Algebraic Properties of Bag Data Types

Joseph Albert*
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304

Abstract

We explore the implications of supporting bags (i.e. multisets) in a data model and associated query language, and present some formal results concerned with the algebraic properties of bags. We extend previous work to provide a formal basis for query optimization and for defining the appropriate semantics for bag operations in data models supporting bags.

1 Introduction

A bag (or multiset) is a collection of elements, like a set, but which may contain duplicates. A bag which contains no duplicates is a set. Various proposed and existing database systems have been designed to support bags in their data model [1, 2, 5, 6, 10, 15, 16, 17]. Typical motivations for this choice are based upon the desired semantic modeling capability of the data model, or upon saving the cost of duplicate elimination that is required to implement some set operations.

Some authors, for example Klug [9], have argued against supporting bags in a data model. It is not our purpose to validate or invalidate the use of bags. Rather, we wish to understand further the properties of bags since many data models have been designed to support them.

In this paper, we extend the typical set operations, union, intersection, difference, and boolean selection to bags, and derive some theoretical results regarding the algebraic properties of these extended operations. In particular, we address the issue of which algebraic properties of union, intersection, difference, and boolean selection can be maintained in the extension of these operations to bags, and which properties of these set operations fail for bags.

This work grew out of a study of the semantics of bag operations in OSQL, the query language for Iris [5]. However, we have presented the results in a general form to be applicable to most data models which support bags. We would like to develop a formal basis for query optimization for data models which are based on bags.

The notion of using algebraic transformations for query optimization was originally developed for the relational algebra. A survey of these techniques for the relational algebra can be found in [18]. Given some expression in the relational algebra, the idea is to try to apply algebraic transformation rules to the expression to find an equivalent expression which is cheaper to evaluate.

The key point regarding union, intersection, and complement for sets is that these operations satisfy the axioms of a boolean algebra. A complete list of these axioms may be found in any one of many texts on the subject, for example [11]. The transformation rules that can be applied to an expression involving union, intersection, and complement (or difference) are precisely the algebraic identities of a boolean algebra (for example, DeMorgan's Laws). Thus, when extending union, intersection, and difference to bags, we would like to preserve (as much as is possible) the boolean algebra structure.

There has been some previous research in extending the set operations union, intersection and difference to bags, as well as in developing techniques for algebraic query optimization for data models supporting bags.

*Author's present address: Computer Sciences Dept., University of Wisconsin-Madison, 1210 W. Dayton St., Madison, WI 53706, albert@cs.wisc.edu. The author was supported under DARPA contract N00014-85-K-0788 while a final revision of this paper was performed.

Dayal et al. [4] define the relational operators for bags, and present some of the usual algebraic identities for sets. They define a framework for query optimization, and state that the algebraic identities applicable to sets will continue to hold for bags, since union, intersection, and difference for bags form a boolean algebra.

We show that in general, no boolean algebra structure is available for bags if it is desired that the bag operations have their standard semantics when restricted to sets. In particular, the operations defined in [4] for bag union, intersection, and difference do not form a boolean algebra.

However, we can define the operations so that many of these set-theoretic identities (in particular, many of those that are relevant to query optimization) continue to hold for bags. We state and prove a number of such results, and derive some other properties of bag operations which have no set-theoretic counterpart. In addition, we discuss some of the properties of boolean algebras that fail for bags.

Vandenberg and DeWitt [19] develop a framework for algebraic query optimization for an object-oriented data model which supports bags. In this work, a large number of specific algebraic transformation rules are presented, and are used for query transformation. The emphasis of this work is on rules for manipulating the instances of types in a super-type/sub-type lattice, with complex type constructors. Union, intersection, and difference are defined for bags, and some transformation rules for these operations are given. Thus, our results complement this work.

The work of Mumick et al. [12, 13, 14] extends the usual set-theoretic operations to bags, and studies the semantics of recursion and aggregates with bag semantics. However, while the primitive operations, union, intersection, and difference are defined, no formal results regarding these operations are given.

On another track, Klausner et al. [8] take the view that, with regard to the relational model, support for bags can be handled by viewing a relation with duplicates as only a part of a (set-theoretic) relation, keeping invisible additional attributes which guarantee the uniqueness of all the tuples in the relation. While this technique may resolve at least some of the problems associated with bags for the relational model, it does not provide an adequate basis for query optimization for other data models which might be based on bags.

