Merging Sets of Taxonomically Organized Data Using Concept Mappings under Uncertainty*

David Thau[1], Shawn Bowers[2], and Bertram Ludäscher[1,2]

[1] Dept. of Computer Science, University of California Davis, CA 95616
[2] Genome Center, University of California Davis, CA 95616
{thau,sbowers,ludaesch}@ucdavis.edu

Abstract. We present a method for using aligned ontologies to merge taxonomically organized data sets that have apparently compatible schemas, but potentially different semantics for corresponding domains. We restrict the relationships involved in the alignment to basic set relations and disjunctions of these relations. A merged data set combines the domains of the source data set attributes, conforms to the observations reported in both data sets, and minimizes uncertainty introduced by ontology alignments. We find that even in very simple cases, merging data sets under this scenario is non-trivial. Reducing uncertainty introduced by the ontology alignments in combination with the data set observations often results in many possible merged data sets, which are managed using a possible worlds semantics. The primary contributions of this paper are a framework for representing aligned data sets and algorithms for merging data sets that report the presence and absence of taxonomically organized entities, including an efficient algorithm for a common data set merging scenario.

1 Introduction

We address the problem of merging data sets when the domains of the data attributes overlap but are not equivalent. Consider, e.g., two data sets that report observations of the presence or absence of biological taxa in a given region and at a given time.[1] Each of the dimensions, biological, spatial, and temporal, may be represented using a taxonomy, and the data sets may each use different taxonomies for any given dimension. In the absence of any information about the relationship between the concepts in their taxonomies, the data sets can be naively merged by simply concatenating the observations into a single data set. This method, however, may result in a self-contradictory data set, or one that contains hidden redundancies and uncertainty. Given information about how the data sets' taxonomies relate (an *alignment*), the data sets can be merged in a more informed way. We present here a methodology for merging data sets that takes advantage of alignments between taxonomies while detecting contradictions, and minimizes uncertainties that may arise in the merge.

* Work supported by NSF awards IIS-0630033, DBI-0743429, and DBI-0753144.

[1] Presence data sets such as this are very common. For example, epidemiological studies track the presence of diseases over time and space [1]. In ecological and biodiversity research, many data sets stored in data repositories (such as Metacat [2]) are composed of lists of biological taxa found in specified geographic extents over given periods of time.

R. Meersman, T. Dillon, P. Herrero (Eds.): OTM 2009, Part II, LNCS 5871, pp. 1103–1120, 2009.

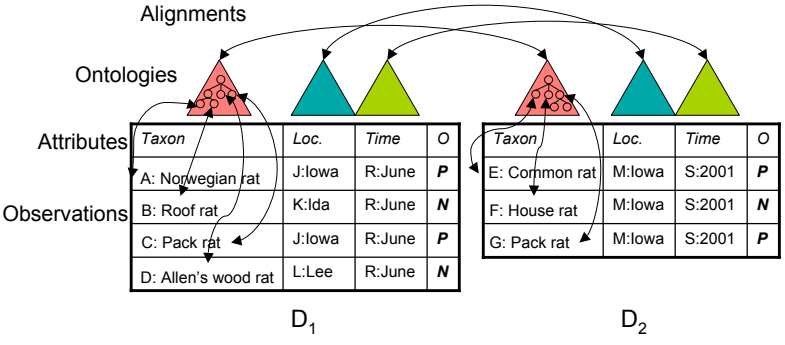

Fig. 1. Two data sets, with corresponding ontologies and ontology alignments

Figure 1 presents a simple example involving two presence data sets D_1 and D_2 that describe types of rats found to be present or absent at specific places and times. The *Taxon* column represents biological taxa, preceded by an abbreviation (e.g., "A" for Norwegian rat). The taxonomies used to define and relate the taxa are represented by ontologies depicted above the Taxon columns of the data sets. The creators of the two data sets may have used different field guides to identify the taxa, in which case the Taxon ontologies must be aligned to account for differences between the field guides. The *Loc* column represents spatial locations: counties in Iowa in the first data set, and the State of Iowa in the second data set. The ontologies from which the location names are drawn are represented above their respective columns, and an alignment relates the location names used in the data sets. Note that Iowa is both the name of a US State, and of a county in that state. *Time* records when the observations are made. Finally, *O* records whether or not a given taxon, at a given place and a given time is present (`P`) or not present (`N`) (absent).[2] We assume here that presence and absence are complements; a taxon cannot be both present and absent at a given location and time.

