Algebraic Optimization for Nested Relations

Yiming Jan
Computer Science Department
Indiana University, Bloomington, IN 47405-4101, U.S.A.

ABSTRACT

This paper gives a more comprehensive understanding of the commutative properties of nested relational operators in the context of optimization. Since there already exist some positive and negative results of commutative properties, the information obtained in this paper will be useful for knowing more algebraic information to optimize nested relational models. By introducing six mutual dependencies between nested relations and other conditions, an algebraic characterization of nested relational operators is given in terms of constraint satisfaction. The results contrast with the algebraic properties of relational operators.

I. INTRODUCTION

Unlike earlier systems that allow only data in atomic values, most database systems studied in the 1980's are able to treat sets as basic values [2,5,6,8,12,18,19,21,24,26]. In addition to increasing the power of relational databases, such extensions can support wider applications. Meanwhile, some systems keep query languages and conceptual views of data as concise as in the relational model. To see why the extended models have more capacities, the papers [1,4,9,14,15,19,27] investigate several models from the expressiveness viewpoint. However, from the querying perspective, query forms in the extended data models are quite different from relational ones. The question of how to query and optimize models for more complex objects is still open in the database community. Since query optimization is quite successful in relational database systems, those designing a nested relational database are looking for good ways to optimize operations by borrowing techniques developed for the relational model [7,17,20,21,24,25]. For example, in relational algebra, pushing selections and projections as far as possible into the parse tree can save a lot of temporary storage and intermediate computation time. The commutativity of selection and projection with other binary operators is the basic characteristic for adopting such strategies. However, in extended databases, applying the same rule becomes a problem. In most algebras involving sets and restructuring, commutativity among operators does not always hold. For example, it is still unknown why commutativity is so hard in the context of nested relations.

There are reasons to consider nested relational model in this paper among various data models. First, the algebras associated with nested relations consist of few operators extending the traditional relational algebra. Secondly, by dropping the first normal form assumption, the data types allowed in this model are conceptually simple extensions of relational data types. The model also provides constructors for data aggregations and groupings. These are important features in extended data models. As noted by [17], nested relational databases are related to object-oriented databases from the perspective of relational model extensions. In terms of object-oriented databases, a nested relation is used for a class and a tuple of the relation is a member of the class. This concept is a simple database interpretation from the set construction of database schemas and instances. Based on these observations, we can view the nested relational model as a bridge between the relational model and other extended data models even though the mappings are not necessarily straightforward. Therefore, to study the optimization of extended data models, algebraic properties of nested relational operators are both theorectically and practically important.

This paper will review previous results concerning the algebraic properties of nested relational operators [26,27]. We will then study more algebraic equations in the sense of Thomas and Fischer [26]. In their study, they pointed out problems for nested relational equations. By introducing dependencies, we will give necessary and sufficient conditions for those equations to hold. In other works [14,24], extensions of the nested relational algebra have been studied. One such example is the powerset algebra. It turns out that similar problems related to algebraic optimization also happen for this algebra. For the purpose of this paper, we only consider the nested relational algebra. Our results in this paper will characterize the commutativity properties of nested relational models. This research will also show that, for nested relational algebras, query optimization is different from the relational model. This solves an open problem in the database community [22]. At the end, we suggest some possible ways to optimize the nested relational models.

0073-1129/90/0000/0278$01.00

II. NESTED RELATIONAL ALGEBRA

From 1970's, the traditional way of representing data formally is to treat the data with only atomic values in the entries of tables [10,11]. However, this restriction seems to be too rigid for many applications. In the nested relational model, data has been generalized to allow relations in turn. This generalization has broadened the knowledge of the traditional model [13,14,15,16,23,24,26,27]. With this simple extension, the properties of this model turn out to be characteristically different from the traditional relational model. To describe this model, there are many different formalism. The nested relational algebra considered in this paper is generated by the relational operators defined for nested relations plus two additional operators, nesting (ν) and unnesting (μ). To define schema and nested relations, assume that the elementary and composed attributes are in the countable set U and the elementary values are in the countable set V. Now, we can define the schema, instance of a schema and nested relations as follows :

Definition 2.1
The schema, Ω, is recursively defined as follows :

- if $\{A_1, \ldots, A_k\} \subset U$ then $\Omega \equiv (A_1 \ldots A_k)$ is a schema ;
- if $\{A_1, \ldots, A_k\} \subset U$ and $\Omega_1, \ldots, \Omega_n$ are all schemas then $\Omega \equiv (A_1 \ldots A_k\ \Omega_1 \ldots\ \Omega_n)$ or $\Omega \equiv (\Omega_1\ \ldots\ \Omega_n)$ is a schema.

$A_1, A_2, A_3, \ldots$ are atomic attributes and $\Omega, \Omega_1, \Omega_2, \Omega_3, \ldots$ are composed attributes.

Definition 2.2
Let ω_Ω denote an instance over schema Ω and let I_Ω denote the family of all instances ω_Ω. For each atomic attribute A in U, there is an associated set of values $DOM(A)$. The symbol $\mathcal{P}$ denotes the powerset used in the set theory. Then ω_Ω and I_Ω are recursively defined as follows :

- if $\Omega \equiv (A_1 \ldots A_k)$ then $\omega_\Omega \subset DOM(A_1) \times \ldots \times DOM(A_k)$ and $I_\Omega = \mathcal{P}(DOM(A_1) \times \ldots \times DOM(A_k))$;
- if $\Omega \equiv (A_1 \ldots A_k\ \Omega_1\ \ldots\ \Omega_n)$ then $\omega_\Omega \subset DOM(A_1) \times \ldots \times DOM(A_k) \times I_{\Omega_1} \times \ldots \times I_{\Omega_n}$ and $I_\Omega = \mathcal{P}(DOM(A_1) \times \ldots \times DOM(A_k) \times I_{\Omega_1} \times \ldots \times I_{\Omega_n})$.

The symbol ω will be used instead of ω_Ω whenever there is no ambiguity. The notation t_Ω is to denote the tuple which is restricted to be a member of ω_Ω.

We are now able to define nested relations.

Definition 2.3
A nested relation is a pair (Ω, ω) with $\omega \in I_\Omega$. If $\{A_1, \ldots, A_k\} \subset U$ and $\Omega \equiv (A_1 \ldots A_k)$, then (Ω, ω) is called a flat relation.

Consider an example of the nested relations.

Example 2.1
The following table represents a piece of information about an architect's database. The relationships among roomtype, room's number and room's area are of major concern.

