Integration of Semistructured Data with Partial and Inconsistent Information

Mengchi Liu
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
mliu@cs.uregina.ca

Tok Wang Ling
School of Computing
National University of Singapore
Lower Kent Ridge Road
Singapore 119260
lingtw@comp.nus.edu.sg

Tao Guan
Department of Computer Science
University of Regina
Regina, Saskatchewan
Canada S4S 0A2
guan@cs.uregina.ca

Abstract

Data integration of several sources has gained considerable attentions with the recent popularity of the Web. In the real world, some information may be missing (i.e., partial) and some may be inconsistent from several sources. How to obtain information as complete as possible and detect inconsistency from these sources is thus an interesting question. Most existing work uses a simple graph-based or tree-based semistructured data model to represent heterogeneous data coming from various sites, which fail to account for the existence of partial and inconsistent information. In this paper, we redefine the notion of semistructured objects to reflect the existence of partial and inconsistent information and study how to integrate such objects spread in various sources and check consistency in the meantime. We propose a new operator integration for this purpose and discuss its semantic properties.

1 Introduction

As the amount of data available on-line has increased dramatically, the need to integrate a wide variety of data has become more and more important. Several semistructured data models such as OEM [2] and labeled tree model [8] have been proposed to represent such heterogeneous data. How to integrate semistructured data coming from various sources has also received considerable attentions [3, 6, 12, 9, 27, 38]. However, most existing work uses a simple graph-based or tree-based semistructured data model to represent heterogeneous data coming from various sources, which fails to account for the existence of partial and inconsistent information. In practice, the information about real world objects may be imprecise, incomplete, or even wrong. Many null/unknown values or inconsistent values exist in the data which come from various sources. For example, most academic people should have a Bibtex database to keep references. While two or more person work together on a paper, an immediate problem is how to merge their Bibtex databases. Although all Bibtex databases have the similar structure, the values in these databases may be partial or inconsistent (conflicted). The typical case is the *authors* of a paper. Someone likes to list all authors in full names, but others may just indicate the first one or two authors. Even for one author, the order of first name (or initial) and last name may be different. Furthermore, partial data for the same record is probably missed or inconsistent, i.e. *page number, published year or the address of publisher.* Therefore, how to obtain information as complete as possible from these sources is an interesting question. For another example, we may find several web pages containing information about the same object, such as a place, a person, an organization, or a company. It is very useful to combine the information spread in several sources to obtain a comprehensive description of the object.

In the past several years, a sub-problem, that is, integrating data with partial and complete tuples and/or partial sets, has been investigated in depth in the context of relational and complex object databases [5, 7, 11, 14, 20, 21, 22, 28, 29, 30, 31, 39, 40]. Specific operators such as union [5] and join [7] are introduced to integrate partial information. However, these works focus on typed data and support homogeneous sets and tuples. Thus, it is difficult to apply them directly on semistructured data, where although the data may have some structure, the structure is not as rigid, regular, or complete as that required by traditional database systems.

In this paper, we first redefine the notion of semistructured objects to reflect the existence of partial and inconsistent information. We then study how to integrate such objects spread in various sources and check consistency in the meantime. Instead of considering the problem of entity identification (i.e., matching objects across sources) addressed in [12, 15, 19, 32]), we focus our study on how to integrate different semistructured data which have been represented in our framework. We propose a new operator *integration* for this purpose and discuss its semantic properties.

1.1 Related Work

There has been a lot of work on the integration of heterogeneous data sources, such as multiple databases or data sources on the web [5, 7, 14, 20, 21, 22, 23, 28, 29, 30, 31, 37, 39, 40, 10]. In particular, there are two questions closely related to our study. One is how to decide that two objects mentioned in two different sources refer to the same entity in the world. Another is how to reconcile partial or inconsistent data values from sources referring to the same entity.

Most systems deal with the first problem using domain specific heuristics [15, 19]. Two notable exceptions are the Smith-Waterman edit distance adopted by Monge and Elkan [36] and statistical information retrieval in [12]. Both are domain-independent.

The problem of reconciling inconsistent data was addressed in the literature by the theory of probability [40] and the evidential theory [32]. An early work [14] introduced a concept of *partial values* to handle it, where a partial value is an interval or a finite set of *possible* values such that exactly one of the values in this set is the true value of the partial value. The approach was further extended in [40] with probabilistic partial values (i.e., partial values with the attached probability of occurrence). In addition, the method proposed in [32] utilizes the relational model extended to incorporate the Dempster-Shafer uncertainty management mechanism. The combination rule from this theory is used to reconcile inconsistent values; that is, to assign appropriate degree of belief to the possible values. In [37], information about the quality of data (i.e. soundness and completeness) in the sources is proposed to use to reconcile inconsistent answers that may result from query processing involving independent data sources. However, partial values (i.e. some attribute values are missing) are not addressed in these work.

The problem of describing completeness of data sources and using this information for query processing is addressed in [26]. An approximating algorithm was proposed in [31] to answer queries for which a precise answer cannot be found. These work are related to ours, but focus on different topics.

Other relevant work include semistructured data [2], unstructured data [8] and data integration [9, 17, 20], and web query (or data manipulation) languages such as W3QS [13], WebSQL [35] and WebLog [24], WebOQL [4] and StruQL [16]).

The rest of the paper is organized as follows. Section 2 defines semistructured objects. Section 3 discusses how to integrate semistructured objects and the semantic properties. Section 4 summarizes and points out further research issues.