2 Algebraic Properties of Bags

2.1 Preliminaries

In this section, we formalize some of the algebraic properties of bags. In particular we define bag operations which correspond to the usual set operations of union, intersection, difference, and boolean selection, and investigate to what extent the typical algebraic identities of these set operations are obeyed. Various ideas for defining the semantics of these bag operations have been proposed [4, 7, 12, 13, 14, 19]. Our definitions agree with those in [4].

We begin with the premise that there is a countable set of primitive objects, **Obj**, and a set of atomic predicates, **Atom**, defined on **Obj**. Let the set **Pred** consist of the quantifier-free predicates that may be built up from **Atom**. That is, **Pred** is the smallest set of predicates which includes the atomic predicates, and is closed under the propositional connectives. We assume that there are no function symbols, so that the only terms on which predicates are evaluated are either variables or the elements of **Obj**.

We deliberately have not specified the particular atomic predicates in order that the results presented apply for any particular set **Atom** that might be chosen. For the remainder of the paper, we assume that, unless otherwise noted, any particular set or bag consists of elements chosen from **Obj**, and that any predicate is in **Pred**.

Note that the construction of a new predicate from atomic predicates may yield a result of greater arity. For example, if $\psi_1(x)$ is the atom $(x = 5)$, and $\psi_2(y)$ is the atom $(y = 6)$, then we have that $\varphi(x, y) = \psi_1(x) \vee \psi_2(y)$ is a predicate of arity 2. From now on we assume that all variables are typeless. For notational convenience we sometimes write $\psi(x)$ when x satisfies ψ, even if ψ has arity greater than 1. In case ψ has arity greater than 1, it is assumed that x is a tuple variable with the correct arity.

A bag is a collection of elements that may contain duplicates. We write $x \in B$ when the bag B contains x. To write down a representation of some finite bag, we use the notation [*list of elements*]. For example, [a,a,a,b,b] is the bag with 3 copies of a and 2 copies of b.

In [4], [7], and [13] it is noted that, while a set is characterized by its membership, a bag is characterized

by the multiplicity of its elements. We will use the infix notation from [7] for the multiplicity of elements of a bag. Specifically, for a bag B, we write $x \in\in B$ for the number of copies of x in B. For our purposes, it will suffice only to consider bags having countable multiplicities of elements. We write $x \in\in B = \omega$ if there are infinitely many copies of x in B. The rules for evaluating arithmetic expressions involving ω are given in the appendix.

For most cases, finite bags suffice. It is necessary to consider bags with countably infinite multiplicities so that the results presented are applicable to algebras containing finite bags with arbitrarily large finite multiplicities (where it is required that increasing sequences of finite bags have least upper bounds). Readers who so choose may ignore the existence of infinite bags for the remainder of the paper without significant loss of content.

2.2 Definitions and Properties of Bag Operations

We consider the set theoretic operations union, intersection, difference, as well as boolean selection, and use the familiar notations: $a \in A$ for membership, $A \subset B$ and $A \subseteq B$ for containment, $A \cup B$ for union, $A \cap B$ for intersection, $A \setminus B$ for difference, $-A$ for complement, $\mathcal{P}(A)$ for power-set, and $\sigma_\psi(A)$ for boolean selection.

With the exception of set complement, we can extend the above set operations to bags in a natural way. We will show that no suitable definition of bag complement exists. We are also interested in some operations whose restrictions to sets are not standard set operations.

First we define the notion of bag containment. Suppose A and B are bags.

$$A \subseteq B \iff^{\text{def}} \forall x \in A,\ (x \in\in A) \leq (x \in\in B).$$

$$A \subset B \iff^{\text{def}} A \subseteq B \text{ and } A \neq B.$$

In either case, we call A a *subbag* of B.

Theorem 1 *Given any set $\mathcal{B}$ of bags, $\subset$ is a partial order relation on $\mathcal{B}$.*

Proof: It is necessary to show that $\subset$ is irreflexive and transitive. Irreflexivity is immediate from the definition of $\subset$.

For transitivity, suppose $A \subset B \subset C$. Then we have that $\forall x \in A$, $(x \in\in A) \leq (x \in\in B)$, and $\forall x \in B$, $(x \in\in B) \leq (x \in\in C)$. Since the containments are strict, we have that $\exists x \in A$, $(x \in\in A) < (x \in\in B)$, and $\exists x \in B$, $(x \in\in B) < (x \in\in C)$. It follows that $\forall x \in A$, $(x \in\in A) \leq (x \in\in C)$ and that $\exists x \in A$, $(x \in\in A) < (x \in\in C)$, so $A \subset C$, which completes the proof.

∎

Now we define the other bag operations. The notions of *largest* and *smallest* in the following definitions refer to largest and smallest with respect to the partial order, $\subset$. Let A and B be bags, and ψ a predicate whose domain includes the elements of A.