(*ROOMTYPE*	*ROOM-NO*	*AREA*)
Classroom	002	530
Classroom	003	541
Classroom	004	446

Alternatively, these data can be represented in a nested relation :

(*ROOMTYPE*	(*ROOM-NO*	*AREA*))
Reception Room		
Classroom	002	530
	003	541
	004	446

The second tuple shows a roomtype having no fixed information about the room's number and room's area. This facilitates the representation of incomplete information [7].

After all these basic concepts of the nested relational model are defined, we can then formulate the nested relational algebra [13,14] :

Definition 2.4
Other than nesting and unnesting, the nested relational algebra shares the same definitions with the relational algebra. However, it is defined on the nested relations. Now, to complete our descriptions, suppose $\{A_h, \ldots, A_k\} = \{A_1, \ldots, A_k\} - \{A_1, \ldots, A_{h-1}\}$ and $\{\Omega_1, \ldots, \Omega_n\} = \{\Omega_1, \ldots, \Omega_s\} \cup$

$\{\Omega_{s+1},\ldots,\Omega_n\}$. Let $\Omega \equiv (A_1\ldots A_k(\Omega_1\ldots\Omega_n))$ and $\Omega' \equiv (A_1\ldots A_{h-1}\Omega_1\ldots\Omega_s(A_h\ldots A_k\Omega_{s+1}\ldots\Omega_n))$. Consider two nested relations $(\Omega,\omega),(\Omega',\omega')$ with $T \equiv (A_h\ldots A_k\ \Omega_{s+1}\ldots\Omega_n)$ and $S \equiv (A_1\ldots A_{h-1}\ \Omega_1\ldots\Omega_s)$. The nesting operator ν and the unnesting operator μ are defined as follows :

- The nesting $\nu_T(\Omega,\omega) = (\Omega',\omega')$ and $\omega' = \{t_{\Omega'} \mid \exists t' \in \omega\ , t_S = t'_S, t[T] = \{t''_T \mid t'' \in \omega, t'_S = t''_S\}\}$.
- The unnesting $\mu_T(\Omega',\omega') = (\Omega,\omega)$ and $\omega = \{t_\Omega \mid \exists t' \in \omega',\ \ t_S = t'_S, t_T \in t'[T]\}$.

Nested algebraic expressions, NAE's, can now be recursively defined as :

- $x, y, z, \ldots$ are NAE's;
- For every Ω, $(\Omega,\emptyset)$ is a NAE;
- For every Ω, $((\Omega),\{\emptyset\})$ is a NAE;
- The expressions formed by applying the nested relational operators and the operators defined above to NAE's are NAE's.

Example 2.2
Reconsider Example 2.1. If we denote the flat relation in this example by (Ω,ω), $\nu_{(ROOM\text{-}NO\ AREA)}$ (Ω,ω) yields the nested relation:

(ROOMTYPE	(ROOM-NO	AREA))
Classroom	002	530
	003	541
	004	446

Clearly, this nesting can be undone by the corresponding unnesting. However, in general, the opposite is not true. In particular, tuples with "empty" values are lost on unnesting.

For this paper's purpose, we take all the elementary attributes in a composed attribute to be different. Based on this naming convention, we will give the following definition for a sequence of operators :

Definition 2.5
Let (Ω,ω) be a relation and Ω' is the list of all elementary attributes in Ω. $\mu^*(\Omega,\omega)$ is the relation derived by a repeated unnesting of (Ω,ω). We call such a sequence as an unnesting sequence.

Example 2.3
Unnesting the second relation in Example 2.1 over $(ROOM\text{-}NO\ AREA)$ gives the first relation in Example 2.1.

Note that we can also define a nesting and unnesting sequence in a similar fashion. In this paper, however, the above formalism is already sufficient.

III. PREVIOUS WORKS

Since the introduction of nested relational models, there have been some results related to the algebraic properties of nested operators. Thomas and Fischer [26] studied the commutative properties among the nest, unnest and relational operators. In their paper, the following operators are of main concern : $\cap$, $-$, $\cup$, $\times$, $\bowtie$, σ, π, ν and μ.

Their paper [26] discusses the following six equations. The idea is to understand interactions of nested relational operators. Their study followed the traditional relational approach of trying to exchange binary operators with unary ones. For most cases of this paper, without loss of generality, we will consider nested relations with the schema $(X\ Y)$ where X, Y are lists of elements of U. The symbol θ denotes a binary operator and r, r_1, r_2 are relations.

1. $\nu_{(X)}(r_1\ \theta\ r_2) = \nu_{(X)}(r_1)\ \theta\ \nu_{(X)}(r_2)$;
2. $\mu_{(X)}(r_1\ \theta\ r_2) = \mu_{(X)}(r_1)\ \theta\ \mu_{(X)}(r_2)$;
3. $\mu_{(X)}(\nu_{(X)}(r_1)\ \theta\ \nu_{(X)}(r_2)) = r_1\ \theta\ r_2$;
4. $\nu_{(X)}(\mu_{(X)}(r)) = r$;
5. $\nu_{(X)}(\nu_{(Y)}(r)) = \nu_{(Y)}(\nu_{(X)}(r))$;
6. $\mu^*(r_1\ \theta\ r_2) = \mu^*(r_1)\ \theta\ \mu^*(r_2)$.

Note that (1) implies (3) by "multiplying" $\mu_{(X)}$ on both sides. For (1) and (2), the equations do not always hold. To see whether algebraic information can be used to optimize the nested relational models, it is sufficient to check if these equations hold. Before studying more details of these equations, we give a brief review of what is already known.

Most of the interactions among the operators to nested relation(s) have been investigated in the papers [24,26]. Basically, commutative properties are of most concern since these are useful in optimizing relational databases. The same reasoning follows in the nested relational model to eliminate cost-consuming nesting and unnesting operations. Some notations are used in this section. The symbol $\Rightarrow$ stands for a nested functional dependency, $\rightarrow\rightarrow$ for MVD and $\overset{weak}{\rightarrow\rightarrow}$ for weak MVD. P is used as a selection predicate. P' is derived from P by changing its attribute and associated values in the context understood. Given relations r, r_1 and r_2, the equations listed below are known to hold in the nested relational model.