2 Semistructured Objects

In this section, we establish terminology for the concepts including *objects* and *semistructured objects.* We assume the existence of a set $\mathcal{A}$ of attribute names, a set $\mathcal{M}$ of markers, and a set $\mathcal{U}$ of constants such that $\mathcal{M}$ and $\mathcal{A} \cup \mathcal{U}$ are disjointed. The notion of *objects* is defined as follows:

(1) Constants in $\mathcal{U}$ are objects called *atomic* objects.
(2) Markers in $\mathcal{M}$ are objects called *marker objects.*
(3) There is a special object $\perp$.
(4) If O_1, ..., O_n are distinct objects other than $\perp$, then $O_1|...|O_n$ is an object called *or-value* object.
(5) If $O_1, ..., O_n, (n > 0)$ are distinct objects, then $\langle O_1, ..., O_n \rangle$ is an object called *partial set.*
(6) If $O_1, ..., O_n, (n \geq 0)$ are distinct objects, then $\{O_1, ..., O_n\}$ is an object called *complete set.*
(7) If $O_1, ..., O_n$ are objects and $A_1, ..., A_n$ are distinct attribute names, then $O = [A_1 \Rightarrow O_1, ..., A_n \Rightarrow O_n]$ is an object called *tuple.* We denote O_i by $O.A_i$. We also assume that $O.A = \perp$ for an attribute A not in $\{A_1, ..., A_n\}$.

(8) If $m_1, ..., m_n \in \mathcal{M}$ are markers with $n > 0$ and O is an object, then $m_1|...|m_n : O$ is an object called *marked object.*

We use $\perp$ for null/unknown object. For example, in a tuple representing a person, if the age of the person is unknown, then we use $[..., age \Rightarrow \perp, ...]$.

As we are dealing with the integration of semistructured objects from different sources, it is possible that we have conflicting information. In this case, we use *or-value* object to record the conflicting result. For example, the or-value object 21|22 in the tuple object [..., age ⇒ 21|22, ...] implies the age is 21 or 22 as there is a conflict right now and it is not clear that which one of these two values is correct. The or-values are used to record conflicting information. We treat an or-value as a set so that any combination of $O_1, ..., O_n$ is the same as $O_1|...|O_n$.

The *markers* are used to identify/refer to an object uniquely. They are similar to *object identifier* in OEM (Object Exchange Model) [2], but different in nature. An *object identifier* is attached to each object in OEM, even for constants. In contrast, *markers* in our framework can be used to identify complex objects. For example, in a Bibtex database, markers correspond to the keys [25]; in a web page, markers correspond to URLs. See Example 1.

Besides null/unknown and inconsistent values, it is quite common that partial rather than complete information is provided for a set. For example, in a Bibtex file, one may only give partial authorship such as *"Bob and others"* [18]. In this case, the set containing *"Bob"* is partial and should be represented with $\langle$"*Bob*"$\rangle$ to indicate that the set only provide partial authorship. On the other hand, if we know the complete authorship such as *"Bob and Tom"*, then the set contains *"Bob"* and *"Tom"* is complete and should be represented in our database as {"*Bob*", "*Tom*"}. The notions of partial and complete set were first introduced in ROL [33] and later extended in Relationlog [34]. They are used to represent open and closed world assumption on sets in a database.

A marked object $m_1|...|m_n : O$ with $n > 1$ means that the object O is obtained from an integration of several marked objects.

A *semistructured object* is an object.

For example, a Bibtex file can be viewed as a set of semistructured objects; see Example 1. A web page can be viewed as a semistructured objects; see Example 2.

Example 1 Consider the following bib file with three entries:

```
@InBook{Bob,
  author   = "Bob and others",
  title    = "The Oracle System",
  chapter  = 1,
  crossref = DB}

@Book{DB,
  booktitle = "Database Systems",
  publisher = "Addison Wesley",
  editor    = "John",
  year      = 1985}

@Article{B80,
  author   = "Bob",
  title    = "Relational DB",
  journal  = "JACM",
  volume   = 10,
  number   = 6,
  year     = 1980,
```

They can be represented as two semistructured objects with the markers *Bob* and *DB* as follows:

$$\begin{array}{llll}
[InBook \Rightarrow Bob : [author & \Rightarrow \langle\text{“}Bob\text{”}\rangle, & & \\
 & title & \Rightarrow \text{“}The\ Oracle\ System\text{”}, & \\
 & chapter & \Rightarrow 1, & \\
 & crossref & \Rightarrow DB &]]
\end{array}$$

$$\begin{array}{llll}
[Book \Rightarrow DB : [booktitle & \Rightarrow \text{“}Database\ Systems\text{”}, & & \\
 & publisher & \Rightarrow \text{“}Addison\ Wesley\text{”}, & \\
 & editor & \Rightarrow \{\text{“}John\text{”}\}, & \\
 & year & \Rightarrow 1985 &]]
\end{array}$$

$$\begin{array}{llll}
[Article \Rightarrow B80 : [author & \Rightarrow \{\text{“}Bob\text{”}\}, & & \\
 & title & \Rightarrow \text{“}Relational\ DB\text{”}, & \\
 & journal & \Rightarrow \{\text{“}JACM\text{”}\}, & \\
 & volume & \Rightarrow 10, & \\
 & number & \Rightarrow 1, & \\
 & year & \Rightarrow 1980 &]]
\end{array}$$