$A \cup B \stackrel{\text{def}}{=}$ the smallest bag C such that

$$A \subseteq C \text{ and } B \subseteq C.$$

$A \cap B \stackrel{\text{def}}{=}$ the largest bag C such that

$$C \subseteq A \text{ and } C \subseteq B.$$

$\delta(B) \stackrel{\text{def}}{=} \{x | x \in B\}.$

$A \setminus B \stackrel{\text{def}}{=}$ the bag $C \subseteq A$ such that $\forall x \in \mathbf{Obj}$,

$$x \in\in C = \max((x \in\in A) - (x \in\in B), 0).$$

$\mathcal{P}(A) \stackrel{\text{def}}{=} \{B \mid B \subseteq A\}.$

$\sigma_\psi(A) \stackrel{\text{def}}{=}$ the bag C such that $\forall x \in \mathbf{Obj}$

$$x \in\in C = x \in\in A \text{ if } \psi(x)$$

$$x \in\in C = 0 \text{ otherwise.}$$

$A \sqcup B \stackrel{\text{def}}{=}$ the bag C such that $\forall x \in \mathbf{Obj}$

$$x \in\in C = (x \in\in A) + (x \in\in B).$$

We will call δ the duplicate elimination function, and use the name bag concatenation for $\sqcup$. The definitions for $\cup$ and $\cap$ will yield the usual set union and intersection when restricted to sets. However, we need to show that these operations are well-defined for bags. That is, we would like to show that there is a unique bag satisfying the definition of $A \cup B$ and $A \cap B$ for some fixed bags A and B. This is the content of the following result.

Theorem 2 *Let A, B, and C be bags.*

$$x \in\in A \cup B = \max(x \in\in A, x \in\in B) \quad (1)$$
$$x \in\in A \cap B = \min(x \in\in A, x \in\in B) \quad (2)$$
$$A \subseteq C \text{ and } B \subseteq C \Rightarrow A \cup B \subseteq C \quad (3)$$
$$C \subseteq A \text{ and } C \subseteq B \Rightarrow C \subseteq A \cap B \quad (4)$$

Proof: These follow from the definition of $\cup$ and $\cap$. For (1), let Y be the bag such that $\forall x \in \mathbf{Obj}$, $x \in\in Y = \max(x \in\in A, x \in\in B)$. It follows that $A \subseteq Y$ and $B \subseteq Y$. We need to show that Y is the smallest such bag. Suppose not. Then there is some bag X such that $A \subseteq X$, $B \subseteq X$, but it is not the case that $Y \subseteq X$. Thus $a \in\in X < a \in\in Y$ for some a. By the definition of Y, either $a \in\in X < a \in\in A$ or $a \in\in X < a \in\in B$, contradicting $A \subseteq X$ and $B \subseteq X$.

The proof of (2) is similar to (1). Let Y be the bag such that $\forall x \in \mathbf{Obj}$, $x \in\in Y = \min(x \in\in A, x \in\in B)$. It follows that $Y \subseteq A$ and $Y \subseteq B$. We need to show that Y is the largest such bag. Suppose not. Then there is some bag X such that $X \subseteq A$, $X \subseteq B$, but it is not the case that $X \subseteq Y$. Thus $a \in\in X > a \in\in Y$ for some a. From the definition of Y, we have that either $a \in\in X > a \in\in A$ or $a \in\in X > a \in\in B$, contradicting $X \subseteq A$ and $X \subseteq B$.

For (3), we have

$$\begin{aligned} &A \subseteq C \text{ and } B \subseteq C \\ &\quad \Rightarrow (x \in\in A) \leq (x \in\in C) \\ &\qquad \text{and } (x \in\in B) \leq (x \in\in C) \\ &\quad \Rightarrow \max(x \in\in A, x \in\in B) \leq (x \in\in C) \\ &\quad \Rightarrow A \cup B \subseteq C. \end{aligned}$$

For (4), we have

$$\begin{aligned} &C \subseteq A \text{ and } C \subseteq B \\ &\quad \Rightarrow (x \in\in C) \leq (x \in\in A) \\ &\qquad \text{and } (x \in\in C) \leq (x \in\in B) \\ &\quad \Rightarrow (x \in\in C) \leq \min(x \in\in A, x \in\in B) \\ &\quad \Rightarrow C \subseteq A \cap B, \end{aligned}$$

and we have proved the theorem.

■

The following result establishes for some of the bag operations, properties which play a central role in the design of a query language having bag semantics. Note that some of these identities are stated in [4] and [19].