- $\mu_{(X)}(\nu_{(X)}(r)) = r$;
- $\mu_{(X)}(\mu_{(Y)}(r)) = \mu_{(Y)}(\mu_{(X)}(r))$;
- $\nu_{(X)}(\mu_{(X)}(r)) = r$ if and only if $\Omega = (Y\ X)$ and $Y \Rightarrow (X)$;
- $\nu_{(X)}(\nu_{(Y)}(r)) = \nu_{(Y)}(\nu_{(X)}(r))$ if and only if $X \overset{\mathbf{weak}}{\rightrightarrows} Y$;
- $\nu_{(X)}(\sigma_P(r)) = \sigma_{P'}(\nu_{(X)}(r))$;
- $\mu_{(X)}(\sigma_P(r)) = \sigma_{P'}(\mu_{(X)}(r))$;
- $\mu_{(X)}(\sigma_P(\nu_{(X)}(r))) = \sigma_{P'}(r)$;
- $\mu^*(\sigma_P(r)) = \sigma_{P'}(\mu^*(r))$;
- $\mu_{(X)}(\pi_{(Y\ (X))}(r)) = \pi_{(Y\ X)}(\mu_{(X)}(r))$;
- $\mu_{(X)}(\pi_{(Y\ (X))}(\nu_{(X)}(r))) = \pi_{(Y\ X)}(r)$;
- $\pi_{(X)}(r) = \pi_{(X)}(r')$ where $r' = \nu_{(Y)}(r)$ and X, Y have no common attributes;
- $\pi_{(X)}(r) = \pi_{(X)}(r')$ where $r' = \mu_{(Y)}(r)$ and X, Y have no common attributes;
- $\mu_{(X)}(r_1 \cup r_2) = \mu_{(X)}(r_1) \cup \mu_{(X)}(r_2)$.

The paper [24] defines a partition normal form (PNF) for the nested relations. A nested relation r in PNF will satisfy :

- $\nu_{(X)}(\mu_{(X)}(r)) = r$;
- $\nu_{(Y_1)}(\cdots\nu_{(Y_n)}(\mu_{(Y_n)}\cdots(\mu_{(Y_1)}(r))\ldots)\ldots) = r$.

This information will provide some algebraic clues for an underlying architecture to explore the various ways to optimize user queries. But it is not yet fully understood in general when equations 1 and 2 will be true. The next section explains how to resolve such questions.

IV. ALGEBRAIC DEPENDENCIES

To resolve the commutative conditions of the equations concerned, we introduce the new definitions of dependencies between two nested structures. In this section, six types of mutual dependencies are newly defined. Generally, these relationships are the generalization of flat relationships related to the operators : intersection, difference, union, join, selection and projection. The organization of proofs is outlined as follows. First, the definition of the data dependency is introduced, if there is one needed to establish the equation. The studied equation is then given before its related lemma is presented. Four groups of lemmas are provided. The first two concern with the interactions of nest (unnest) operators with intersection, difference, union and join (difference, intersection and join) operators. Simple selection and projection with nest are treated as the last two lemmas of commutative properties. For notational conveniences, LHS refers to the final instance of the left-hand side of the equation mentioned most recently and RHS refers to the righ-hand side. The set of equations related to nesting and binary operators is discussed first. As mentioned earlier, we need to define new mutual dependencies to establish nested relational equations.

Definition 4.1

Let $r_1 = (\Omega, \omega_1)$ and $r_2 = (\Omega, \omega_2)$ be two relations such that $\Omega = (Y\ X)$ and $X_y(\omega) = \{t[X] \mid \exists t \in \omega,\ t[Y] = y\}$ where $r = (\Omega, \omega)$ is a relation. A ***mutual dependency md1*** between r_1 and r_2, $r_1 \overset{md1}{\underset{X}{\Longleftrightarrow}} r_2$, holds $\equiv \forall t_1 \in \omega_1\ \ \forall t_2 \in \omega_2,\ \ t_1[Y] = t_2[Y] \Longrightarrow (X_{t_1[Y]}(\omega_1) = X_{t_2[Y]}(\omega_2)) \vee (X_{t_1[Y]}(\omega_1) \cap X_{t_2[Y]}(\omega_2) = \emptyset)$.

By fixing a Y-value, for each relation we can collect all the X-values into a block. A more intuitive interpretation to the first dependency is that, once there is a common Y-value in both relations, the corresponding X-valued blocks are either identical or mutual exclusive. In the semantics of nested relational algebra, the X-valued blocks are naturally obtained by applying $\nu_{(X)}$ to the nested relation. The dependency introduced next has a more restricted condition. It requires that the X-valued blocks for a specific Y-value in both relations must be entirely identical.

Definition 4.2

Let r_1 and r_2 be defined as in definition 4.1. $r_1 \overset{md2}{\underset{X}{\Longleftrightarrow}} r_2 \equiv \forall t_1 \in \omega_1\ \forall t_2 \in \omega_2,\quad t_1[Y] = t_2[Y] \Longrightarrow X_{t_1[Y]}(\omega_1) = X_{t_2[Y]}(\omega_2)$.

Example 4.1

Consider two simple room relations, (Ω_1, ω_1) and (Ω_2, ω_2). They all have the same scheme (*ROOM-TYPE ROOM-NO AREA*) and can be represented as:

(*ROOMTYPE*	*ROOM-NO*	*AREA*)
Classroom	002	530
Classroom	003	541
Classroom	004	446
Hallway	0099	108
Hallway	0099A	214
Hallway	0099B	504
Hallway	0099C	549

(*ROOMTYPE*	*ROOM-NO*	*AREA*)
Classroom	002	530
Classroom	003	541
Classroom	004	446
Carrels	0001A	144
Carrels	0001B	209
Hallway	0099D	450
Hallway	0099E	639

Let $Y = ROOMTYPE$ and $X = ROOM-NO\ AREA$. Then, by definitions these two relations satisfy the mutual dependency *md1*. But they violate *md2*.

Based on these two mutual dependencies, we will establish the following three lemmas. They are related to the equations for nesting, intersection, difference and union.

Lemma 4.1

$\nu_{(X)}(r_1 \cap r_2) = \nu_{(X)}(r_1) \cap \nu_{(X)}(r_2) \iff r_1 \overset{md1}{\underset{X}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md1}{\underset{X}{\Longleftrightarrow}} r_2$ does not hold. That is, there are two tuples $\langle y\ x_3\rangle, \langle y\ x_4\rangle$ such that $\langle y\ x_3\rangle$ is in both ω_1 and ω_2 and $\langle y\ x_4\rangle$ is exclusively in either ω_1 or ω_2. On LHS, $\langle y\ x_3\rangle \in \omega_1 \cap \omega_2$ and, after nesting on X, there will be a tuple t such that $t[Y] = y$ and $t[(X)] \ni x_3$. On RHS, since $X_y(\omega_1) \neq X_y(\omega_2)$, there will be no tuple with Y component containing the value y after nesting on X. Thus, $LHS \neq RHS$.