Example 2 Consider the following simplified web page:

```
<html>
<head>
<title>CSDept</title>
</head>

<body>
CSDept<br>
<a href="faculty.html"> Faculty </a>
<a href="staff.html">   Staff   </a>
<a href="students.html">Students</a>
</body>
</html>
```

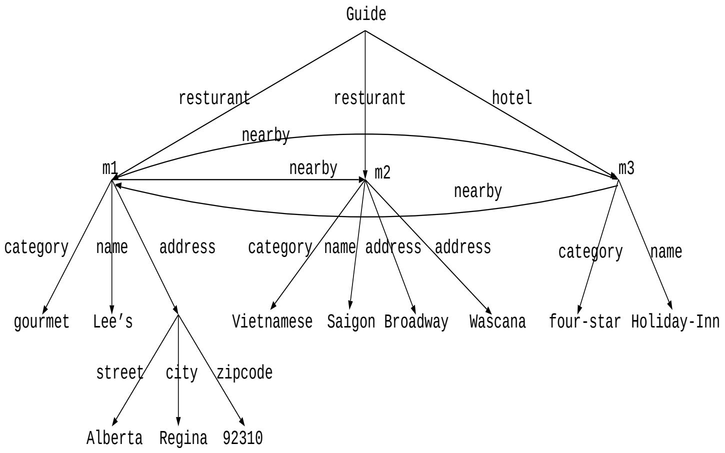
This web page can be represented as a semistructured object in our framework as follows:

$$\begin{array}{ll} [CSDept \Rightarrow [Faculty & \Rightarrow faculty.html, \\ Staff & \Rightarrow staff.html, \\ Students & \Rightarrow students.html]] \end{array}$$

where faculty.html, staff.html, students.html are markers.

Note that web pages and Bibtex files differs significantly. For a web page, there are markers (URLs) but no marked objects as we have to expand the markers to obtain the corresponding objects. For a Bibtex file, however, there are marked objects.

Example 3 Consider the following labeled graph based on the one taken from [1]:



This graph can be represented as a set of semistructured objects in our framework as follows:

$$\begin{array}{llll} [Guide \Rightarrow & [\ restaurant & \Rightarrow & \{m_1, m_2\}, \\ & \ \ hotel & \Rightarrow & \quad m_3 \quad]] \end{array}$$

$$\begin{array}{lll} m_1 : [\ category & \Rightarrow & gourmet, \\ \quad\ \ name & \Rightarrow & Lee's, \\ \quad\ \ address & \Rightarrow & [street \Rightarrow Alberta, \\ & & \ city \Rightarrow Regina, \\ & & \ zipcode \Rightarrow 92310], \\ \quad\ \ nearby & \Rightarrow & \{m_2, m_3\} \qquad\qquad], \end{array}$$

$$\begin{array}{lll} m_2 : [\ category & \Rightarrow & Vietnamese, \\ \quad\ \ name & \Rightarrow & Saigon, \\ \quad\ \ address & \Rightarrow & \{Broadway, Wascana\}], \end{array}$$

$$\begin{array}{lll} m_3 : [\ category & \Rightarrow & four\text{-}star, \\ \quad\ \ name & \Rightarrow & Holiday\text{-}Inn, \\ \quad\ \ nearby & \Rightarrow & m_1 \qquad\qquad\qquad] \end{array}$$

where the leaf nodes represent constants, non-leave nodes represent tuples or sets, and edges represent attributes in our framework. The markers m_1, m_2 and m_3 are introduced to identify the objects in cycles.

Our semistructured objects can capture more information than the existing semistructured data models such as OEM [2] and labeled tree model [8] since null/unknown, *or-value*, *partial* and *complete* set objects are supported.

3 Integrating Semistructured Objects

In this section, we discuss how to integrate semistructured objects. Consider the following semistructured objects that represents simplified Bibtex items in two different bib files. Note that strings are represented without quotes when there is no ambiguity for simplicity.

$$\begin{array}{ll} [article \Rightarrow B80 : [title & \Rightarrow Oracle, \\ \qquad\qquad\qquad\quad author & \Rightarrow Bob, \\ \qquad\qquad\qquad\quad year & \Rightarrow 1980 \quad]] \end{array}$$

$$\begin{array}{ll} [article \Rightarrow B82 : [title & \Rightarrow Oracle, \\ \qquad\qquad\qquad\quad year & \Rightarrow 1980, \\ \qquad\qquad\qquad\quad journal & \Rightarrow JACM]] \end{array}$$

The first nested tuple has null value for the attribute *journal* whereas the second has null value for the attribute *author*. They have different markers. Let us assume that articles can be identified by their title. Then the above two objects represent different portion of the same object and can be integrated to generate the following tuple:

$$\begin{array}{ll} [article \Rightarrow B80|B82 : [title & \Rightarrow Oracle, \\ \qquad\qquad\qquad\qquad\quad author & \Rightarrow Bob, \\ \qquad\qquad\qquad\qquad\quad year & \Rightarrow 1980, \\ \qquad\qquad\qquad\qquad\quad journal & \Rightarrow JACM]] \end{array}$$

where $B80|B82$ means that the two Bibtex terms from two different bib files have different markers that refer to the same article. Thus, more complete information of the article is obtained by integrating the two bib files.

In order to formalize the integration of semistructured objects, we first introduce the following notions.