Theorem 3 *Given any bags A and B, and predicates ψ and φ whose domains include $\delta(A)$, we have:*

$$\sigma_{\psi \vee \varphi}(A) = \sigma_{\psi}(A) \cup \sigma_{\varphi}(A) \quad (1)$$
$$\sigma_{\psi \wedge \varphi}(A) = \sigma_{\psi}(A) \cap \sigma_{\varphi}(A) \quad (2)$$
$$\sigma_{\neg \varphi}(A) = A \setminus \sigma_{\varphi}(A) \quad (3)$$
$$\delta(\sigma_{\varphi}(A)) = \sigma_{\varphi}(\delta(A)) \quad (4)$$
$$x \in\in \sigma_{\varphi}(A) = (x \in\in A) * (x \in\in \sigma_{\varphi}(\delta(A))) \quad (5)$$
$$A \cup B = (A \setminus B) \sqcup B \quad (6)$$
$$A \cup B = (A \sqcup B) \setminus (A \cap B) \quad (7)$$
$$A \cap B = A \setminus (A \setminus B) \quad (8)$$
$$A \cap B = (A \sqcup B) \setminus (A \cup B) \quad (9)$$

Proof: The proofs of these identities follow from simple arithmetic on the bag multiplicity functions.

■

Now we discuss the importance of the above properties. First note that (1), (2), (3) above imply that the semantics of $\vee$, $\wedge$, and $\neg$ with respect to boolean selection correspond to $\cup$, $\cap$, and $\setminus$, respectively.

Item (5) is a formal way of saying that the bag selection operator has what we will refer to as *all-or-nothing* semantics. In accord with the definition on page 3, if B is a bag, then $\sigma_{\varphi}(B)$ will select all elements of B which satisfy the predicate φ. If some $x \in B$ does not satisfy φ, then it is not selected. Either *all* copies of some $x \in B$ are selected, or *none* are.

Finally, (6) through (9) demonstrate that $\sqcup$ and $\setminus$ are sufficient to define $\cup$ and $\cap$. It is natural to ask the converse question, that is, can $\sqcup$ or $\setminus$ be constructed out of the remaining operations? The following two results give this a negative answer.

Theorem 4 *There is no bag expression involving the symbols $\{A, B, \cup, \cap, \setminus\}$ which is equivalent to $A \sqcup B$ for all bags A and B.*

Proof: We first claim that if E is the value of any well-formed bag expression constructed from the symbols $\{A,B,\cup,\cap,\setminus\}$ then $E \subseteq A \cup B$. This claim is established by induction on the number of $\cup,\cap$, or $\setminus$ operators occurring in E. As a basis, suppose E has 0 of the $\cup,\cap$, or $\setminus$ operators. Then either $E = A$ or $E = B$, and certainly $E \subseteq A \cup B$.

For the inductive step, let E be an expression having exactly n operators, where $n > 0$, and suppose that the claim is true for expressions having fewer than n

operators. Then E has one of the following forms:

$$E = E_1 \cup E_2 \quad (1)$$
$$E = E_1 \cap E_2 \quad (2)$$
$$E = E_1 \setminus E_2. \quad (3)$$

The number of operators in either E_1 or E_2 is less than n, so, by the induction hypothesis, $E_1 \subseteq A \cup B$ and $E_2 \subseteq A \cup B$. But then, it immediately follows that $E_1 \cup E_2 \subseteq A \cup B$, $E_1 \cap E_2 \subseteq A \cup B$, and $E_1 \setminus E_2 \subseteq A \cup B$, which establishes the claim.

To complete the proof, suppose, to the contrary, that there was an expression, E, constructed from the symbols {A,B,∪,∩,\}, which was equivalent to $A \sqcup B$ for all bags A and B. Then $E = A \sqcup B$, and by the claim, $A \sqcup B \subseteq A \cup B$. This leads to a contradiction. Choose any bags A and B such that $A \cap B \neq \emptyset$, in which case $A \sqcup B \subseteq A \cup B$ is false, and the theorem follows.

∎

Theorem 5 *There is no bag expression involving the symbols $\{A,B,\cup,\cap,\sqcup\}$ which is equivalent to $A \setminus B$ for all bags A and B.*

Proof: This result is established with a proof very similar to that for Theorem 4. As before, we begin with an auxiliary claim. If E is the value of any well-formed bag expression constructed from the symbols {A,B,∪,∩,⊔} then $A \cap B \subseteq E$. This claim is established by induction on the number of ∪,∩, or ⊔ operators occurring in E. As a basis, suppose E has 0 of the ∪,∩, or ⊔ operators. Then either $E = A$ or $E = B$, and certainly $A \cap B \subseteq E$.