$\Leftarrow$: Suppose $r_1 \overset{md1}{\underset{X}{\Longleftrightarrow}} r_2$. By definitions of relational operations and tuple calculus,

$$\begin{aligned}
LHS = \{t \mid\ & \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ \ t_1 = t_2, t[Y] = t_1[Y] = t_2[Y], t[(X)] = \{t_3[X] \mid \exists t_3 \in \omega_1\ \exists t_4 \in \omega_2,\ t_3 = t_4,\ t_3[Y] = t_4[Y] = t[Y]\}\} \\
= \{t \mid\ & \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ \ t[Y] = t_1[Y] = t_2[Y], t[(X)] = X_{t_1[Y]}(\omega_1) = X_{t_2[Y]}(\omega_2)\}. (By\ md1) \\
RHS = \{t \mid\ & \exists t_1 \in \omega_1,\ t[Y] = t_1[Y], t[(X)] = \{t_3[X] \mid \exists t_3 \in \omega_1,\ t_1[Y] = t_3[Y]\}\} \cap \\
\{t \mid\ & \exists t_2 \in \omega_2,\ t[Y] = t_2[Y], t[(X)] = \{t_4[X] \mid \exists t_4 \in \omega_2,\ t_2[Y] = t_4[Y]\}\} \\
= \{t \mid\ & \exists t_1 \in \omega_1\ \ \exists t_2 \in \omega_2,\ t[Y] = t_1[Y] = t_2[Y], t[(X)] = X_{t_1[Y]}(\omega_1) = X_{t_2[Y]}(\omega_2)\}.\ (By\ md1)
\end{aligned}$$

Thus $LHS = RHS$.

Lemma 4.2

$\nu_{(X)}(r_1 - r_2) = \nu_{(X)}(r_1) - \nu_{(X)}(r_2) \iff r_1 \overset{md1}{\underset{X}{\Longleftrightarrow}} r_2$.

Proof:

By the definitions of $-$, ν and the constraint *md1*.

Lemma 4.3

$\nu_{(X)}(r_1 \cup r_2) = \nu_{(X)}(r_1) \cup \nu_{(X)}(r_2) \iff r_1 \overset{md2}{\underset{X}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md2}{\underset{X}{\Longleftrightarrow}} r_2$ were false. On LHS, there is only one tuple with a Y-value y and a X-value $X_y(\omega_1) \cup X_y(\omega_2)$ where $X_y(\omega_1) \neq X_y(\omega_2)$ by assumption. On RHS, after the union operation, the resulting relation will have two tuples $\langle y\ X_y(\omega_1)\rangle$ and $\langle y\ X_y(\omega_2)\rangle$ since $X_y(\omega_1) \neq X_y(\omega_2)$. Thus, $LHS \neq RHS$.

$\Leftarrow$: Suppose $r_1 \overset{md2}{\underset{X}{\Longleftrightarrow}} r_2$.

$$\begin{aligned}
LHS = \{t \mid\ & \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2, t[Y] = t_1[Y] = t_2[Y], t[(X)] = \{t_3[X] \mid \exists t_3 \in \omega_1, t_3[Y] = t_1[Y]\} \cup \{t_4[X] \mid \exists t_4 \in \omega_2, t_4[Y] = t_2[Y]\}\}\ \cup \\
\{t \mid\ & \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2, t[Y] = t_1[Y] \neq t_2[Y], t[(X)] = \{t_3[X] \mid \exists t_3 \in \omega_1,\ t_3[Y] = t_1[Y]\}\}\ \cup \\
\{t \mid\ & \exists t_2 \in \omega_2\ \ \forall t_1 \in \omega_1,\ t[Y] = t_2[Y] \neq t_1[Y], t[(X)] = \{t_4[X] \mid \exists t_4 \in \omega_2,\ t_4[Y] = t_2[Y]\}\} \\
= \{t \mid\ & \exists t_1 \in \omega_1 \exists t_2 \in \omega_2, t[Y] = t_1[Y] = t_2[Y], t[(X)] = X_{t_1[Y]}(\omega_1) = X_{t_2[Y]}(\omega_2)\} \cup \\
\{t \mid\ & \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t[Y] = t_1[Y] \neq t_2[Y], t[(X)] = X_{t_1[Y]}(\omega_1)\}\ \cup \\
\{t \mid\ & \exists t_2 \in \omega_2\ \forall t_1 \in \omega_1,\ t[Y] = t_2[Y] \neq t_1[Y], t[(X)] = X_{t_2[Y]}(\omega_2)\}. (By\ md2)
\end{aligned}$$

Again, by *md2*, $RHS = LHS$ naturally follows.

Since join involves two relations with different schemas, the next defined mutual dependency will constrain the relation values of common attributes. It has a similar restriction as *md1*.

Definition 4.3

Let $r_1 = (\Omega_1, \omega_1)$ and $r_2 = (\Omega_2, \omega_2)$ be two relations such that $\Omega_1 = (X\ Y), \Omega_2 = (X\ Z)$, Y and Z have no common attributes. $r_1 \overset{md3}{\underset{X}{\Longleftrightarrow}} r_2 \equiv \forall t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ \ t_1[X] = t_2[X] \Rightarrow (X_{t_1[Y]}(\omega_1) = X_{t_2[Z]}(\omega_2)) \vee (X_{t_1[Y]}(\omega_1) \cap X_{t_2[Z]}(\omega_2) = \emptyset)$.

Lemma 4.4

$\nu_{(X)}(r_1 \bowtie r_2) = \nu_{(X)}(r_1) \bowtie \nu_{(X)}(r_2) \iff r_1 \overset{md3}{\underset{X}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md3}{\underset{X}{\Longleftrightarrow}} r_2$ does not hold. There are tuples $\langle y\ x_1 \rangle \in \omega_1, \langle x_1\ z \rangle \in \omega_2$ and either $\langle y\ x_2 \rangle \in \omega_1$ but $\langle x_2\ z \rangle \notin \omega_2$ or $\langle y\, x_2 \rangle \notin \omega_1$ but $\langle x_2\ z \rangle \in \omega_2$. On LHS, $\langle y\ x_1\ z \rangle \in \omega_1 \bowtie \omega_2$ and after nesting on X, there is a tuple $t \in LHS$ such that $t[YZ] = yz$ and $x_1 \in t[(X)]$. On RHS, $\nu_{(X)}(r_1)$ has the tuple t_1 such that $t_1[Y] = y$ and $t_1[(X)] = X_y(\omega_1)$ whereas $\nu_{(X)}(r_2)$ has $t_2[Z] = z$ and $t_2[(X)] = X_z(\omega_2)$. By $(X_y(\omega_1) \cap X_z(\omega_2) \neq \emptyset)$ and $(X_y(\omega_1) \neq X_z(\omega_2))$, the resulting relation would not have a tuple t such that $t[YZ] = yz$ and $\dot{x}_1 \in t[(X)]$. Thus, $LHS \neq RHS$.