An object O is *less or equal informative* than an object O', denoted by $O \trianglelefteq O'$, if and only if one of the following holds:

(1) $O = O'$
(2) $O = \bot$

(3) $O = O_1|...|O_m$ and $O' = O_1|...|O_n$ with $1 \leq m < n$

(4) O is a partial set and O' is a partial or complete set, and for each $O_i \in O - O'$, there exists $O'_i \in O' - O$ such that $O_i \trianglelefteq O'_i$

(5) $O = [A_1 \Rightarrow O_1, ..., A_m \Rightarrow O_m]$ and $O' = [A_1 \Rightarrow O'_1, ..., A_n \Rightarrow O'_n]$ with $1 \leq m \leq n$ such that $O_i \trianglelefteq O'_i$ for $i \in \{1..m\}$

(6) O is a non-marked object and O' is a marked object $m_1|...|m_n : O''$ such that $O \trianglelefteq O''$

(7) $O{=}m_1|...|m_l : O_1$ and $O'{=}m_1|...|m_n : O'_1$ with $1 \leq l \leq n$ such that $O_1 \trianglelefteq O'_1$

The following are several examples:

$$\begin{array}{lll}
a & \trianglelefteq & a \\
\perp & \trianglelefteq & a \\
\perp & \trianglelefteq & \{\perp\} \\
\perp & \trianglelefteq & [A \Rightarrow a] \\
a_1 & \trianglelefteq & a_1|a_2 \\
a_1|a_2 & \trianglelefteq & a_1|a_2|a_3 \\
m_1 : a_1 & \trianglelefteq & m_1 : a_1|m_2 : a_2 \\
\langle a_1 \rangle & \trianglelefteq & \langle a_1, a_2 \rangle \\
\langle a_1 \rangle & \trianglelefteq & \{a_1, a_2\} \\
\{a_1, a_2\} & \trianglelefteq & \{a_1, a_2\}
\end{array}$$

$$\begin{array}{l}
[A \Rightarrow \perp, B \Rightarrow b] \quad \trianglelefteq [A \Rightarrow a, B \Rightarrow b] \\
[A \Rightarrow a, B \Rightarrow \langle b_1 \rangle] \trianglelefteq m_1 : [A \Rightarrow a, B \Rightarrow \langle b_1, b_2 \rangle] \\
\qquad\qquad\qquad\quad \trianglelefteq m_1|m_2 : [A \Rightarrow a, B \Rightarrow \{b_1, b_2\}]
\end{array}$$

It turns out that the less or equal relationship has the following property.

Proposition 1 The less or equal informative relationship is a partial order.

The less or equal informative relationship is used to express that one object is part of another object. It is used to determine when two objects can be integrated and to show the properties of the integration operation.

Let O and O' be two objects or semistructured objects and $K = \{A_1, ..., A_m\}$ a set of attributes. Then O and O' are *integratable* with respect to K if and only if one of the following holds:

(1) both are constants and are equal

(2) both are markers and are equal

(3) both are or-values and are equal set-wise

(4) both are partial sets

(5) O is a partial set and O' is a complete set such that $O \trianglelefteq O'$

(6) both are complete sets and are equal

(7) both are tuples such that $O.A_i$ and $O'.A_i$ are integratable with respect to K for $1 \leq i \leq m$, or O and O' do not have attributes $A_1, ..., A_m$

(8) O is a non-marked object and O' is a marked object $m_1|...|m_n : O''$ such that O and O'' are integratable with respect to K

(9) $O{=}m_1|...|m_l : O_1$ and $O'{=}m'_1|...|m'_n : O'_1$, such that O_1 and O'_1 are *integratable* with respect to K

Note that the set of attributes $\{A_1, ..., A_m\}$ in the above definition is similar to the notion of key in the relational model. It is used as the basis for the integration operation to be introduced shortly. Also note that two null/unknown values ($\perp$) or two distinct or-values are not considered to be the same.

The following pairs of objects are integratable on $K = \{A, B\}$:

$$\begin{array}{lllr}
a & \text{and} & a & \text{by (1)} \\
m & \text{and} & m & \text{by (2)} \\
a_1|a_2 & \text{and} & a_1|a_2 & \text{by (3)} \\
\langle a_1 \rangle & \text{and} & \langle a_1, a_2 \rangle & \text{by (4)} \\
\langle a_1 \rangle & \text{and} & \{a_1, a_2\} & \text{by (5)} \\
\{a_1, a_2\} & \text{and} & \{a_1, a_2\} & \text{by (6)}
\end{array}$$

$$\begin{array}{lr}
[A \Rightarrow a, B \Rightarrow b, C \Rightarrow \langle c_1 \rangle] \quad \text{and} & \\
[A \Rightarrow a, B \Rightarrow b, C \Rightarrow \langle c_2 \rangle] & \text{by (7)} \\
[A \Rightarrow a, B \Rightarrow \langle b_1 \rangle, C \Rightarrow \langle c_1 \rangle] \text{ and} & \\
m : [A \Rightarrow a, B \Rightarrow \{b_1, b_2\}, C \Rightarrow \{c_2, c_3\}] & \text{by (8)} \\
m_1 : [A \Rightarrow a, B \Rightarrow b, C \Rightarrow [D \Rightarrow d_1]] \text{ and} & \\
m_2 : [A \Rightarrow a, B \Rightarrow b, C \Rightarrow [D \Rightarrow d_2]] & \text{by (9)}
\end{array}$$