For the inductive step, let E be an expression having exactly n operators, where $n > 0$, and suppose that the claim is true for expressions having fewer than n operators. Then E has one of the following forms:

$$E = E_1 \cup E_2 \quad (1)$$
$$E = E_1 \cap E_2 \quad (2)$$
$$E = E_1 \sqcup E_2. \quad (3)$$

The number of operators in either E_1 or E_2 is less than n, so, by the induction hypothesis, $A \cap B \subseteq E_1$ and $A \cap B \subseteq E_2$. But then, it immediately follows that $A \cap B \subseteq E_1 \cup E_2$, $A \cap B \subseteq E_1 \cap E_2$, and $A \cap B \subseteq E_1 \sqcup E_2$, which establishes the claim.

To complete the proof, suppose, to the contrary, that there was an expression, E, constructed from the symbols {A,B,∪,∩,⊔}, which was equivalent to $A \setminus B$ for every bag A and B. Then $E = A \setminus B$, and by the claim, $A \cap B \subseteq A \setminus B$. If $A = [x]$ and $B = [x]$, then $[x] = A \cap B \subseteq A \setminus B = \emptyset$, a contradiction, and the theorem follows.

∎

The implication of these results is that ⊔ and \ are strictly more general operations than ∪ and ∩. Note that some query languages, for example SQL [2], use the keyword 'union' for the operator ⊔. We have chosen to call the ∪ operation 'union' since this operator shares the same algebraic properties as the standard set-theoretic union, and in fact is the usual set-theoretic union when restricted to sets. However, Theorems 4 and 5 imply that a query language which supports bag concatenation (the ⊔ operator) as well as union, intersection, and difference, is more powerful than a language supporting only union, intersection, and difference.

Now we discuss bag difference. The bag difference operator is just the usual notion of set difference when restricted to sets. However, taking any bag that is not a set as a universe, this operator does *not* induce a complement operator relative to this universe. In fact, we can demonstrate a stronger result, that no such operator exists. Thus, looking for a different semantics for bag difference cannot resolve this problem.

The properties that a complement operator must satisfy are a subset of the axioms of a boolean algebra. If we restrict the axioms of a boolean algebra to those in which complement is not referenced, then we have the axioms of a distributive lattice. That is, if $(\mathcal{P}(U),\cup,\cap,-)$ is a boolean algebra (with unit U, and zero $\emptyset$), then $(\mathcal{P}(U),\cup,\cap)$ is a distributive lattice.

Given a bag U, we would like to have some suitable unary operator, $-$, such that $(\mathcal{P}(U),\cup,\cap,-)$ is a boolean algebra, with unit U, and zero $\emptyset$. What we can show is that, while a distributive lattice structure is available on $\mathcal{P}(U)$, there is no unary operator which extends the structure to a boolean algebra. We establish these facts in the following two theorems.

Theorem 6 *$(\mathcal{P}(U),\cup,\cap)$ is a distributive lattice induced by the partial order $\subset$. We will call it the subbag lattice.*

Proof: The operations ∪ and ∩ were defined, respectively, as the least upper bound and greatest lower bound of the partial order, $\subset$. Thus, to establish that, given some arbitrary bag, U, $(\mathcal{P}(U),\cup,\cap)$ is the lattice

induced by the partial order $\subset$, it suffices to show that $\mathcal{P}(U)$ is closed with respect to $\cup$ and $\cap$. This was already shown in Theorem 2, taking C in items (3) and (4) to be U.

It remains to be shown that $\cup$ distributes over $\cap$, and $\cap$ distributes over $\cup$, which follow from simple arithmetic. Fix x. Then:

$$x \in\in A \cup (B \cap C)$$
$$= \max(x \in\in A, \min(x \in\in B, x \in\in C))$$
$$= \min(\max(x \in\in A, x \in\in B), \max(x \in\in A, x \in\in C))$$
$$= x \in\in (A \cup B) \cap (A \cup C).$$

And similarly, we have:

$$x \in\in A \cap (B \cup C)$$
$$= \min(x \in\in A, \max(x \in\in B, x \in\in C)$$
$$= \max(\min(x \in\in A, x \in\in B), \min(x \in\in A, x \in\in C))$$
$$= x \in\in (A \cap B) \cup (A \cap C).$$

■

Theorem 7 *If U is a bag which is not a set, then there is no unary operator, $-$, that can be defined on $\mathcal{P}(U)$ such that $(\mathcal{P}(U),\cup,\cap,-)$ is a boolean algebra.*

Proof: The axioms that a unary operator, $-$, must satisfy to be a complement are:

$$\forall A \in \mathcal{P}(U), \quad A \cup (-A) = U$$
$$\forall A \in \mathcal{P}(U), \quad A \cap (-A) = \emptyset.$$

Let U be a bag which is not a set. Then there is some $x \in U$ such that $x \in\in U > 1$. Fix $A \subset U$ an arbitrary bag such that $0 < (x \in\in A) < (x \in\in U)$. ($[x] \in \mathcal{P}(U)$ is such a bag, so such A always exist).