$\Leftarrow$: Suppose $r_1 \overset{md3}{\underset{X}{\Longleftrightarrow}} r_2$.

$$\begin{aligned}
LHS = \{t \mid & \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2, t_1[X] = t_2[X], t[Y] \\
& = t_1[Y], t[Z] = t_2[Z], t[(X)] = \{t_3[X] \mid \\
& \exists t_3 \in \omega_1\ \exists t_4 \in \omega_2,\ t[Y] = t_3[Y], t[Z] \\
& = t_4[Z], t_3[X] = t_4[X]\}\} \\
= \{t \mid & \exists t_1 \in \omega_1 \exists t_2 \in \omega_2, t[Y] = t_1[Y], t[Z] = \\
& t_2[Z], t[(X)] = X_{t_1[Y]}(\omega_1) = X_{t_2[Z]}(\omega_2)\}. \\
& (By\ md3) \\
RHS = \{t \mid & \exists t_1 \in \omega_1,\ t[Y] = t_1[Y], t[(X)] = X_{t_1[Y]} \\
& (\omega_1)\} \bowtie \{t \mid \exists t_2 \in \omega_2,\ t[Z] = t_2[Z], \\
& t[(X)] = X_{t_2[Z]}(\omega_2)\} \\
= \{t \mid & \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ t[Y] = t_1[Y], \\
& t[Z] = t_2[Z], t[(X)] = X_{t_1[Y]}(\omega_1) = \\
& X_{t_2[Z]}(\omega_2)\}. (By\ \ definition)
\end{aligned}$$

Thus, $LHS = RHS$.

Before proceeding to prove other lemmas related to unnesting , some comments are necessary. First, when the unnest operator is considered, the multiset values of a nested attribute will make some equivalence harder to capture. A constraint like $X_y(\omega_1) \cap X_y(\omega_2) \neq \emptyset$ could not express the notion of set inclusion, multiset and the range of set elements. Because of this, we propose new constraints of nested structures to catch this semantics of data.

Definition 4.4

Let $r_1 = (\Omega, \omega_1)$ and $r_2 = (\Omega, \omega_2)$ where $\Omega = (Y\ (X))$. $r_1 \overset{md4}{\underset{(X)}{\Longleftrightarrow}} r_2 \equiv \forall t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t_1[Y] = t_2[Y] \Rightarrow (t_1[(X)] = t_2[(X)]) \vee (t_1[(X)] \cap t_2[(X)] = \emptyset) \vee (\exists t_3 \in \omega_2,\ (t_1[Y] = t_3[Y]) \wedge (t_1[(X)] = t_3[(X)]))$.

Lemma 4.5

$\mu_{(X)}(r_1 - r_2) = \mu_{(X)}(r_1) - \mu_{(X)}(r_2) \iff r_1 \overset{md4}{\underset{(X)}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md4}{\underset{(X)}{\Longleftrightarrow}} r_2$ were not true. Consequently, there exist tuples $\langle y\ x_1 \rangle \in \omega_1$ and $\langle y\ x_2 \rangle \in \omega_2$ where $\exists x \in x_1 \cap x_2$. The tuple $\langle y\ x_1 \rangle$ is not equal to any tuple in ω_2, thus $\langle y\ x_1 \rangle \in \omega_1 - \omega_2$ on LHS. After unnesting, $\langle y\ x \rangle \in LHS$. On RHS, $\langle y\ x \rangle$ is in both $\mu_{(X)}(r_1)$ and $\mu_{(X)}(r_2)$. After the difference operation, $\langle y\ x \rangle \notin RHS$. Thus, $LHS \neq RHS$.

$\Leftarrow$: Suppose $r_1 \overset{md4}{\underset{(X)}{\Longleftrightarrow}} r_2$.

$$\begin{aligned}
LHS = \{t \mid & \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t[Y] = t_1[Y] \neq \\
& t_2[Y], t[X] \in t_1[(X)]\}\ \cup \\
& \{t \mid \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t[Y] = t_1[Y] = \\
& t_2[Y], t_1[(X)] \cap t_2[(X)] = \emptyset, t[X] \\
& \in t_1[(X)]\}. \quad (By\ \ md4) \\
RHS = \{t \mid & \exists t_1 \in \omega_1, t[Y] = t_1[Y], t[X] \in t_1[(X)]\} \\
& - \{t \mid \exists t_2 \in \omega_2,\ t[Y] = t_2[Y], t[X] \in t_2[(X)]\} \\
& = \{t \mid \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t[Y] = t_1[Y] \neq \\
& t_2[Y], t[X] \in t_1[(X)]\}\ \cup \\
& \{t \mid \exists t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t[Y] = t_1[Y] = \\
& t_2[Y], t_1[(X)] \cap t_2[(X)] = \emptyset, t[X] \in \\
& t_1[(X)]\}. (By\ \ md4)
\end{aligned}$$

Thus, $LHS = RHS$.

As we shall see later, the last two defined mutual dependencies have more complicated restrictions to the relation instances. That implies the unnesting operator has negative effects on the intersection and join operators. To preserve the identity related to intersection and unnest, we introduce a fifth mutual dependency.

Definition 4.5

Let r_1, r_2 be defined as earlier and $(X)_y(\omega_1, \omega_2) = \{x \mid \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ t_1 = t_2, t_1[Y] = y, x \in t_1[(X)]\}$. $r_1 \overset{md5}{\underset{(X)}{\Longleftrightarrow}} r_2 \equiv \forall t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ t_1[Y] = t_2[Y] \Rightarrow (t_1[(X)] - (X)_{t_1[Y]}(\omega_1, \omega_2)) \cap t_2[(X)] = \emptyset$.