Merge Scenarios. Each data set shown in Figure 1 provides a perspective on the state of the world at a given place and time, according to a given observer. We call each data set a *scenario*. Merging the data sets should provide a more complete description of the state of the world. However, it may not be clear how to best merge the data sets, and many scenarios may be possible. For example, the merged data set shown in Table 1(a) describes the scenario arising from a simple union of the source data sets. Although it seems like an obvious merge, it makes many, possibly incorrect, assumptions. First, it assumes every name is distinct from every other name. However, concepts between data sets can be equivalent, potentially rendering the merge in Table 1(a) inconsistent. If concept A in D_1 (Norwegian rat) is equivalent to concept F in D_2 (House rat), and R in D_1 is equivalent to S in D_2 (both studies were carried out in June, 2001), and concept J in D_1 (Iowa County) is a proper part of M (Iowa State) in D_2, then the observations corresponding to rows 1 and 6 in Table 1(a) would be reporting both the presence and absence of the same taxon at the same place and time. Table 1(a) further assumes that

[2] Note that the presence of a taxon does not imply that only *one* instance of that taxon was seen at that place, at that time.

Table 1. Three possible merges of the data sets in Figure 1

(a)

Taxon	*Loc.*	*Time*	*O*
A	J	R	P
B	K	R	N
C	J	R	P
D	L	R	N
E	M	S	P
F	M	S	N
G	M	S	P

(b)

Taxon	*Loc.*	*Time*	*O*
AE	JM	RS	P
AE	$\bar{J}\bar{K}\bar{L}M$	RS	P
BF	KM	RS	N
BF	$\bar{J}\bar{K}\bar{L}M$	RS	N
CG	JM	RS	P
CG	$\bar{J}\bar{K}\bar{L}M$	RS	N
D	LM	RS	N
D	$\bar{J}\bar{K}\bar{L}M$	RS	N

(c)

Taxon	*Loc.*	*Time*	*O*
AE	*JKLM*	*RS*	P
BF	*JKLM*	*RS*	N
CG	*JKLM*	*RS*	P
D	*JKLM*	*RS*	N

an unreported taxon does not imply the absence of that taxon. If an unreported taxon is assumed to be absent, and, e.g., if Norwegian rat in D_1 is disjoint from all the taxa listed in D_2, it would be problematic that D_1's observer reported the presence of at least one Norwegian rat and D_2's observer did not. Table 1(b) and (c) present two alternative scenarios. Table 1(b) assumes an alignment in which certain concepts are equivalent (e.g., $A \equiv E$ as represented by the new taxon AE). The alignment also asserts that certain concepts are proper parts of others. For example, concept J is aligned as a proper part of concept M ($J \subsetneq M$). This is represented by introducing new location concepts, JM represents the region where J and M overlap ($J \cap M$), and $\bar{J}\bar{K}\bar{L}M$ represents the region of M that excludes J, K and L ($M \setminus (J \cup K \cup L)$).

Sources of Uncertainty. Uncertainty induces multiple possible merges. For example, the different merges in Table 1 occur because of uncertainty in the alignment between ontologies: the concepts A and E might be distinct concepts, as in Table 1(a), or equivalent concepts, as in Table 1(b). This uncertainty may have been explicitly stated by the ontology aligner ($A \equiv E$ or $A \,!\, E$), or it may have been inferred from an incomplete alignment [3]. We call this kind of uncertainty *disjunctive relation uncertainty* (DRU) because it involves a disjunction of relations (equivalent *or* disjoint, in this case). Disjunctive relations may also exist within individual ontologies. For example, the traditional interpretation of "isa" as "equals or is included in" [4] is a disjunctive relation.

Even when the relationship between two concepts is certain, the relationship may lead to uncertainty. For example, if an alignment holds that concept A according to D_1 is a kind (i.e., proper subset) of concept E according to D_2 ($A \subsetneq E$), it is unclear whether or not any of the E's reported in data set 2 are also A's. There are two possibilities: either all the observed rats are both A's and E's (AE), or some of the rats are E's but not A's ($\bar{A}E$). We call this source of uncertainty *basic relation uncertainty* (BRU) because it arises from basic set relations. Whereas disjunctive relation uncertainty exists at the ontology level, basic relation uncertainty occurs at the level of the observations in the data sets. To reliably resolve this uncertainty, one would have to ask for clarification from the data set's observer.

