On Representation and Querying Incomplete Information in Databases with Bags

Leonid Libkin
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974, USA
Email: libkin@research.att.com

Limsoon Wong
Real World Computing Partnership Novel Function
Institute of Systems Science Laboratory
Heng Mui Keng Terrace, Singapore 0511
Email: limsoon@iss.nus.sg

Abstract

We extend the approach to representation of partial information based on orderings on objects from sets to multisets. We characterize orderings arising under closed- and open-world assumptions and analyze their complexity. In contrast to the set case, where orderings are first-order definable and are thus expressible in standard database query languages, the orderings on bags are not expressible in standard bag languages. We give an example of a query on nested relations whose inexpressibility in the extension of relational algebra to nested objects cannot be proved by reduction to the first-order case.

1 Introduction

One approach to representing partial information in databases that treats objects as elements of partially ordered sets, where the meaning of the order is "being more informative", has proved to be very useful for analyzing incompleteness of information in the relational database model and its extensions [3, 5, 9, 12]. In particular, it has allowed a number of powerful tools from denotational semantics of programming languages to be used to analyze the phenomenon of incomplete information [1, 4, 7, 8]. All these papers deal with set-based databases. However, real database systems frequently use bags (also called multisets) as the underlying model. While there was a flurry of activity in studying many aspects of bag-based databases – query language design, expressive power and complexity, query optimization and view maintenance, to name a few – no work has been done on partial information and bags.

We present two main results. First, we extend the order-theoretic approach from sets to bags. This results in two orderings that must be used for bags under the open- and the closed-world assumptions. Second, we prove that these orderings are not definable in basic bag algebras such as those in [2, 11]. This may impact query language design for bag-based databases. While in the set case, orderings are definable in any language that extends relational calculus, in the bag case one may have to enrich basic languages with primitives capable of expressing these orderings. In addition, as a somewhat unexpected corollary, we prove that the existence of systems of distinct representatives cannot be tested in the nested relational algebra, which is a natural extension of relational algebra to nested relations or complex objects. Unlike all other known limitations of expressive power of the nested relational algebra, this one cannot be proved by a reduction to the first-order case.

2 Ordering on Objects and Partiality

Database objects are built from base objects such as tuples of reals, booleans, etc. by using *collection type* constructors such as sets and bags. The "being more informative" ordering is given for values of base types, usually by specifying possible null values; for example, the no-information null $\perp$ is less informative than any non-null value. Therefore, to order arbitrary database objects, one has to *lift* orderings to collections.

We consider sets first. Suppose a partially-ordered set $\langle A, \leq \rangle$ is given and we want to lift $\leq$ to finite subsets of A. Three liftings are frequently used in semantics of concurrency [4]: the Hoare, Smyth, and Plotkin orderings defined below. They are generally pre-orders. However, when restricted to *antichains*, which are sets without comparable elements, they become partial orders.

(Hoare)	$X \leq^{\mathrm{H}} Y$	$\Leftrightarrow$	$\forall x \in X\ \exists y \in Y:\quad x \leq y$
(Smyth)	$X \leq^{\mathrm{S}} Y$	$\Leftrightarrow$	$\forall y \in Y\ \exists x \in X:\quad x \leq y$
(Plotkin)	$X \leq^{\mathrm{P}} Y$	$\Leftrightarrow$	$X \leq^{\mathrm{H}} Y$ and $X \leq^{\mathrm{S}} Y$

Ordering Sets of Incomplete Objects

To define orderings for collections, we adopt the approach of [9]. A collection C_1 is more informative than C_2 if C_2 can be obtained from C_1 by a sequence of elementary updates that add information. This approach reduces the problem of choosing an ordering to the problem of formulating elementary updates. Such updates depend on certain assumptions on partial data. We consider two, following [5, 3]. The *closed-world assumption* or *CWA*, says that only elementary updates that improve knowledge about objects in the database are allowed. That is, adding new objects is not allowed, unless they improve upon objects already in the database. In contrast, the *open-world assumption* or *OWA*, allows both adding objects and improving knowledge of objects already stored. That is, databases are open for new objects.