Lemma 4.6

$\mu_{(X)}(r_1 \cap r_2) = \mu_{(X)}(r_1) \cap \mu_{(X)}(r_2) \iff r_1 \overset{md5}{\underset{(X)}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md5}{\underset{(X)}{\Longleftrightarrow}} r_2$ were false. The set $(X)_{t_1[Y]}(\omega_1, \omega_2)$ contains all the elements of (X) derived from the equal tuples of ω_1 and ω_2 sharing the same Y value $t_1[Y]$. By assumption, there exist two tuples $\langle y\ x_1 \rangle \in \omega_1, \langle y\ x_2 \rangle \in \omega_2$ such that $x \in (x_1 \cap x_2)$ and x is not in the $(X)_y(\omega_1, \omega_2)$. On LHS, $\langle y\ x \rangle$

will not "survive" through the intersection since the (X) components of the intersected tuples in ω_1 and ω_2 would not contain x by $x \notin (X)_y(\omega_1, \omega_2)$. On RHS, $\langle y\ x \rangle$ is in both $\mu_{(X)}(r_1)$ and $\mu_{(X)}(r_2)$. That means $\langle y\ x \rangle \in RHS$ by the definition of intersection. Thus, $LHS \neq RHS$.
$\Leftarrow$: Suppose $r_1 \overset{md5}{\underset{(X)}{\Longleftrightarrow}} r_2$. There are basically two possibilities to consider under the constraint $md5$ with respect to Y value of ω_1.
(Case 1) $\exists t_1 = \langle y\ x_1 \rangle \in \omega_1\ \forall t_2 = \langle y\ x_2 \rangle \in \omega_2,\ (x_1 - (X)_y(\omega_1, \omega_2)) \cap x_2 = \emptyset$.
Depending on the result of $x_1 - (X)_y(\omega_1, \omega_2)$, there are two cases :
(Case 1.1): $x_1 - (X)_y(\omega_1, \omega_2) = \emptyset \Rightarrow x_1 \subseteq (X)_y(\omega_1, \omega_2)$. That is, x_1 is a subset of the set composed from the (X) components of the same tuples in ω_1 and ω_2. Assume $x \in x_1$. After unnesting, $\langle y\ x \rangle$ will be merged into $\{\langle y\ x' \rangle \mid x' \in (X)_y(\omega_1, \omega_2)\}$. It implies that $\langle y\ x \rangle$ would not affect the net result in either LHS or RHS. Note the definition of $(X)_y(\omega_1, \omega_2)$ is directly related to LHS.
(Case 1.2): $(x_1 - (X)_y(\omega_1, \omega_2) \neq \emptyset) \wedge ((x_1 - (X)_y(\omega_1, \omega_2)) \cap x_2 = \emptyset)$. That is, x_1 contains a set $\{x', x'', \ldots, x^n\} = T$ such that the element in T is not in $(X)_y(\omega_1, \omega_2)$ nor in x_2. For every element x in the set T, after unnesting to r_1 and r_2, $\langle y\ x \rangle$ would not be in the result with respect to either LHS or RHS. For the elements of x_1 in $(X)_y(\omega_1, \omega_2)$, the argument is similar to (Case 1.1) above.
(Case 2) $\exists t_1 = \langle y\ x_1 \rangle \in \omega_1\ \ \forall t_2 \in \omega_2,\ \ t_1 \neq t_2$ For $x \in x_1$, RHS will not contain $\langle y\ x \rangle$ at the end. Then, the only common tuples between $\mu_{(X)}(r_1)$ and $\mu_{(X)}(r_2)$ are of the form $\langle y\ x \rangle$, where $x \in (X)_y(\omega_1, \omega_2)$ if such tuple exists. Therefore, $RHS = \{t \mid \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ t[Y] = t_1[Y] = t_2[Y], t[X] \in t_1[(X)] = t_2[(X)]\}$ (By $md5$).
Thus, $LHS = RHS$.

The last mutual dependency will establish a condition under which the identity related to join and unnest will hold. Notice that such dependency is not very common in the setting of traditional relational database studies.

Definition 4.6
Let $r_1 = (\Omega_1, \omega_1)$ and $r_2 = (\Omega_2, \omega_2)$ where $\Omega_1 = (Y\ (X)), \Omega_2 = (Z\ (X))$ and $(X)_{yz}(\omega_1, \omega_2) = \{x \mid \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2,\ t_1[Y] = y, t_2[Z] = z, t_1[(X)] = t_2[(X)], x \in t_1[(X)]\}$. $r_1 \overset{md6}{\underset{(X)}{\Longleftrightarrow}} r_2 \equiv \forall t_1 \in \omega_1\ \forall t_2 \in \omega_2,\ (t_1[(X)] \cap t_2[(X)] \neq \emptyset) \wedge (t_1[(X)] \neq t_2[(X)]) \Rightarrow (t_1[(X)] - (X)_{t_1[Y], t_2[Z]}(\omega_1, \omega_2)) \cap t_2[(X)] = \emptyset$.

Lemma 4.7
$\mu_{(X)}(r_1 \bowtie r_2) = \mu_{(X)}(r_1) \bowtie \mu_{(X)}(r_2) \iff r_1 \overset{md6}{\underset{(X)}{\Longleftrightarrow}} r_2$.

Proof:

$\Rightarrow$: Suppose $r_1 \overset{md6}{\underset{(X)}{\Longleftrightarrow}} r_2$ does not hold. In relational terms, that means there are two tuples $\langle y\ x_1 \rangle \in \omega_1$ and $\langle y\ x_2 \rangle \in \omega_2$ such that $x \in (x_1 \cap x_2)$. The x is not in any element of the set composed from the (X) components derived from $\langle y\ * \rangle$ and $\langle *\ z \rangle$ where $*$ denotes the common (X) value. On LHS, $\langle y\ x\ z \rangle$ would not be in the resulting relation after unnesting. On RHS, after unnesting to r_1 and r_2, $\langle y\ x\ z \rangle$ is in the joined relation. Therefore, $LHS \neq RHS$.
$\Leftarrow$: Suppose $r_1 \overset{md6}{\underset{(X)}{\Longleftrightarrow}} r_2$. There are two possibilities to discuss under the constraint $md6$ with respect to the (X) attribute of ω_1.
(Case 1) $\exists t_1 = \langle y\ x_1 \rangle \in \omega_1\ \forall t_2 = \langle x_2\ z \rangle \in \omega_2,\ (x_1 \cap x_2 \neq \emptyset) \wedge (x_1 \neq x_2)$
(Case 1.1): $(x_1 - (X)_{yz}(\omega_1, \omega_2)) = \emptyset) \Rightarrow x_1 \subseteq (X)_{yz}(\omega_1, \omega_2)$. That is, x_1 is a subset of the set of (X) values which are in common from ω_1 and ω_2 having Y value y and Z value z respectively. After unnesting on ω_1 and ω_2, $\langle y\ x\ z \rangle$ will be merged into $\{\langle y\ x'\ z \rangle \mid x' \in (X)_{yz}(\omega_1, \omega_2)\}$ where $x \in (x_1 \cap x_2)$. This tuple would not affect the result of the join.
(Case 1.2): $(x_1 - (X)_{yz}(\omega_1, \omega_2) \neq \emptyset) \wedge ((x_1 - (X)_{yz}(\omega_1, \omega_2)) \cap x_2 = \emptyset)$. This situation is similar to (Case 1.2) of Proof 4.6 in $\Leftarrow$ direction where these tuples did not affect the join operation.
(Case 2) $\exists t_1 = \langle y\ x_1 \rangle \in \omega_1\ \forall t_2 \in \langle x_2\ z \rangle \in \omega_2,\ (x_1 \cap x_2 = \emptyset) \vee (x_1 = x_2)$.
(Case 2.1): $x_1 = x_2$ then $x_1 \subseteq (X)_{yz}(\omega_1, \omega_2)$. The joined tuples in RHS are derived in the same way as LHS.
(Case 2.2): $x_1 \cap x_2 = \emptyset$. This tuple $\langle y\ x_1 \rangle$ does not affect the join even after unnesting. Then, $RHS = \{t \mid \exists t_1 \in \omega_1\ \exists t_2 \in \omega_2, t[X] \in t_1[(X)] = t_2[(X)], t[Y] = t_1[Y], t[Z] = t_2[Z]\}$. Thus $LHS = RHS$.