Now assume, to the contrary of the statement of the theorem, that there is a unary operator, $-$, satisfying the above axioms for a complement. Applying the axioms to A, we have: $A \cup (-A) = U$ which implies that $(x \in\in (-A)) = (x \in\in U)$. Similarly, $A \cap (-A) = \emptyset$ implies that $(x \in\in (-A)) = 0$. Thus $(x \in\in U) = 0$, contradicting that $(x \in\in U) > 1$.

■

Since $(\mathcal{P}(U),\cup,\cap)$ is a distributive lattice, all of the usual algebraic properties of $\cup$ and $\cap$ are satisfied by their extensions to bags. However, since it is observed that bag difference does not generate a proper complement operator, it must be that some of the properties of a boolean algebra fail for bags. We will first give an example of an identity for sets that fails for bags, and then try to understand when bag difference behaves like set difference.

Suppose A, B, and C are sets. Then we have the following identity.

$$A \setminus (B \cup C) = A \cap \overline{(B \cup C)} = A \cap (\overline{B} \cap \overline{C})$$
$$= (A \cap \overline{B}) \cap \overline{C} = (A \setminus B) \setminus C.$$

Here we have used the equivalence of set difference and the intersection with the complement, DeMorgan's law, and the associativity of intersection. The point is, that the above identities hold for any sets A, B, and C.

However, it may be that $A \setminus (B \cup C) \neq (A \setminus B) \setminus C$ when A, B, and C are bags. Here is a simple example.

$$A = [x, x, x, x]$$
$$B = [x, x]$$
$$C = [x]$$
$$A \setminus (B \cup C) = [x, x]$$
$$(A \setminus B) \setminus C = [x].$$

We would like to know which transformations of set expressions are valid for bags, since these are useful for query optimization. What we will show is that expressions involving $\cup$, $\cap$, and $\setminus$ exhibit the same behavior for bags as for sets either when these operators arise from selection predicates, or when the operands are bags which are formed by applying a boolean selection to some universal bag.

Looking back at the above example, where we have $A = [x, x, x, x]$, $B = [x, x]$ and $C = [x]$, we see that there is no predicate ψ such that $B = \sigma_\psi(A)$. This is because of the all-or-nothing semantics— it is not possible to select two copies of x. Either all four copies are selected, or none are selected.

The following result is suggestive.

Theorem 8 *Let E_1 and E_2 be two bag operators constructed as a composition of σ operators. That is,*

$$E_1 = \sigma_{\varphi_1} \circ \sigma_{\varphi_2} \circ \cdots \circ \sigma_{\varphi_n}$$
$$E_2 = \sigma_{\psi_1} \circ \sigma_{\psi_2} \circ \cdots \circ \sigma_{\psi_k}$$

Then, if for every set A, $E_1(A) = E_2(A)$, then also $E_1(B) = E_2(B)$ for every bag B.

Proof: We establish the all-or-nothing property of expressions having several composed selections by induction on the length of the expression. That is, if $E = \sigma_{\phi_1} \circ \sigma_{\phi_2} \circ \cdots \circ \sigma_{\phi_m}$ is such an expression, then we claim that for any bag, B, we have:

$$x \in\in E(B) \;=\; (x \in\in B) * (x \in\in E(\delta(B))).$$

As a basis for the induction take $m = 1$. Then the desired property was established in Theorem 3, item (5).

For the inductive step, fix $m > 1$ and suppose that the result holds for expressions having length $m - 1$. Then if $E = \sigma_{\phi_1} \circ \sigma_{\phi_2} \circ \cdots \circ \sigma_{\phi_m}$, we have that:

$$\begin{aligned}
x \in\in E(B) &= x \in\in \sigma_{\phi_1} \circ \sigma_{\phi_2} \circ \cdots \circ \sigma_{\phi_m}(B) \\
&= x \in\in \sigma_{\phi_1} \circ \sigma_{\phi_2} \circ \cdots \circ \sigma_{\phi_{m-1}}(\sigma_{\phi_m}(B)) \\
&= (x \in\in \sigma_{\phi_m}(B)) * (x \in\in \sigma_{\phi_1} \circ \cdots \circ \sigma_{\phi_{m-1}}(\delta(\sigma_{\phi_m} B))) \\
&= (x \in\in \sigma_{\phi_m}(B)) * (x \in\in \sigma_{\phi_1} \circ \cdots \circ \sigma_{\phi_{m-1}}(\sigma_{\phi_m} \delta(B))) \\
&= (x \in\in \sigma_{\phi_m}(B)) * (x \in\in E(\delta(B))) \\
&= (x \in\in B) * (x \in\in \sigma_{\phi_m}(\delta(B))) * (x \in\in E(\delta(B))) \\
&= (x \in\in B) * (x \in\in E(\delta(B))),
\end{aligned}$$

which establishes the claim.

Now if E_1 and E_2 are as in the statement of the theorem, then for any bag B,

$$\begin{aligned}
x \in\in E_1(B) &= (x \in\in B) * (x \in\in E_1(\delta(B))) \\
x \in\in E_2(B) &= (x \in\in B) * (x \in\in E_2(\delta(B))).
\end{aligned}$$

But $\delta(B)$ is a set, so $E_1(\delta(B)) = E_2(\delta(B))$, and the theorem follows.

∎

Theorem 8 states that, with respect to selection operations, bags behave like sets. A selection expression involving bags may be transformed into a logically equivalent expression either by cascading the predicates, or by substituting logically equivalent predicates. In fact, we can demonstrate a more basic result— that the operations of union, intersection, and difference for bags give rise to a boolean algebra when the operations arise from boolean selection.

Given a bag, U, we will define a structure, $\mathcal{S}(U)$, analogous to the power set of U, but containing only subbags obtained by applying a selection operator to U.

$$\mathcal{S}(U) \stackrel{\text{def}}{=} \{A \subseteq U : (\exists \psi \in \mathbf{Pred})(A = \sigma_\psi(U)\}.$$

Then we would like to define a boolean algebra structure on $\mathcal{S}(U)$. For $A \in \mathcal{S}(U)$, let $(-A) \stackrel{\text{def}}{=} U \setminus A$. On $\mathcal{S}(U)$, bag difference does indeed generate a proper complement operator, as the following result elucidates.

Theorem 9 *$(\mathcal{S}(U),\cup,\cap,-)$ is a boolean algebra.*

Proof: Rather than checking all of the axioms of a boolean algebra, it will suffice to show that $(\mathcal{S}(U),\cup,\cap)$ is a sub-lattice of $(\mathcal{P}(U),\cup,\cap)$, and verify the axioms for complement. To this end, it suffices to show that $\mathcal{S}(U)$ is closed under $\cup$, $\cap$, and $-$, and that $-$ satisfies the axioms for a complement operator.

Let A and B be subbags of U such that $A = \sigma_\psi(U)$ for some ψ, and $B = \sigma_\varphi(U)$ for some φ. Then:

$$\begin{aligned}
A \cup B &= \sigma_\psi(U) \cup \sigma_\varphi(U) = \sigma_{\psi \vee \varphi}(U) \in \mathcal{S}(U) \\
A \cap B &= \sigma_\psi(U) \cap \sigma_\varphi(U) = \sigma_{\psi \wedge \varphi}(U) \in \mathcal{S}(U) \\
(-A) &= U \setminus A = U \setminus \sigma_\psi(U) = \sigma_{\neg\psi}(U) \in \mathcal{S}(U).
\end{aligned}$$

Now checking the complement axioms, we see that:

$$\begin{aligned}
A \cup (-A) &= \sigma_\psi(U) \cup \sigma_{\neg\psi}(U) = \sigma_{\psi \vee \neg\psi}(U) = U \\
A \cap (-A) &= \sigma_\psi(U) \cap \sigma_{\neg\psi}(U) = \sigma_{\psi \wedge \neg\psi}(U) = \emptyset,
\end{aligned}$$

and we have established the theorem.

∎

3 Conclusions

In summary, we have extended union, intersection, difference, and boolean selection to bags, giving them a semantics which agrees with the usual set-theoretic semantics when the operands are sets. In addition, we have defined and studied the notion of bag concatenation.

We have observed that union and intersection for bags form a distributive lattice. Thus, the usual algebraic properties of these set operations continue to hold for bags. Further, union and intersection correspond, respectively, to disjunction and conjunction for boolean selection.

We have also shown that there is no unary operator available to play the role of a complement, so that union, intersection and the complement would form a boolean algebra. However, we have defined a

bag difference operator that conforms to set difference when applied to sets, and corresponds to negation for boolean selection over bags.

While showing that some of the algebraic properties of sets fail for bags, we have shown that the collection of bags which are the result of a boolean selection applied to some universal bag forms a boolean algebra with respect to union, intersection, and difference. In this case, the usual algebraic properties for sets will hold, and in particular, the algebraic transformations that are applied to sets for query optimization continue to be valid.

4 Appendix

Here we extend the usual arithmetic functions to operate on ω. For any natural number n,

$$\begin{aligned}
\omega + n &= n + \omega = \omega \\
\omega - n &= \omega \\
\omega + \omega &= \omega \\
\omega - \omega &= 0 \\
\omega * n &= n * \omega = \omega \\
\max(\omega, n) &= \max(n, \omega) = \omega \\
\min(\omega, n) &= \min(n, \omega) = n \\
\max(\omega, \omega) &= \min(\omega, \omega) = \omega
\end{aligned}$$

5 Acknowledgements

The development of this material was aided by a number of conversations with Peter Schauble, who first pointed out to me that there should not be a boolean algebra structure available for bags. I am grateful to Bill Kent, Ravi Krishnamurthy, Scott Vandenberg, and the anonymous referees for many insightful suggestions that have led to improvements in this paper.

References

[1] Carey, M., DeWitt, D., Vandenberg, S., 'A Data Model and Query Language for EXODUS', *Proc. ACM SIGMOD 1988 Int'l Conf. on Management of Data*, Chicago, IL, May 1988.

[2] Chamberlain, D., et al., 'SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control', *IBM Journal of Research and Development*, 20:6, November 1976.

[3] Codd, E.F., 'A Relational Model for Large Shared Data Banks', *CACM* 13:6, June 1970.

[4] Dayal, U., Goodman, N., Katz, R.H., 'An Extended Relational Algebra with Control Over Duplicate Elimination', *Proc. ACM Symposium on Principles of Database Systems*, Los Angeles, CA, March 1982.

[5] Fishman, D.H. et al., 'Iris: An Object-Oriented Database Management System', *ACM Trans. Office Information Systems*, 5:1, January 1987.

[6] Hammer, M., McLeod, D., 'Database Description with SDM: A Semantic Database Model', *ACM Trans. on Database Systems*, 6:3, September 1981.

[7] Kent, W., 'Profile Functions and Bag Theory', HPL-SAL-89-19, Hewlett-Packard Laboratories, Palo Alto, CA, January 1989.

[8] Klausner, A., Goodman, N., 'Multirelations– Semantics and Languages', *Proc. 11th Int'l Conf. on Very Large Databases*, Stockholm, Sweden, August 1985.

[9] Klug, A., 'Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions', *JACM*, 29:3, July 1982.

[10] Manola, F., Dayal, U., 'PDM: An Objected-Oriented Data Model', *Proc. Int'l Workshop on Object-Oriented Databases Systems*, Asilomar, CA, September 1986.

[11] Monk, J.D., Bonnett, R., ed., *Handbook of Boolean Algebras*, North-Holland, Amsterdam 1989.

[12] Mumick, I.S., et al., 'Magic Conditions', *Proc. 8th ACM Symposium on Principles of Database Systems*, Philadelphia, PA, 1989.

[13] Mumick, I.S., et al., 'Magic is Relevant', *Proc. ACM SIGMOD 1990 Int'l Conf. on Management of Data*, Atlantic City, NJ, May 1990.

[14] Mumick, I.S., et al., 'The Magic of Duplicates and Aggregates', *Proc. 16th Int'l Conf. on Very Large Databases*, Brisbane, Australia, August 1990.

[15] Schwartz, P., et al., 'Extensibility in the Starburst Database System', *Proc. Int'l Workshop on Object-Oriented Database Systems*, Pacific Grove, CA, 1986.

[16] Shipman, D., 'The Functional Data Model and the Data Language DAPLEX', *ACM Trans. on Database Systems*, 6:1, March 1981.

[17] Stonebraker, M., et al., 'The Design and Implementation of INGRES', *ACM Trans. on Database Systems*, 12:3, September 1976.

[18] Ullman, J.D., *Principles of Database and Knowledge-Base Systems*, Computer Science Press, Rockville, MD 1989.

[19] Vandenberg, S.L., DeWitt, D.J., 'Algebraic Support for Complex Objects with Arrays, Identity, and Inheritance', *Proc. ACM SIGMOD 1991 Int'l Conf. on Management of Data*, Denver, CO, May 1991.