The following pairs of objects are not integratable on $K = \{A, B\}$ as they have different values for A and B whenever applicable:

$$\begin{array}{lll}
a_1 & \text{and} & a_2 \\
a_1 & \text{and} & a_1|a_2 \\
\langle a_1 \rangle & \text{and} & \{a_2, a_3\}
\end{array}$$

$$\begin{array}{llll}
[A \Rightarrow a_1, & B \Rightarrow \perp, & C \Rightarrow \{c_1\}] & \text{and} \\
[A \Rightarrow a_1, & B \Rightarrow b_1, & C \Rightarrow \{c_1\}] & \\
[A \Rightarrow \perp, & B \Rightarrow b_1, & C \Rightarrow \{c_1\}] & \text{and} \\
[A \Rightarrow \perp, & B \Rightarrow b_2, & C \Rightarrow \{c_1\}] &
\end{array}$$

A nonempty set S of objects (or semistructured objects) is *integratable* with respect to a set K of attributes if and only if every pair of objects (or semistructured objects) in S are integratable on K.

Let S be a set of objects (or semistructured objects) and S' an integratable subset of S with respect to K. Then S' is a *maximal integratable set* in S with respect to K if there does not exist an object (or semistructured object) in $S - S'$ that is integratable with each object in S' with respect to K.

Consider the following heterogeneous set of objects:

$$\begin{array}{l}
S = \{a, b, a_1|a_2, \langle a_1 \rangle, \langle a_2 \rangle, \langle a_3 \rangle, \{a_1\}, \{a_1, a_2\}, \\
\qquad [A \Rightarrow a_1], [A \Rightarrow a_1, B \Rightarrow b_1]\}
\end{array}$$

The maximal integratable sets in S based on $K = \{A\}$ are as follows:

$$\begin{array}{l}
S_1 = \{a\} \\
S_2 = \{b\} \\
S_3 = \{a_1|a_2\} \\
S_4 = \{\langle a_1\rangle, \langle a_2\rangle, \langle a_3\rangle\} \\
S_5 = \{\langle a_1\rangle, \{a_1\}\} \\
S_6 = \{\langle a_1\rangle, \langle a_2\rangle, \{a_1, a_2\}\} \\
S_7 = \{[A \Rightarrow a_1], [A \Rightarrow a_1, B \Rightarrow b_1]\}
\end{array}$$

As shown in S_5 and S_6 above, an integratable set of partial/complete sets contains at most one complete set.

The main goal of this paper is to define how to integrate semistructured objects. Given a set of semistructured objects, we first divide them into a number of maximal integratable sets and then integrate each integratable set.

Now we introduce the integration operator I_K which is used to integrate a set of semistructured objects based on a set of attributes K. It is similar to the union operator in [5], the join operator in [7], and the grouping operator in [34], but it handles partial and inconsistent information.

Let $K = \{A_1, ..., A_m\}$ be a non-empty set of attributes. The *integration* operator I on a set of objects based on K, denoted by I_K, is defined recursively on a set of objects as follows:

(1) $I_K(\{O\}) = O$.

(2) $I_K(\{O_1, ..., O_n, \bot\}) = I_K(\{O_1, ..., O_n\})$.

(3) If S is a set of partial sets, then $I_K(S) = \langle I_K(S'') \mid S'' = \{O \mid O \in S' \text{ and } S' \in S\}$ is a maximal integratable set in $S\rangle$.

(4) If S is an integratable set with a complete set O, then $I_K(S) = O$.

(5) If S is an integratable set of tuples, then $I_K(S) = [A_1 \Rightarrow I_K(S_1), ..., A_n \Rightarrow I_K(S_n)]$, where $S_i = \{O_i \mid [A_1 \Rightarrow O_1, ..., A_i \Rightarrow O_i, ..., A_n \Rightarrow O_n] \in S\}$ for $1 \leq i \leq n$.

(6) $I_K(\{O_1, ..., O_i, m_1 : O'_1, ..., m_j : O'_j\}) = m_1|...|m_j$: $I_K(\{O_1, ..., O_i, O'_1, ..., O'_j\})$.

(7) If S is a set of objects divided into maximal integratable sets $S_1, ..., S_n$ with respect to K, then $I_K(S) = I_K(S_1)|...|I_K(S_n)$.

The following are several examples where $K = \{A, B\}$:

$$\begin{array}{ll}
I_K(\{a\}) = a & \text{by } (1) \\
I_K(\{a, \bot\}) = a & \text{by } (2),(1) \\
I_K(\{a_1, a_2\}) = a_1|a_2 & \text{by } (7) \\
I_K(\{a_1, a_2|a_3\}) = a_1|a_2|a_3 & \text{by } (7) \\
I_K(\{a_1, [A \Rightarrow a_1]\}) = a_1|[A \Rightarrow a_1] & \text{by } (7) \\
I_K(\{a_1, \{a_1\}\}) = a_1|\{a_1\} & \text{by } (7) \\
I_K(\{a_1, a_2, \bot\}) = I_K(\{a_1, a_2\}) = a_1|a_2 & \text{by } (2),(7)
\end{array}$$

$$\begin{array}{ll}
I_K(\{\langle a_1\rangle, \langle a_2\rangle, \langle a_1, a_2\rangle, \bot\}) & \\
= I_K(\{\langle a_1\rangle, \langle a_2\rangle, \langle a_1, a_2\rangle\}) & \\
= \langle I_K(\{a_1\}), I_K(\{a_2\})\rangle = \langle a_1, a_2\rangle & \text{by } (2),(3) \\
I_K(\{\langle a_1\rangle, \langle a_1, a_2\rangle, \{a_1, a_2, a_3\}\}) & \\
= \{a_1, a_2, a_3\} & \text{by } (4) \\
I_K(\{[A \Rightarrow a, B \Rightarrow \langle b_1\rangle, C \Rightarrow c_1], & \\
\quad [A \Rightarrow a, B \Rightarrow \langle b_2\rangle, C \Rightarrow c_2]\}) & \\
= [A \Rightarrow a, B \Rightarrow \{b_1, b_2\}, C \Rightarrow c_1|c_2] & \text{by } (5) \\
I_K(\{m_1 : \langle a_1\rangle, m_2 : \langle a_2\rangle, m_3 : \langle a_3\rangle\}) & \\
= m_1|m_2|m_3 : \langle a_1, a_2, a_3\rangle & \text{by } (6) \\
I_K(\{\langle a_1\rangle, \langle a_2\rangle, \{a_2, a_3\}\}) & \\
= \langle a_1, a_2\rangle|\{a_2, a_3\} & \text{by } (7),(3),(4)
\end{array}$$

The integration operator is further extended to sets of semistructured objects as follows:

(1) If S is a set of semistructured objects divided into maximal integratable sets $S_1, ..., S_n$ with respect to K, then $I_K(S) = \{I_K(S_1)\} \cup ... \cup \{I_K(S_n)\}$.

(2) If $S_1, ..., S_n$ are sets of semistructured objects, then $I_K(S_1, ..., S_n) = I_K(S_1 \cup ... \cup S_n)$.

Example 4 Consider the following two sets of semistructured objects which are essentially two Bibtex files, one of which (S_1) contains pure journal papers and the other (S_2) contains both journal and conference papers.

$$\begin{array}{llll}
S_1 = \{[\mathit{article} \Rightarrow A78 : [& \mathit{title} & \Rightarrow \mathit{Datalog}, & \\
 & \mathit{author} & \Rightarrow \mathit{Ann}, & \\
 & \mathit{year} & \Rightarrow 1978 &]], \\
\quad [\mathit{article} \Rightarrow B80 : [& \mathit{title} & \Rightarrow \mathit{Oracle}, & \\
 & \mathit{author} & \Rightarrow \mathit{Bob}, & \\
 & \mathit{year} & \Rightarrow 1980 &]], \\
\quad [\mathit{article} \Rightarrow J88 : [& \mathit{title} & \Rightarrow \mathit{DOOD}, & \\
 & \mathit{author} & \Rightarrow \mathit{Joe}, & \\
 & \mathit{year} & \Rightarrow 1988 &]] \\
\quad [\mathit{article} \Rightarrow S78 : [& \mathit{title} & \Rightarrow \mathit{Ingres}, & \\
 & \mathit{author} & \Rightarrow \mathit{Sam}, & \\
 & \mathit{journal} & \Rightarrow \mathit{TODS}, & \\
 & \mathit{year} & \Rightarrow 1978 &]], \\
\quad [\mathit{article} \Rightarrow S85 : [& \mathit{title} & \Rightarrow \mathit{NF2}, & \\
 & \mathit{author} & \Rightarrow \mathit{Sam}, & \\
 & \mathit{journal} & \Rightarrow \mathit{IS}, & \\
 & \mathit{year} & \Rightarrow 1985 &]] \\
\quad \} & & &
\end{array}$$

$$
\begin{array}{llll}
S_2 = \{[article \Rightarrow A78 : & [\,title & \Rightarrow Datalog, & \\
 & author & \Rightarrow Tom, & \\
 & year & \Rightarrow 1978 &]], \\
\quad [article \Rightarrow B82 : & [\,title & \Rightarrow Oracle, & \\
 & author & \Rightarrow Bob, & \\
 & journal & \Rightarrow JACM &]], \\
\quad [article \Rightarrow P90 : & [\,title & \Rightarrow DOOD, & \\
 & author & \Rightarrow Pam, & \\
 & journal & \Rightarrow JLP &]], \\
\quad [inProc \Rightarrow A75 : & [title & \Rightarrow NF2, & \\
 & author & \Rightarrow Ann, & \\
 & year & \Rightarrow 1975 &]], \\
\quad [inProc \Rightarrow T79 : & [title & \Rightarrow RDB, & \\
 & author & \Rightarrow Tom, & \\
 & conf & \Rightarrow PODS &]], \\
\quad [inProc \Rightarrow S76 : & [title & \Rightarrow Ingres, & \\
 & author & \Rightarrow Sam, & \\
 & confs & \Rightarrow EDBT &]] \\
\quad \} & & &
\end{array}
$$

If we want to combine them based on the value of the attribute *title*, then we can use the integration operator I_K with $K = \{title\}$ as follows:

$$
\begin{array}{llll}
\multicolumn{4}{l}{I_K(S_1, S_2) = I_K(S_1 \cup S_2) =} \\
\{[article \Rightarrow \quad A78: & [title & \Rightarrow Datalog, & \\
 & author & \Rightarrow Ann|Tom, & \\
 & year & \Rightarrow 1978 &]], \\
\;[article \Rightarrow B80|B82: & [title & \Rightarrow Oracle, & \\
 & author & \Rightarrow Bob, & \\
 & journal & \Rightarrow JACM, & \\
 & year & \Rightarrow 1980 &]] \\
\;[article \Rightarrow J88|P90: & [title & \Rightarrow DOOD, & \\
 & author & \Rightarrow Joe|Pam, & \\
 & journal & \Rightarrow JLP, & \\
 & year & \Rightarrow 1988 &]] \\
\;[article \Rightarrow \quad S78: & [title & \Rightarrow Ingres, & \\
 & author & \Rightarrow Sam, & \\
 & journal & \Rightarrow TODS, & \\
 & year & \Rightarrow 1978 &]], \\
\;[article \Rightarrow \quad S85: & [title & \Rightarrow NF2, & \\
 & author & \Rightarrow Sam, & \\
 & journal & \Rightarrow IS, & \\
 & year & \Rightarrow 1985 &]], \\
\;[inProc \Rightarrow \quad A75: & [title & \Rightarrow NF2, & \\
 & author & \Rightarrow Ann, & \\
 & year & \Rightarrow 1975 &]], \\
\;[inProc \Rightarrow \quad T79: & [title & \Rightarrow RDB, & \\
 & author & \Rightarrow Tom, & \\
 & conf & \Rightarrow PODS &]], \\
\;[inProc \Rightarrow \quad S76: & [title & \Rightarrow Ingres, & \\
 & author & \Rightarrow Sam, & \\
 & confs & \Rightarrow EDBT &]] \\
\} & & &
\end{array}
$$

Note that the two semistructured objects with title Ingres (also NF2) are not integratable as one is an *article* object and another is an *inProc* object. In addition, if we integrate S_1 and S_2 based on some other attributes, such as *title, author*, the results will be different.

As the above example shows, the integration operation integrates sets of semistructured objects and records inconsistency in the meantime. The user can then solve the inconsistency based on the results. This feature is unique compared to other approaches.

The *integration* operator has the following properties.

Proposition 2 Let $S_1, ..., S_n$ be sets of semistructured objects and K a non-empty set of attributes. Then $S_i \trianglelefteq I_K(S_1, ..., S_n)$ for $1 \leq i \leq n$.

Continuing with the above example, we have

$$S_1 \trianglelefteq I_K(S_1, S_2)$$

$$S_2 \trianglelefteq I_K(S_1, S_2)$$

Proposition 3 Let $S_1, ..., S_n$ be sets of semistructured objects and K_1 and K_2 non-empty sets of attributes. Then $K_1 \subseteq K_2$ implies $I_{K_2}(S_1, ..., S_n) \trianglelefteq I_{K_1}(S_1, ..., S_n)$.

For example, let $K_1 = \{title\}$ and $K_2 = \{title, author\}$. Then for the two sets of semistructured objects in Example 4, we have

$$I_{K_2}(S_1, S_2) \trianglelefteq I_{K_1}(S_1, S_2).$$

4 Conclusion

The need for integrating semistructured data naturally arises in the real-world applications. In this paper, we present a novel approach for integrating semistructured data with partial and inconsistent information. A powerful operator called *integration* is defined and the semantic properties are discussed in detail. This work provides a firm foundation in discussing the semantics of semistructured data in which heterogeneous data may come from various data sources with incomplete or inconsistent information.

Further work will focus on other possible operations in manipulating semistructured databases. The current *integration* operator extends *union* and *join* and also handles null/unknown and inconsistent values. Other potential methods could be *intersection*, *difference*, *expand*, etc. which will be investigated in the future work.

References

[1] S. Abiteboul. Querying Semistructured Data. In *Proceedings of the International Conference on Data Base Theory*, pages 1–18. Springer-Verlag LNCS 1186, 1997.

[2] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener. The Lorel Query Language for Semistructured Data. *Journal of Digital Library*, 1(1):68–88, 1997.

[3] J. Ambite, N. Ashish, G. Barish, G. Knoblock, S. Minton, P. Modi, I. Muslea, A. Philpot, and S. Tejada. ARIADNE: A system for constructing mediators for internet sources. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998.

[4] G. Arocena and A. Mendelzon. WebOQL: Restructuring Documents, Databases and Webs. In *Proceedings of the International Conference on Data Engineering*, pages 24–33. IEEE Computer Society, 1998.

[5] F. Bancilhon and S. Khoshafian. A Calculus for Complex Objects. *J. Computer and System Sciences*, 38(2):326–340, 1989.

[6] C. Beeri, G. Elber, T. Milo, Y. Sagiv, O. Shmueli, N. Tishby, Y. Kogan, D. Konopnicki, P. Mogilevski, and N. Slonim. Websuite – A tool suite for harnessing web data. In *Proceedings of the International Workshop on the Web and Databases*, 1998.

[7] O. P. Buneman, S. B. Davidson, and A. Watters. A Semantics for Complex Objects and Approximate Answers. *J. Computer and System Sciences*, 43(1):170–218, 1991.

[8] P. Buneman, S. Davidson, G. Hilebrand, and D. Suciu. A Query Language and Optimization Techniques for Unstructured Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 505–516, 1996.

[9] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In *Proceedings of the 10th Meeting of the Information Processing Society of Japan*, pages 7–18, 1994.

[10] A. Chen, P. Tsai, and J. Hoh. Identifying Object Isomerism in Multidatabase Systems. *Distributed and Parallel Databases*, 4(2):143–165, 1996.

[11] Q. Chen and W. Chu. HILOG: A High-Order Logic Programming Language for Non-1NF Deductive Databases. In W. Kim, J. Nicolas, and S. Nishio, editors, *Proceedings of the International Conference on Deductive and Object-Oriented Databases*, pages 431–452, Kyoto, Japan, 1989. North-Holland.

[12] W. Cohen. Integration of heterogeneous databases without common domains using queries based textual similarity Detecting Approximately Duplicate Database Records. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 201–212, 1998.

[13] O. S. David Konopnicki. W3QS: A Query System for the World-Wide Web. In *Proceedings of the International Conference on Very Large Data Bases*, pages 54–65, Santiago de Chile, Chile, 1995. Morgan Kaufmann Publishers, Inc.

[14] L. Demichiel. Resolving Database Incompatibility: An Approach to Performing Relational Operations over Mismatched Domains. *IEEE Transactions on Knowledge and Data Engineering*, 1(4):485–493, 1989.

[15] D. Fang, J. Hammer, and D. Mcleod. The Identification and Resolution of Semantic Heterogenerity in Multidatabase Systems. In *Multidatabase Systems: An Advanced Solution for Global Information Sharing*, pages 52–60, 1994.

[16] M. Fernandez, D. Florescu, A. Levy, and D. Suciu. A Query Language for A Web-Site Management System. *SIGMOD Record*, 26(3), 1997.

[17] D. Florescu, D. Koller, and A. Levy. Using Probabilistic Information in Data Integration. In *Proceedings of the International Conference on Very Large Data Bases*, pages 216–225, Athens, Greece, 1997. Morgan Kaufmann Publishers, Inc.

[18] M. Goossens, F. Mittelbach, and A. Samarin. *The Latex Companion*. Addison-Wesley, 1994.

[19] S. Huffman and D. Steier. Heuristic Joins to Integrate Structured Heterogeneous Data. In *Working notes of the AAAI spring Symposium on information gethering in heterogeneous distributed environments*, 1995.

[20] R. Hull and G. Zhou. A Framework for Supporting Data Integration Using the Materialized and Virtual Approaches. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 481–492, 1996.

[21] T. Imielinski and W. L. Jr. Incomplete Information in Relational Databases. *Journal of ACM*, 31(4):761–791, 1984.

[22] W. L. Jr. On Databases with Incomplete Information. *Journal of ACM*, 28(1):41–70, 1981.

[23] M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of ACM*, 42(4):741–843, 1995.

[24] L. Lakshmanan, F. Sadri, and I. Subramanian. A Declarative Language for Querying and Restructuring the Web. In *Proceedings of the 6th International Workshop on Research Issues in Data Engineering*, 1996.

[25] L. Lamport. *Latex User Guide and Reference Manual*. Addison Wesley, 2 edition, 1994.

[26] A. Levy. Obtaining Complete Answers from Incomplete Databases. In *Proceedings of the International Conference on Very Large Data Bases*, pages 402–412. Morgan Kaufmann Publishers, Inc., 1996.

[27] A. Levy, A. Rajaraman, and J. Ordille. Querying heterogeneous information sources using source descriptions. In *Proceedings of the International Conference on Very Large Data Bases*, pages 251–262. Morgan Kaufmann Publishers, Inc., 1996.

[28] L. Libkin. A Relational Algebra for Complex Objects based on Partial Information. In *Proceedings of the Conference on Mathematical Foundations of Programming Semantics*, pages 26–41, Rostock, Germany, 1991. Springer-Verlag LNCS 495.

[29] L. Libkin. *Aspects of Partial Information in Databases*. Ph.D Thesis, University of Pennsylvania, 1994.

[30] L. Libkin. Approximation in Databases. In *Proceedings of the International Conference on Data Base Theory*, pages 414–424, Prague, Czech Republic, 1995. Springer-Verlag LNCS 326.

[31] L. Libkin. Normalizing Incomplete Databases. In *Proceedings of the ACM Symposium on Principles of Database Systems*, pages 219–230, San Jose, California, 1995.

[32] E. Lim, J. Srivastava, and S. Shekhar. Resolving attribute incompatibility in database integration: an evidential reasoning approach. In *Proceedings of the International Conference on Data Engineering*, pages 154–163. IEEE Computer Society, 1994.

[33] M. Liu. ROL: A Deductive Object Base Language. *Information Systems*, 21(5):431 – 457, 1996.

[34] M. Liu. Relationlog: A Typed Extension to Datalog with Sets and Tuples. *Journal of Logic Programming*, 36(3):271–299, 1998.

[35] A. Mendelzon, G. Mihaila, and T. Milo. Querying the World Wide Web. In *Proceedings of the First International Conference on Parellel and Distributed Information System*, pages 80–91, 1996.

[36] A. Monge and C. Elkan. An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 383–394, 1997.

[37] A. Motro and I. Rakov. Estimating the Quality of Data in Relational Databases. In *Proceedings of the 1996 Conference on Information Quality*, pages 94–106, 1996.

[38] K. Munakata. Integration of Semistructured Data Using Outer Joins. In *Proceedings of the Workshop on Management of Semistructured Data*, 1997.

[39] A. Ohori. Semantics of Types for Database Objects. *Theoretical Computer Science*, 76(1):53–91, 1990.

[40] F. Tseng, A. Chen, and W. Yang. Answering Heterogeneous Databases Queries with Degrees of Uncertainty. *Distributed and Parallel Databases*, 1(3):281–302, 1993.