Next we focus on the effects of interactions among nesting, selection and projection operators. For this paper, taking simple selection with equality into considerations is enough. In equation 4.9, Y, X and Z are assumed to have no attributes in common. Note that if $\Omega = (Y\ X)$, equation 4.8 will trivially hold.

Lemma 4.8
Let $r = (\Omega, \omega)$ and $\Omega = (Y\ X)$. Then we have $\nu_{(X)}(\sigma_{X=v}(r)) = \sigma_{v \in (X)}(\nu_{(X)}(r)) \iff \forall t \in \omega,\ v \in X_{t[Y]} \Rightarrow X_{t[Y]} = \{v\}$.

Proof:

$\Rightarrow$: Suppose $\exists t \in \omega, v \in X_{t[Y]}$ and $X_{t[Y]} \neq \{v\}$. That is, there exist two tuples $\langle y\ v \rangle, \langle y\ e \rangle$ in ω and

$v \neq e$. On LHS, there is one and only one tuple t such that $t[Y] = y$ and $e \notin t[(X)]$. On RHS, after nesting on X, there is an unique tuple t_1 such that $t_1[Y] = y$ and $\{v, e\} \subseteq t_1[(X)]$. The selection predicate then sieves out this tuple t_1. Thus, $LHS \neq RHS$.

$\Leftarrow$: Suppose $\forall t \in \omega,\ v \in X_{t[Y]} \Rightarrow X_{t[Y]} = \{v\}$.
Let $\sigma_{X=v}(r) = (\Omega, \omega')$.
Then, since $\omega' = \{t \mid \exists t \in \omega\ t[X] = v\}$,
$LHS = \{t \mid \exists t_1 \in \omega,\ t[Y] = t_1[Y], t[(X)] = \{v\} = X_{t_1[Y]} \ni v\}$. $\quad$ (*By assumption*)
$RHS = \{t \mid \exists t_1 \in \omega,\ t[Y] = t_1[Y], t[(X)] = X_{t_1[Y]} \ni v\}$
$= \{t \mid \exists t_1 \in \omega,\ t[Y] = t_1[Y], t[(X)] = \{v\} = X_{t_1[Y]} \ni v\}$. $\quad$ (*By assumption*)
Therefore, $LHS = RHS$.

Almost all the previous proofs were established with the help of newly defined mutual dependencies. In the context of traditional relational dependencies, lemma 4.9 will be the only relevant case in our current study. This reveals that in nested relational models the algebraic conditions of nested relational equations are not entirely related to the constraints in relational models.

Lemma 4.9
Let $r = (\Omega, \omega)$ and $\Omega = (Y\ X\ Z)$. $\nu_{(X)}(\pi_{(Y\ X)}(r)) = \pi_{(Y\ (X))}(\nu_{(X)}(r)) \iff Y \twoheadrightarrow X$.

Proof:

$\Rightarrow$: Suppose $Y \twoheadrightarrow X$ were not true. That is, there are two tuples $\langle y\ x\ z\rangle$, $\langle y\ x'\ z'\rangle$ in ω such that $X_{yz}(\omega) \neq X_{yz'}(\omega)$. Since $X_{yz}(\omega)$ and $X_{yz'}(\omega)$ are not empty and not equal, there are two possible cases :

(Case 1) : $\langle y\ x''\ z\rangle \in \omega\ \wedge \langle y\ x''\ z'\rangle \notin \omega$ where $x'' \in DOM(X)$. There is a tuple t on LHS such that $t[Y] = y$ and $t[(X)] \ni x''$. On RHS, after nesting on X, there would be two tuples t_1, t_2 such that $t_1[Y] = t_2[Y] = y, t_1[(X)] = X_{yz}(\omega) \neq X_{yz'}(\omega)(= t_2[(X)])$ by assumption. Thus, after projecting on $(Y\ (X))$, there will be no tuples with value y in the Y component. Therefore, $LHS \neq RHS$.

(Case 2): $\langle y\ x''\ z\rangle \notin \omega \wedge \langle y\ x'\ z'\rangle \in \omega$ The proof is analogous to (Case 1) in a symmetric sense.

$\Leftarrow$: Suppose $Y \twoheadrightarrow X$. By tuple calculus, $LHS = \{t \mid \exists t_1 \in \omega\ \forall t_2 \in \omega,\ t[Y] = t_1[Y] = t_2[Y], t[(X)] = X_{t_1[Y],t_1[Z]}(\omega) = X_{t_2[Y],t_2[Z]}(\omega)\}$ (*By MVD*). Since $\nu_{(X)}(\omega) = \{t \mid \exists t_1 \in \omega,\ t[YZ] = t_1[YZ], t[(X)] = X_{t_1[Y],t_1[Z]}\}$, $RHS = \{t \mid \exists t_1 \in \omega\ \forall t_2 \in \omega,\ t[Y] = t_1[Y] = t_2[Y], t[(X)] = X_{t_1[Y],t_1[Z]}(\omega) = X_{t_2[Y],t_2[Z]}(\omega)\}$ (*By MVD*). Therefore, $LHS = RHS$.

The proofs show that we need to introduce some new data dependencies establishing most of the equalities discussed in this section. Interactions of the nested relational operators are different from the relational operators. The conditions to establish the equations discussed in this section are not trivial. Consequently, the traditional algebraic optimization for flat relations can not be entirely applied to the nested relational models. But, there might be other modified operators having commutative properties as in the relational model. In our continued study not reported in this paper, commutativity among restructuring operators does not always hold. This contrasts with the earlier results [3] of algebraically optimizing queries mainly based on the commutativity of relational operators.