To formalize this, we define the following updates on subsets of $\langle A, \leq \rangle$:

$X \overset{\mathrm{CWA}}{\sqcup\!\!\rightarrow} (X \perp \{x\}) \cup X'$ where $x \in X$, $X' \neq \emptyset$ and $x \leq x'$ for all $x' \in X'$.
$X \overset{\mathrm{OWA}}{\sqcup\!\!\rightarrow} Y$ iff $X \overset{\mathrm{CWA}}{\sqcup\!\!\rightarrow} Y$ or $X \subseteq Y$.

Let $\sqsubseteq^{\mathrm{CWA}}$ and $\sqsubseteq^{\mathrm{OWA}}$ denote the transitive-reflexive closure of $\overset{\mathrm{CWA}}{\sqcup\!\!\rightarrow}$ and $\overset{\mathrm{OWA}}{\sqcup\!\!\rightarrow}$. That is, $X \sqsubseteq^{\mathrm{CWA}} Y$ or $X \sqsubseteq^{\mathrm{OWA}} Y$ if Y is obtained from X by a sequence of allowed updates that add information.

Proposition 2.1 (See [9, 8].) *$X \sqsubseteq^{\mathrm{OWA}} Y$ iff $X \leq^{\mathrm{H}} Y$, and $X \sqsubseteq^{\mathrm{CWA}} Y$ iff $X \leq^{\mathrm{P}} Y$. Moreover, this continues to hold when subsets are restricted to antichains and updates are modified in such a way that after each update only maximal (most informative) elements are retained.* □

We should remark that the Smyth ordering $\leq^{\mathrm{S}}$ corresponds to *or-sets* that are sets of possible choices. This was first observed in [14] and formalized in a way similar to Proposition 2.1 in [9]. It should also be noted that in early work on using orderings for partiality, orderings for collections were usually chosen in an *ad hoc* way, without any justification.

Ordering Bags of Incomplete Objects

We now use similar techniques to define orderings for *bags*. To extend the update idea, notice that in bags we do not identify objects even if information we have about them is the same, since later we may obtain additional information that would distinguish one object from the other.

This justifies the following definition. Bag B_2 is more informative than bag B_1 if B_2 can be obtained from B_1 by a sequence of updates of the following form: (1) an element a in B_1 is replaced by an element b such that b is more informative than a, and under the OWA, (2) an element b is added to B_1. Notice that in contrast to the set case, updates of form (1) replace an element by an element. In sets we do identify elements if we have the same information about them. Thus, we had to permit replacement of an element by a set to account for the fact that one element of a set may represent more than one object.

Formally, let $\langle A, \leq \rangle$ be a partially-ordered set. We use the $\{\!|\,|\!\}$ brackets for bags. We also use $\uplus$ for additive union and $\dot{-}$ for bag difference. Updates are defined as follows:

$B \overset{\text{CWA}}{\rightsquigarrow} (B \dot{-} \{\!|a|\!\}) \uplus \{\!|b|\!\}$ if $a \in B$ and $a \leq b$.
$B \overset{\text{OWA}}{\rightsquigarrow} B'$ iff $B \overset{\text{CWA}}{\rightsquigarrow} B'$ or $B' = B \uplus \{\!|b|\!\}$.