Our goal is to create data set merges free of BRU and DRU. While BRU and DRU may appear in source data sets, in our experience high quality data sets do not contain these types of uncertainty. We provide algorithms for merging data sets that are free of BRU and DRU, as well as those that are not. However, the algorithm for merging data

sets that do not contain BRU or DRU is considerably more efficient than the one for merging data sets that already contain uncertainty.

Contributions and Road Map. This paper contributes a novel modeling framework for merging data sets with aligned domains under uncertainty. We describe several sources of uncertainty within data sets as well as arising from the merging of data sets; and present a possible worlds semantics for managing this uncertainty. Finally, we provide algorithms for merging data sets in this context, providing NEXP-time algorithms for the general case of generating possible worlds, and an NP-time SAT-based solution for the common case of merging source data sets that do not contain BRU and DRU.

We proceed as follows. Section 2 describes our basic approach for managing uncertainty that arises while merging data sets with aligned attribute domains. Section 3 provides formal representations for the various aspects of our framework: data sets, observations, relationships between domains, presence, absence, and possible merges. Section 4 describes algorithms for merging data sets that contain BRU and DRU, as well as data sets that do not contain such sources of uncertainty. Section 5 compares the efficiency of the different merging algorithms, demonstrating improvements in the feasibility and performance of the optimized algorithms. Finally, Section 6 describes related work and concludes the paper.

2 Basic Approach

This section provides an informal description of the elements involved in data set merging, and a high level description of our approach to performing the merge. We consider data sets that can be defined as relations over finite sets of attributes. Data items within a data set are tuples of values, where the values are drawn from their respective attribute domains. In this work, the values represent *concepts* (classes), which are sets of instances. For example, taxa are sets of (perhaps unknown) biological specimens, locations are sets of points in space, and times are sets of moments. The attribute domains may be structured, containing the domain concepts and relations between them stated in some language (e.g., first-order logic, monadic logic, or description logic). To emphasize the richness of the domains, we call them *ontologies* and we call a data set's collection of ontologies its *metadata*. We assume the source data sets are internally consistent. Inconsistency, however, can occur in a number of places. A data set may contain contradictory information if, e.g., it states both the absence and presence of a taxon at a given place and time. A data set may also be inconsistent with its metadata, e.g., if the metadata states that taxa A and B are equivalent (represent equivalent sets), but the data say that A is present at a given place and time and B is absent. Finally, the ontologies in the metadata may be inconsistent. We define a *legal* data set as one that does not violate any of these consistency constraints. We further define an *unambiguous* data set as a legal data set that contains neither basic nor disjunctive relation uncertainty.

Merging data sets is enabled by alignments between data set ontologies. Alignments are sets of *articulations* of the form: "$A\ r\ B$", where A and B are ontology concepts, and r is a relation between the sets that the concepts represent. Relations are drawn from the RCC-5 algebra [5], which has proven to be useful in biological taxonomy alignment [6,7]. A key feature of RCC-5 is that in addition to five basic set relations (e.g., set

equivalence and set disjointness) disjunctions of relations are represented. These disjunctive relations are necessary when the relationship between two sets is only partially known (e.g., set A *either* overlaps with *or* is a proper part of set B).

Each scenario in Table 1 describes one unambiguous data set. We propose treating each possible merged data set as one of many possible worlds [8,9] in a *possible worlds set* (PWS). Given two data sets, one could generate the appropriate PWS by generating an *initial world set* (IWS) containing every conceivable world (restricted by the finite domains of the metadata), including those worlds that violate the alignment and certainty constraints, and then reducing this set by eliminating columns and rows that violate the constraints.

Unfortunately, this approach is intractable. Consider the extremely simple scenario shown in Figure 2(a) having two data sets D_1 and D_2 with taxon A present in D_1, and B present in D_2. Each data set has a single biological attribute, and that attribute can only take one value: A for D_1 and B for D_2, and an articulation between these concepts states that $A \subsetneq B$. To generate an IWS, we first determine all conceivable conditions that may or may not hold based on the concepts in the data set ontologies. There are four ways to combine the biological concepts A and B: a biological specimen might be an example of AB,$A\bar{B}$,$\bar{A}B$, or $\bar{A}\bar{B}$. We call each of these combinations a *combined concept*. Each combined concept represents a set of instances, and a data set reports whether there are no instances of the set present within the context of the data set (absence), or at least one instance from the set present (presence). The resulting IWS has $2^2 = 4$ conditions and $2^4 = 16$ worlds. This IWS can be conveniently represented with a *world set relation* [10] as shown in Figure 2(b). In this table, the conditions are represented as columns, and each world is a row in the table. The number 1 indicates that instances of the condition are present in a given possible world, and 0 represents the absence of instances of that condition. The first world represents the (impossible) situation in which instances of all the conditions are present. This is impossible because the first combined concept, $A\bar{B}$ cannot be present (in fact, is not satisfiable) because $A \subsetneq B$.