V. CONCLUSIONS

This paper discussed the algebraic properties of nested relational models. The commutative conditions of the nested relational operators considered in this paper are less straightforward compared to the relational relational operators. Besides, in general the strategy of pushing selections down the parse tree of algebraic queries in the relational model has no direct correspondence to the nested relational model. We explain the reason why commutativity is hard in algebras for nested relations. Our characterization is based on the constraint satisfaction of nested relational equations. This contrasts with the results of optimizing queries based on the commutativity of relational operators. Our suggestion is that algebraic attempts should be pursued cautiously for the query optimization of nested relational models.

Actually, we also should look at the positive side of this optimization problem. One of the advantages of the nested relational model is its ability to store the implicit join information in the nested relations. Therefore, for some queries, join computations are thus saved. As mentioned in [17], indexing methods used in the physical implementation could be useful for optimizing queries. The performance issues of such strategy need to be analyzed. Some directions different from the algebraic approach to the nested relational models are possible. Transformation of queries in between the nested relational algebra and calculus is one possibility. Optimization in the context of restricted nested relational model would be another direction.

VI. ACKNOWLEDGEMENTS

I thank Dirk Van Gucht for discussions on this subject. My thanks also go to Richard Mong, Bob Montante and referees for their suggestions on this paper.

REFERENCES

1. S. Abiteboul, C. Beeri, "On the Power of Languages for the Manipulation of Complex Objects", *Proc. International Workshop on Theory and Applications of Nested Relations and Complex Objects*, Darmstadt, April 1987, also: *INRIA Technical Report 846*, May 1988.
2. S. Abiteboul, P. C. Fischer, H.-J. Schek (Eds.), "Nested Relations and Complex Objects in Databases", *in Lecture Notes of Computer Science 361*, Springer Verlag, 1989.
3. A.V. Aho, J.D. Ullman, "Universality of Data Retrieval Languages", *Proc. 6th Symposium on Principles of Programming Languages*, San-Antonio TX, January 1979, pp. 110–117.
4. F.Bancilhon, "On the Completeness of Query Languages for Relational Data Bases", *Proc. 7th Symp. on Mathematical Foundations of Computer Science*, Zakopane, Poland, *in Lecture Notes of Computer Science 64*, Springer Verlag, 1978, pp. 112–123.
5. J. Banerjee, W. Kim, H-J Kim, H. F. Korth, "Semantics and Implementation of Schema Evolution in Object-Oriented Databases", *Proc. of ACM SIGMOD*, 1987, pp. 311-322.
6. C. Beeri, S. Naqvi, R. Ramakrishnan, O. Shmueli, S. Tsur, "Sets and Negation in Logic Database Language", *Proc. 6th PODS*, 1987, pp. 21–37.
7. N. Bidoit, "The Verso algebra or How to Answer Queries with Fewer Joins", *JCSS 35*, 1987, pp. 321-364.
8. M.J. Carey, D.J. DeWitt, J.E. Richardson, E.J. Shekita, "Object and file management in the exodus extensible database system", *Proc. 12th International Conf. on VLDB*, 1986, pp. 91–100.
9. A.K. Chandra, D. Harel, "Structure and Complexity of Relational Queries", *Journal of Computer and System Sciences 25*, 1985, pp 99–128.
10. E.F. Codd, "A Relational Model of Data for Large Shared Data Banks", *Communications of ACM 13:6*, June 1970, pp. 377–387.
11. E.F. Codd, "Further Normalizations of the Relational Data Base Model", *Data Base Systems*, R. Rustin, ed., Prentice Hall, Englewood Hills NJ, 1972, pp. 33–64.
12. U. Dayal, F. Manola, A. Bachmann, U. Chakravarthy, D. Goldhirsch, S. Heiler, J. Orenstein, A. Rosenthal, "Simplifying complex objects : The *PROBE* approach to modelling and querying them", *Proc. GI Conf. on Database Systems for Office Automation, Engineering and Scientific Applications*, 1987.
13. M. Gyssens, J. Paredaens, D. Van Gucht, "A Uniform Approach towards Handling Atomic and Structured Information in the Nested Relational Database Model", *Technical Report 88-17*, University of Antwerp, 1988, to appear in *Journal of ACM*.
14. M. Gyssens, D. Van Gucht, "The Powerset Algebra as an Algebraic Tool for Understanding Least Fixpoint Semantics in the Context of Nested Relations", *Technical Report 233*, Indiana University, 1987.
15. R. Hull, J. Su, "On the Expressive Power of Database Queries with Intermediate Types", *Proc. 7th Symp. on Principles of Database Systems*, Austin TX, 1988, pp. 39–51.
16. G. Jaeschke, H.J. Schek, "Remarks on the Algebra on Non First Normal Form Relations", *Proc. 1st Symp. on Principles of Database Systems*, Los Angeles, 1982, pp. 124–138.
17. H.F. Korth, "Optimization of Object-Retrieval Queries", *2nd International Workshop on Object-Oriented Database Systems*, 1988, pp. 352–357.
18. G.M. Kuper, "Logic Programming with sets", *Proc. 6th PODS*, San Diego, 1987, pp. 11-20.
19. G.M. Kuper, M.Y. Vardi, "On the Complexity of Queries in the Logical Data Model", *Proceedings 2nd International Conference on Database Theory, in Lecture Notes in Computer Science 326*, Springer-Verlag, 1988, pp. 267–280.
20. L. S. Colby, "A Recursive Algebra and Query Optimization for Nested Relations", *Proc. of 1989 ACM SIGMOD*, 1989, pp. 273-283.
21. S. L. Osborn, "Identity, Equality and Query Optimization", *2nd International Workshop on Object-Oriented Database Systems*, 1988, pp. 346–351.
22. P. C. Kanellakis, "Elements of Relational Database Theory", *Technical Report No. CS-88-09*, April, 1988.
23. J. Paredaens, D. Van Gucht, "Possibilities and Limitations of Using Flat Operators in Nested Algebra Expressions", *Proc. 7th ACM PODS*, 1988, pp. 29-38.
24. M.A. Roth, H.F. Korth, A. Silberschatz, "Extended Algebra and Calculus for ¬1NF Relational Databases", *Trans. on Database Systems 13:4*, December 1988, pp. 389–417.
25. M.H. Scholl, "Theoretical Foundation of Algebraic Optimization Utilizing Unnormalized Relations", *ICDT*, 1986, pp. 380–396.
26. S.J. Thomas, P.C. Fischer, "Nested Relational Structures", *The Theory of Databases*, P.C. Kanellakis, ed., JAI Press, 1986, pp. 269–307.
27. D. Van Gucht, "On the Expressive Power of the Extended Relational Algebra for the Unnor-

malized Relational Model", *Proc. 6th PODS*, San Diego CA, March 1987, pp. 302–312.