As for sets, we denote the reflexive-transitive closure of $\overset{\text{CWA}}{\rightsquigarrow}$ and $\overset{\text{OWA}}{\rightsquigarrow}$ respectively by $\trianglelefteq^{\text{CWA}}$ and $\trianglelefteq^{\text{OWA}}$. To describe these relations, let $\mathbb{N}^{\shortparallel}$ denote the totally unordered poset whose elements are natural numbers. (The superscript is used to distinguish it from $\mathbb{N}$, which typically denotes natural numbers with the usual ordering.) Given a finite bag B and an injective map $\phi : B \to \mathbb{N}^{\shortparallel}$, which is called a *labeling*, we denote the set $\{(b, \phi(b)) \mid b \in B\}$ by $\phi(B)$. In other words, ϕ assigns a unique label to each element of a bag. The ordering on pairs (b, n) where $b \in B$ and $n \in \mathbb{N}^{\shortparallel}$ is the usual pair ordering; that is, $(b, n) \leq (b', n')$ iff $b \leq b'$ and $n = n'$.

Proposition 2.2 *Binary relations $\trianglelefteq^{\text{CWA}}$ and $\trianglelefteq^{\text{OWA}}$ on bags are partial orders. Given two bags B_1 and B_2, $B_1 \trianglelefteq^{\text{CWA}} B_2$ ($B_1 \trianglelefteq^{\text{OWA}} B_2$) iff there exist labelings ϕ and ψ on B_1 and B_2 such that $\phi(B_1) \leq^{\text{P}} \psi(B_2)$ (respectively $\phi(B_1) \leq^{\text{H}} \psi(B_2)$).*

Proof. We prove the statement about $\trianglelefteq^{\text{OWA}}$; the statement about $\trianglelefteq^{\text{CWA}}$ is proved similarly. We write $B_1 \preccurlyeq B_2$ if there exist ϕ and ψ such that $\phi(B_1) \leq^{\text{H}} \psi(B_2)$. We first demonstrate that $\preccurlyeq$ is a partial order.

Reflexivity is obvious. To prove transitivity, let $B_1 \preccurlyeq B_2$ and $B_2 \preccurlyeq B_3$. That is, $\alpha(B_1) \leq^{\text{H}} \beta(B_2)$ and $\phi(B_2) \leq^{\text{H}} \psi(B_3)$. Let γ be a bijection on $\mathbb{N}$ such that $\gamma \circ \beta = \phi$. Define δ as $\gamma \circ \alpha$. Then for every $b \in B_1$ there is $b' \in B_2$ such that $b \leq b'$ and $\alpha(b) = \beta(b')$. Therefore, $\delta(b) = \phi(b')$ and there exists $b'' \in B_3$ such that $\psi(b'') = \phi(b')$ and $b'' \geq b'$. This shows $\delta(B_1) \leq^{\text{H}} \psi(B_3)$ and hence $B_1 \preccurlyeq B_3$.

To show that $\preccurlyeq$ is anti-symmetric, let $B_1 \preccurlyeq B_2$ and $B_2 \preccurlyeq B_1$. As was shown above, there exist α, ϕ and ψ such that $\alpha(B_1) \leq^{\text{H}} \phi(B_2) \leq^{\text{H}} \psi(B_1)$. In particular, if we define $g : \alpha(B_1) \to \psi(B_1)$ by $g(b, n) = (b', n)$ where $\psi(b') = n$, then g is one-to-one and inflationary. Since B_1 is finite, it is the identity map. If $b'' \in B_2$ and $\phi(b'') = n$, then $b \leq b'' \leq b' = b$, so $b = b''$ where $\alpha(b) = \psi(b') = n$. Therefore, every element of B_1 is in B_2 and vice versa, i.e. $B_1 = B_2$. This shows that $\preccurlyeq$ is a partial order.