Once the IWS table is created, it may be reduced by removing conditions and possible worlds that violate constraints or are unsupported by the input data sets. For example, because $A \subsetneq B$, $A\bar{B}$ is an impossible combined concept, any condition involving it cannot hold. Similarly, because D_1 reported the presence of A, and $A \subsetneq B$, AB must be 1 in every possible world, and any world with 0 in that column should be removed. In addition, conditions for which there is no evidence should be removed. In this example, the last

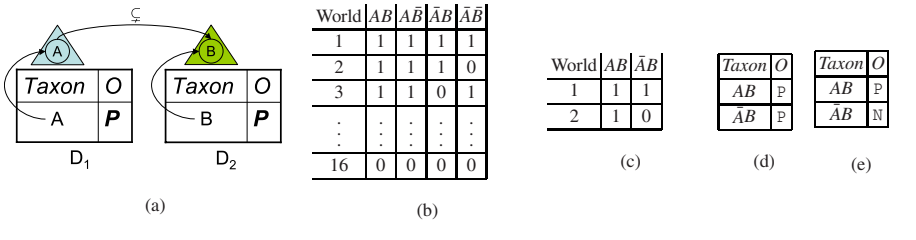

Fig. 2. (a) A very simple scenario, (b) its initial world set, (c) the reduced possible world set, (d) and (e) the corresponding merged data sets

condition of the IWS should be removed because neither data set describes specimens that are neither A nor B. Finally redundant rows created by the deletion of combined contexts should be removed. Removing all of the impossible, redundant, and unsupported information results in the two possible worlds in the PWS shown in Figure 2(c). The two merged data sets that correspond to these possible worlds are shown in Figures 2(d) and 2(e).

In more typical situations, this approach will not work. For example, merging two data sets with three attributes, where each attribute has a corresponding ontology (O_1, O_2 and O_3) with $|O_n|$ concepts will result in a IWS with $C = 2^{|O_1|+|O_2|+|O_3|}$ columns, and 2^C rows. The simple scenario in Figure 1 would lead to an IWS with $2^{7+4+2} = 8192$ conditions and 2^{8192} worlds; a number of worlds too large to enumerate, much less manipulate. A primary contribution of this work is a set of more tractable algorithms for generating the appropriate PWS. To do so, we more formally define the framework within which the merge occurs.

3 Framework

Dimensions, Concepts, and Ontologies. We distinguish between distinct types of objects using *classification dimensions* (or *dimensions* for short). Here we are primarily concerned with three dimensions: *spatial* (e.g., locations and regions), *temporal* (e.g., points in time and intervals), and *biological* (e.g., organisms classified via biological taxonomies). Vocabularies for classifying objects are represented using *ontologies* $\mathbf{O} = (\{C_1, ..., C_n\}, \Sigma)$ each consisting of a finite set of *concepts* and a set of constraints Σ on those concepts. Each concept C specifies a set of objects that are considered to be *instances* of C. Each ontology $\mathbf{O}$ is associated with a dimension given by a function $dim(\mathbf{O})$. Thus, each concept of a particular ontology classifies objects of the *same* dimension.[3] Below, we assume that biological ontology concepts describe sets of organisms, spatial ontology concepts describe sets of points in space, and temporal ontology concepts represent sets of moments in time.

Each concept within an ontology may be represented as a unary predicate, and relations between predicates can be described using first-order logic (or some appropriate subset). For example, we may define the biological ontology for a data set as a set of concepts $B_1, \cdots, B_n \in \mathbf{B}$, and a set of "isa" relations between these concepts represented in monadic first-order logic as $\forall x : B_i(x) \rightarrow B_j(x)$. This formula states that any instance of biological concept (or *taxon*) B_i is also an instance of taxon B_j. When merging data sets, we exploit the constraints given by the structure of $\mathbf{B}$.

Data Sets and Observations. Data sets are represented as relations D over the schema

$$\mathbf{C}_1 \times \cdots \times \mathbf{C}_n \times \mathbf{D}_1 \times \cdots \times \mathbf{D}_m$$

where each $\mathbf{C}_i$ denotes a *context attribute* and each $\mathbf{D}_j$ denotes a *data attribute*. A total function, $m : C \rightarrow O$ maps each context attribute to an associated ontology, and the domain of the attribute is restricted to the concepts in the associated ontology. A data

[3] An ontology typically contains terms from different dimensions and can be viewed in our framework as consisting of one or more domains.

attribute represents a set of possible values corresponding to observations made over the given context attributes. In our example, we consider the following special case

$$\mathbf{C_B} \times \mathbf{C_S} \times \mathbf{C_T} \times \mathbf{D_O}$$

where $\mathbf{C_B}$ represents a required biological context attribute (e.g., organisms classified via biological taxonomies), $\mathbf{C_S}$ represents an optional *spatial* context attribute, $\mathbf{C_T}$ represents an optional *temporal* context attribute, and $\mathbf{D_O}$ represents a simple data attribute denoting a presence or absence observation over context attributes. In general, presence data sets are represented by one or more records of the form $D(b,s,t,o)$ where $b \in \mathbf{C_B}$, $s \in \mathbf{C_S}$, and $t \in \mathbf{C_T}$ are concepts and $o \in \mathbf{D_O}$ is either `P`, meaning at least one b was observed in region s during time t, or `N`, meaning no instances of b were found in region s during time t. We call each record in a data set an *observation*. Although biodiversity data sets often contain additional context information and measurements [11], the features described above are sufficient to demonstrate the core issues of data set merging that we address.

Absence Closure. So far we have described data sets containing presence and absence information explicitly. In some cases, a data set may contain only presence information, but intend that absence is implied when an observation is not made. We say that a presence data set is closed under absence if for each context term $b_i \in \mathbf{C_B}$, $s_j \in \mathbf{C_S}$, and $t_k \in \mathbf{C_T}$ there is a record $D(b_i,s_j,t_k,o)$. If no such record exists in the data set, we can close the data set by asserting an absence observation via the record $R(b_i,s_j,t_k,\mathtt{N})$.

Relationships Between Ontologies. In this work we describe merging two data sets of the aforementioned schema. Although the schemas are the same, the ontologies for the biological, spatial and temporal context attributes may differ between data sets. We allow concepts within and across ontologies of the same dimension to be related through sets of (first-order) constraints Σ. Given an ontology $\mathbf{O}$, we write $\Sigma_\mathbf{O}$ to denote the constraints of $\mathbf{O}$. Constraints expressed between concepts of different ontologies are referred to as *articulations*. We call a set of articulation constraints $A = \Sigma_{\mathbf{O_1O_2}}$ an *alignment*, and refer to the ontologies in an alignment as $A.1$ and $A.2$. In this work, we only consider articulations between concepts that appear in ontologies of the same dimension, $dim(A.1) = dim(A.2)$. A set of alignments, $\mathcal{A} = \{A_1, \cdots, A_n\}$ where $\forall x,y \in \mathcal{A} : x \neq y \rightarrow dim(x.1) \neq dim(y.1)$, is called an *alignment set*.

We use the five basic relations of the *region connection calculus* RCC-5 for expressing constraints between ontologies [5,6]. Specifically, RCC-5 constraints relate pairs of (non-empty) concepts using the relations shown in Fig. 3. Any two concepts C_1, C_2 may be related by one or more of the five basic relations, e.g., $C_1 \{\subsetneq, \oplus\} C_2$ states that C_1 is either a proper subset of *or* overlaps C_2. Similarly, the constraint $C_1 \{\equiv, \subsetneq\} C_2$ represents the standard "isa" relation between concepts. Unless otherwise given (i.e., by default), any two concepts are assumed to be related by the disjunction of all five constraints, sometimes called the *universal relation*.

Merging Ontologies. Merging the context ontologies described here is a straightforward generalization of [7] which describes a method for merging taxonomies under RCC-5 articulations. Given two ontologies, O_1 and O_2 and an alignment $\Sigma_{\mathbf{O_1O_2}}$ describing the RCC-5 articulations between the concepts in O_1 and O_2, the merge algorithm

(i) *congruence* (ii) *proper inclusion* (iii) *proper inverse inclusion* (iv) *partial overlap* (v) *exclusion*

Fig. 3. The five basic, pairwise disjoint relations of the *region connection calculus*: (i) $N \equiv M$ stating that the set denoted by N is equivalent to M, (ii) $N \subsetneq M$ stating that N is a proper subset of M, (iii) $N \supsetneq M$ stating that M is a proper subset of N, (iv) $N \oplus M$ stating that N and M overlap, and (v) $N \,!\, M$ stating that N and M are disjoint, for two non-empty sets N and M. