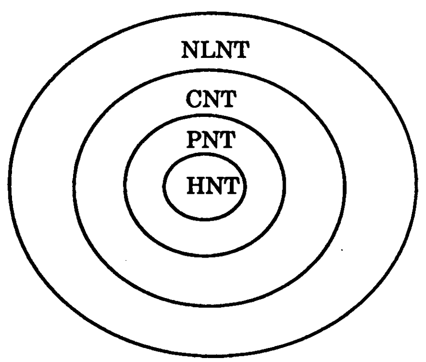

Figure 7. Subclasses of Nested Relations

It was proved in [21] that the class of Hierarchical Structures is properly contained in the class of Permutable Nested Relations. Therefore, from Propositions 10 and 15, we conclude that the class of HNTs is a proper subset of the class of PNTs. The inclusion relationship among the classes of NLNTs, CNTs, PNTs, and HNTs is shown in Figure 7.

6. Conclusion

A variety of algebraic operations have been proposed for nested relations. The set of operations proposed in [6] is one of well known instances of the shallowly nested algebra, in which operations are applicable only at the outermost level in nested relations. Most theoretical studies on nested relations have been based on the shallowly nested algebra because of its logical simplicity. However, data manipulation can be expressed more succinctly when algebraic operations are directly applicable to internal table structures without nesting and flattening. We refer to an nested algebra with this property as a deeply nested algebra. In this paper, we have investigated classification of nested relations under the deeply nested algebra, in particular the deeply nested NEST/FLAT.

Van Gucht and Fischer identified interesting subclasses of nested relations under the shallowly nested NEST and FLAT. They were the Normalization Lossless Structures, the Nested Relations, the Permutable Nested Relations, and the Hierarchical Structures. We have defined corresponding subclasses of nested relations based on the deeply nested NEST and FLAT in the nested table data model and on the extended hierarchical nest operation HNEST. They are named Normalization Lossless Nested Tables (NLNTs), Canonical Nested Tables (CNTs), Permutable Nested Tables (PNTs), and Hierarchical Nested Tables (HNTs). Then, we have proved that each of these subclasses is equal to its counterpart defined under the shallowly nested NEST and FLAT. The interpretation of each subclass based on the deeply nested NEST/FLAT is more intuitively understandable, while that based on the shallowly nested NEST/FLAT lends itself better to theoretical analysis. The above conclusion has been drawn from the study of reversibility and commutativity of the deeply nested NEST, FLAT, and HNEST, and some of their interesting properties have also been clarified in the paper. The research results presented here do not only contribute to taxonomy of nested relations but also to the in-depth analysis of data manipulation by the deeply nested algebra.

Acknowledgements

The authors are grateful to Prof. Isao Suzuki and Prof. Yuzuru Fujiwara of Institute of Information Sciences and Electronics, University of Tsukuba for their encouragement to our research.

References

[1] Abiteboul, S. and Bidoit, N., "Non First Normal Form Relations to Represent Hierarchically Organized Data," *Proc. ACM PODS,* March 1984, pp. 191-200.

[2] Abiteboul, S., Fischer, P. C., and Schek, H. J. (eds.), *Nested Relations and Complex Objects in Databases,,* LNCS 361, Springer-Verlag, 1989.

[3] Colby, L. S., "A Recursive Algebra and Query Optimization for Nested Relations," *Proc. ACM SIGMOD,* Portland, June 1989, pp. 273-283.

[4] Deshpande, V. and Larson, P. A., "Transforming from Flat Algebra to Nested Algebra, " *Proc. 23rd HICSS,* Vol. 2, Hawaii, January 1990, pp. 298-307.

[5] Fischer, P. C. and Thomas, S. J., "Operators for Non-First-Normal-Form Relations," *Proc. IEEE COMPSAC 83,* Chicago, November 1983, pp. 464-475.

[6] Fischer, P. C. and Van Gucht, D., "Determining When a Structure is a Nested Relation," *Proc. 12th VLDB,* Stockholm, Sweden, August 1985, pp. 171-180.

[7] Gyssens, M. and Van Gucht, D., "The Powerset Algebra as a Result of Adding Programming Constructs to the Nested Relational Algebra," *Proc. ACM SIGMOD,* Chicago, June 1988, pp. 225-232.

[8] Jaeschke, G. and Schek, H. J., "Remarks on the Algebra of Non First Normal Form Relations," *Proc. ACM PODS,* Los Angeles, March 1982, pp. 124-138.

[9] Jaeschke, G., "Nonrecursive Algebra for Relations with Relation Valued Attributes," Technical Report TR85-03-001, IBM, March 1985.

[10] Kitagawa, H. and Kunii, T. L., "APAD : Application Adaptable Database System - Its Architecture and Design -," *Proc. 13th IBM Computer Science Symposium,* October 1979, also in *Data Base Design Techniques II,* Yao, S. B. and Kunii, T. L. (eds.), LNCS 133, Springer-Verlag, Berlin, 1982, pp. 320-344.

[11] Kitagawa, H. and Kunii, T. L., "Form

Transformer - A Formalism for Office Form Manipulation -," *Proc.14th IBM Computer Science Symposium*, October 1980, also in *Operating System Engineering*, Maekawa, M. and Belady, L. A. (eds.), LNCS 143, Springer-Verlag, 1982, pp. 392-406. Revised version also appears in *Proc. 15th HICSS*, Hawaii, January 1982, pp. 132-141.

[12] Kitagawa, H., Gotoh, M., Misaki, S. and Azuma, M., "Form Document Management System SPECDOQ - Its Architecture and Implementation -," *Proc. 2nd ACM SIGOA Conf.*, Toronto, Canada, June 1984, pp. 132-142.

[13] Kitagawa, H., Kunii, T. L., Azuma, M. and Misaki, S., "Formgraphics: A Form-Based Graphics Architecture Providing a Database Workbench," *IEEE CGA*, Vol. 4, No. 6, June 1984, pp. 38-56.

[14] Kitagawa, H. and Kunii, T. L., *The Unnormalized Relational Data Model - For Office Form Processor Design*, Springer-Verlag, 1989.

[15] Kitagawa, H. and Kunii, T. L., "Nested Table Handling by Flat Table Operators," *Proc. 23rd HICSS*, Vol. 2, Hawaii, January 1990, pp. 288-297.

[16] Kitagawa, H. Kunii, T. L., and Ohbo, N., "Classification of Nested Tables under Deeply Nested Algebra," *Proc. 24th HICSS*, Vol. 2, Hawaii, January 1991, pp. 165-173.

[17] Makinouchi, A., "A Consideration on Normal Form of Not-Necessarily-Normalized Relation in the Relational Data Model," *Proc. 3rd VLDB*, Tokyo, Japan, October 1977, pp. 447-453.

[18] Ozsoyoglu, Z. M. (ed.), Special Issue on Nested Relations, *IEEE Data Engineering,* Vol. 11, No. 3, September 1988.

[19] Roth, M. A., Korth, H. F. and Silberschatz, A., "Extended Algebra and Calculus for Nested Relational Databases," *ACM TODS*, Vol. 13, No. 4, December 1988, pp. 389-417.

[20] Schek, H. J. and Scholl, M. H., "The Relational Model with Relation-Valued Attributes," *Information Systems*, Vol. 11, No. 2, 1986, pp, 137-147.

[21] Van Gucht, D. and Fischer, P. C., "Some Classes of Multilevel Relational Structures," *Proc. 5th ACM PODS*, 1986, pp. 60-69.