Since $B_1 \overset{\text{OWA}}{\rightsquigarrow} B_2$ implies $B_1 \preccurlyeq B_2$, we conclude $\trianglelefteq^{\text{OWA}} \subseteq \preccurlyeq$. Conversely, if $B_1 \preccurlyeq B_2$, i.e. $\phi(B_1) \leq^{\text{H}}$

$\psi(B_2)$, then, according to Proposition 2.1, $\psi(B_2)$ can be obtained from $\phi(B_1)$ by a sequence of $\overset{\mathrm{OWA}}{\sqcup\!\!\to}$ updates which, if we drop labels, are translated into $\overset{\mathrm{OWA}}{\rightsquigarrow}$ updates on bags. Therefore, $B_1 \trianglelefteq^{\mathrm{OWA}} B_2$, which proves $\trianglelefteq^{\mathrm{OWA}} = \preccurlyeq$. $\square$

The Hoare ordering $\leq^{\mathrm{H}}$ on sets can be effectively verified. Indeed, if two sets are given, there is an $O(n^2)$ time algorithm to check if they are comparable. The description of $\trianglelefteq^{\mathrm{OWA}}$ and $\trianglelefteq^{\mathrm{CWA}}$ given above seems to be somewhat awkward, algorithmically. However, it is not much harder to test for.

Proposition 2.3 *There exists an $O(n^{5/2})$ time algorithm that, given two bags B_1 and B_2 of elements of a poset A, returns* true *if $B_1 \trianglelefteq^{\mathrm{OWA}} B_2$ ($B_1 \trianglelefteq^{\mathrm{CWA}} B_2$) and* false *otherwise, provided that the ordering on A can be tested in $O(1)$ time.*

Proof. The proof is almost the same for both $\trianglelefteq^{\mathrm{OWA}}$ and $\trianglelefteq^{\mathrm{CWA}}$. Given B_1 and B_2, consider two labelings ϕ and ψ on B_1 and B_2 with disjoint codomains. Define a bipartite graph $G = \langle V, E\rangle$ by $V := \phi(B_1) \cup \psi(B_2)$ and $E := \{((b,n),(b',n')) \mid (b,n) \in \phi(B_1), (b',n') \in \psi(B_2), b \leq b'\}$. It can be easily concluded from Proposition 2.2 that $B_1 \trianglelefteq^{\mathrm{OWA}} B_2$ iff there is a matching in G that contains all $\phi(B_1)$. In other words, $B_1 \trianglelefteq^{\mathrm{OWA}} B_2$ iff the cardinality of the maximal matching in G is that of B_1. The proposition now follows from the facts that all maximal matching in G have the same cardinality and that the Hopcroft-Karp algorithm finds a maximal matching in $O(n^{5/2})$ where $n = |V|$. $\square$

There is a big difference between orders on sets and bags. While $X \leq^{\mathrm{H}} Y$ does not say anything about cardinality of X and Y, $B_1 \trianglelefteq^{\mathrm{OWA}} B_2$ implies that the cardinality of B_1 is at most the cardinality of B_2. Indeed, elements of a bag represent distinct objects that cannot be identified in the process of gaining information. Under the CWA, cardinality is always preserved, which also conforms to the intuition about the closed worlds.

Remark. Using bags to represent incomplete information was also studied in [6, 15], but the focus of these papers is very different from ours. In [15] a *bagdomain* is defined, which is a category whose objects are bags over an ordered set, and morphisms are so-called refinements. Both $\trianglelefteq^{\mathrm{OWA}}$ and $\trianglelefteq^{\mathrm{CWA}}$ are examples of refinements, but so are many other orderings that are not well suited for representing partiality in databases. In [6] the main construction of [15] is extended so that it can be viewed as freely generated in a certain sense by the underlying poset.

3 Querying Bags of Incomplete Objects

The orderings $\leq^{\mathrm{H}}$ and $\leq^{\mathrm{P}}$ used for sets under the OWA and the CWA are defined by first-order formulae. Thus, any language with relational algebra or calculus as a sublanguage has enough power to lift $\leq$ to $\leq^{\mathrm{H}}$ and $\leq^{\mathrm{P}}$. The situation is very different in the bag case.

In order to demonstrate this result, we need a "standard" language for bags that has a role similar to that of relational calculus for sets. Such a language has recently been proposed and studied [2, 11]. The object types are given by the grammar

$$t := b \mid t \times t \mid \{\!|t|\!\}$$

where b ranges over a collection of base types (among them *bool*, the type of Booleans), $t_1 \times t_2$ is the type of pairs (we use pairs rather than records to keep notation simple), and values of type $\{\!|t|\!\}$ are finite bags of values of type t. Note that we deal with complex objects, which may involve nesting of bags, and not just flat bags, which are bags of records of base types.

The operations of the language include pair formation and projections, function composition, *if-then-else*, and the following operations on bags (in addition, empty bag is available as constant):

- $b\_\eta : t \to \{\!|t|\!\}$ forms singleton bags.
- $\uplus : \{\!|t|\!\} \times \{\!|t|\!\} \to \{\!|t|\!\}$ is the additive union of bags.
- $\dot{-} : \{\!|t|\!\} \times \{\!|t|\!\} \to \{\!|t|\!\}$ is the bag difference.
- $b\_\mu : \{\!|\{\!|t|\!\}|\!\} \to \{\!|t|\!\}$ flattens a bag of bags, adding up multiplicities.
- $unique : \{\!|t|\!\} \to \{\!|t|\!\}$ is the duplicate elimination operation.
- $b\_map(f) : \{\!|s|\!\} \to \{\!|t|\!\}$ applies the function f of type $s \to t$ to every element of a bag of type $\{\!|s|\!\}$.
- $b\_\rho_2 : s \times \{\!|t|\!\} \to \{\!|s \times t|\!\}$ is the function that pairs an object with every element in a bag.

We call this language BQL [11] (also called BALG without powerset in [2]). Now we prove our main result that BQL cannot express an algorithm that lifts a binary relation $\leq$ to $\trianglelefteq^{\mathrm{OWA}}$ or $\trianglelefteq^{\mathrm{CWA}}$ in the way described in Proposition 2.2.

Theorem 3.1 *The orderings $\trianglelefteq^{\mathrm{OWA}}$ and $\trianglelefteq^{\mathrm{CWA}}$ cannot be defined in BQL.*

Proof. Let b be a base type with infinite domain, with only equality test available on it. A directed graph $X : \{\!|b \times b|\!\}$ is called a **chain** if it has the form $\{\!|(x_1, x_2), (x_2, x_3), \ldots, (x_{m-1}, x_m)|\!\}$, where all x_i's are distinct. Let $\mathsf{chaineven} : \{\!|b \times b|\!\} \to bool$ be a predicate such that for every chain X, $\mathsf{chaineven}(X)$ is true iff X has an even number of nodes.

A bag of bags $\mathcal{B} = \{\!|B_1, \ldots, B_n|\!\} : \{\!|\{\!|b|\!\}|\!\}$ is said to have a family of distinct representatives iff it is possible to pick an element x_i from each B_i such that $x_i \neq x_j$ whenever $i \neq j$. Note that both $\mathcal{B}$ and its elements are allowed to have duplicates, but the x_i picked must be distinct. Let $\mathsf{sdr} : \{\!|\{\!|b|\!\}|\!\} \to bool$ be the predicate such that for every bag of bags $\mathcal{B}$, $\mathsf{sdr}(\mathcal{B})$ is true iff $\mathcal{B}$ has a family of distinct representatives.

The proof is based on the following two lemmas.

Lemma 3.2 *Let $X_m : \{\!|b|\!\}$ be a chain $\{\!|(x_1, x_2), \ldots, (x_{m-1}, x_m)|\!\}$. Define $\mathcal{S}_m : \{\!|\{\!|b|\!\}|\!\}$ to be the bag $\{\!|\{\!|x_1|\!\}, \{\!|x_m|\!\}, \{\!|x_1, x_3|\!\}, \{\!|x_2, x_4|\!\}, \ldots, \{\!|x_{m-2}, x_m|\!\}|\!\}$, as depicted in Figure 1. Then for $m > 2$, $\mathsf{sdr}(\mathcal{S}_m)$ is true iff m is even.*

Lemma 3.3 *BQL cannot express* $\mathsf{chaineven}$.

First, let us show how the theorem follows from the lemmas. Consider a chain X_m as in Lemma 3.2 and construct two bags of bags (objects of type $\{\!|\{\!|b|\!\}|\!\}$): one is $\mathcal{S}_m$ and the other is $\mathcal{T}_m = \{\!|\{\!|x_1|\!\}, \{\!|x_2|\!\}, \ldots, \{\!|x_{m-1}|\!\}, \{\!|x_m|\!\}|\!\}$. Both $\mathcal{T}_m$ and $\mathcal{S}_m$ are definable in BQL. For example, $\mathcal{T}_m = b\_map(b\_\eta)(unique(\uplus(b\_map(\pi_1)(X_m), b\_map(\pi_2)(X_m))))$, where π_1 and π_2 are the first and the second projections.

For bags of type $\{\!|b|\!\}$, define $B_1 \leq B_2$ iff B_1 is a subbag of B_2. This ordering is definable in BQL; see [11]. Assume that an algorithm lifting $\leq$ to $\trianglelefteq^{\mathrm{OWA}}$ or to $\trianglelefteq^{\mathrm{CWA}}$ is definable in BQL. Since $\mathcal{T}_m$ and $\mathcal{S}_m$ have the same cardinality, $\mathcal{T}_m \trianglelefteq^{\mathrm{OWA}} \mathcal{S}_m$ iff $\mathcal{T}_m \trianglelefteq^{\mathrm{CWA}} \mathcal{S}_m$. Moreover, it is immediately seen from

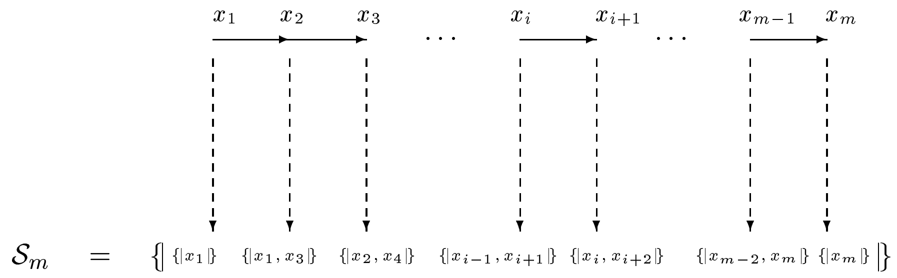

Figure 1: Definition of $\mathcal{S}_m$

Proposition 2.2 that $\mathcal{T}_m \trianglelefteq^{\text{OWA}} \mathcal{S}_m$ iff $\mathcal{S}_m$ has a family of distinct representatives. (Those representatives are given by the matching between $\mathcal{S}_m$ and $\mathcal{T}_m$ that $\trianglelefteq^{\text{OWA}}$ or $\trianglelefteq^{\text{CWA}}$ establishes.)

Thus, if $\trianglelefteq^{\text{OWA}}$ or $\trianglelefteq^{\text{CWA}}$ were definable, it would be possible to define a function in BQL that, for a chain X_m, tests if $\mathsf{sdr}(\mathcal{S}_m)$ is true. That is, it tests if m is even, according to Lemma 3.2. Then it would be possible to write the $\mathsf{chaineven}$ query in BQL, but this is impossible, according to Lemma 3.3. So, to finish the proof of the theorem, it remains to prove the two lemmas.

Proof of Lemma 3.2. First, fix some notation. Given X_m, let Y_i^m be $\{\!|x_1|\!\}$ for $i = 1$, $\{\!|x_m|\!\}$ for $i = m$, and $\{\!|x_{i-1},\ x_{i+1}|\!\}$ for $1 < i < m$. If a family $\{\!|Y_i^m|\!\}$ of bags has a system of distinct representatives, then we use $c(Y_i^m)$ to denote the representative of Y_i^m.

We prove this claim by induction on m. For $m = 3$ or 4 it is easy to see that claim is true. Now, assume that $m > 4$ and m is even. By induction hypothesis, we know $\mathcal{S}_{m-2}$ has a system of distinct representatives. For any $i < m \perp 2$, $Y_i^m = Y_i^{m-2}$. Furthermore, $Y_{m-2}^{m-2} = \{\!|x_{m-2}|\!\}$, $Y_{m-2}^m = \{\!|x_{m-3}, x_{m-1}|\!\}$, $Y_{m-1}^m = \{\!|x_{m-2}, x_m|\!\}$, $Y_m^m = \{\!|x_m|\!\}$. Then $\mathcal{S}_m$ has a system of distinct representatives: for $k < m \perp 2$, $c(Y_k^m) = c(Y_k^{m-2})$, $c(Y_{m-2}^m) = x_{m-1}$, $c(Y_{m-1}^m) = x_{m-2}$ and $c(Y_m^m) = x_m$.

Now let $m > 4$ be odd. We know $\mathcal{S}_{m-2}$ does not have a system of distinct representatives. Assume $\mathcal{S}_m$ does have it. Then $c(Y_m^m) = x_m$, $c(Y_{m-1}^m) = x_{m-2}$, and $c(Y_{m-2}^m)$ is either x_{m-3} or x_{m-1}. If $c(Y_{m-2}^m) = x_{m-3}$, then x_{m-1} is not present in any other Y_l^m and hence will never get selected. But since the cardinalities of X_m and $\mathcal{S}_m$ coincide, this means $\mathcal{S}_m$ does not have a system of distinct representatives. Thus, $c(Y_{m-2}^m) = x_{m-1}$ and for any $i < m \perp 2$, $c(Y_i^m) = x_j$ where $j < m \perp 2$. Since $Y_i^m = Y_i^{m-2}$ for $i < m \perp 2$, then by taking $c(Y_i^{m-2}) = c(Y_i^m)$ for $i < m \perp 2$ and $c(Y_{m-2}^{m-2}) = x_{m-2}$ we obtain a system of distinct representatives for $\mathcal{S}_{m-2}$, contradiction. Lemma 3.2 is proved.

Proof of Lemma 3.3. We prove this claim via a detour to a nested relational algebra with arithmetic operations and aggregate functions. According to [11], the language BQL has exactly the same power as the set language that we call NRL^{nat} here (cf. [11, 10]). Its types are given by the grammar $t := b \mid \mathbb{N} \mid t \times t \mid \{t\}$ where values of type $\{t\}$ are finite sets of values of type t. The operations on records are the same as those of BQL. The set operations are s_η, $\cup$, s_μ, s_ρ_2 and s_map, which correspond to similar operations of BQL, but duplicates are eliminated. Also, equality test $eq : t \times t \to bool$ is available for all types. The operations on natural numbers include addition, multiplication, modified subtraction $\dot{-}$, and the summation operator $\sum[f] : \{t\} \to \mathbb{N}$, where $f : t \to \mathbb{N}$, with semantics $\sum[f](\{x_1, \ldots, x_n\}) = f(x_1) + \ldots + f(x_n)$. The language NRL obtained from NRL^{nat} by removing the arithmetic operations is equivalent to the nested relational algebra, which is a generalization of relational algebra to complex objects (cf. [13, 16]).

According to [11], for any boolean query q of type $\{b \times b\} \rightarrow bool$ in NRL^{nat}, there exists a number k such that for any $l_1, l_2 \geq k$ and any two cycles C_1 and C_2 of length l_1 and l_2 respectively, $q(C_1) = q(C_2)$. Thus, NRL^{nat} cannot define a query that is equivalent to chaineven on sets, because it is possible to use chaineven to distinguish cycles of even and odd cardinality. Since BQL and NRL^{nat} are equally expressive, we conclude that chaineven is not expressible in BQL. This finishes the proof of the theorem. □.

Now observe the we can extract the following corollary from the proof of our main theorem.

Corollary 3.4 *The nested relational algebra cannot test whether a family of sets has a system of distinct representatives.* □

What makes this result different from other known limitations of the nested relational algebra is that it cannot be proved by reduction to the first-order case. So far, all inexpressibility results for the nested relational languages were proved in the following way. First, a *conservativity* result is established that shows that expressive power of the language is independent of the depth of set nesting in intermediate results (see [10, 13, 16] for examples of such results). Then the desired results are proved by reduction to the first-order case, when no nested relations are allowed. For example, the flat fragment of the nested relational algebra is equivalent to the relational algebra [13, 16]. Hence, recursive queries such as transitive closure cannot be expressed. In contrast, the query asking whether there exists a system of distinct representatives requires set nesting of depth two, and it does not have a flat analog. Thus, it cannot be proved by standard conservativity techniques.

Acknowledgements. We thank anonymous referees for their helpful suggestions.

References

[1] P. Buneman, A. Jung, A. Ohori. Using powerdomains to generalize relational databases. *Theoretical Computer Science* 91(1991), 23–55.

[2] S. Grumbach and T. Milo. Towards tractable algebras for bags. *Proc. of the 12th Symposium on Principles of Database Systems*, Washington DC, 1993, pages 49–58.

[3] G. Grahne. *"The Problem of Incomplete Information in Relational Databases"*. Springer-Verlag, Berlin, 1991.

[4] C. Gunter. *"Semantics of Programming Languages"*. The MIT Press, 1992.

[5] T. Imielinski and W. Lipski. Incomplete information in relational databases. *Journal of ACM* 31(1984), 761–791.

[6] P.T. Johnstone. Partial products, bagdomains and hyperlocal toposes. In *Applications of Categories in Computer Science*, London Math. Soc. Lecture Notes, v. 177, 1992, pages 315–339.

[7] M. Levene and G. Loizou. The nested relation type model: An application of domain theory to databases. *The Computer Journal* 33 (1990), 19-30.

[8] L. Libkin. Approximation in databases. In G. Gottlob and M. Vardi, eds., *LNCS 893: Proc. Internat. Conference on Database Theory, Prague, 1995*, pages 411–424. Springer-Verlag, 1995.

[9] L. Libkin and L. Wong. Semantic representations and query languages for or-sets. *Proc. of the 12th Symposium on Principles of Database Systems,* Washington, DC, May 1993, pages 37–48.

[10] L. Libkin and L. Wong. Conservativity of nested relational calculi with internal generic functions. *Information Processing Letters* 49 (1994), 273–280.

[11] L. Libkin and L. Wong. New techniques for studying set languages, bag languages and aggregate functions. In *Proc. of the 13th Symposium on Principles of Database Systems,* Minneapolis MN, May 1994, pages 155–166.

[12] A. Ohori. Orderings and types in databases. In *"Advances in Database Programming Languages"* (F. Bancilhon and P. Buneman, eds.), ACM Press, 1990, pages 97–116.

[13] J. Paredaens and D. Van Gucht. Converting nested relational algebra expressions into flat algebra expressions. *ACM Transaction on Database Systems*, 17 (1992), 65–93.

[14] B. Rounds. Situation-theoretic aspects of databases. In *Proceedings of Conference on Situation Theory and Applications*, CSLI vol. 26, 1991, pages 229-256.

[15] S. Vickers. Geometric theories and databases. In *Applications of Categories in Computer Science*, London Math. Soc. Lecture Notes, v. 177, 1992, pages 288–314.

[16] L. Wong. Normal forms and conservative properties for query languages over collection types. In *Proc. of the 12th Symposium on Principles of Database Systems,* Washington, DC, May 1993, pages 26–36.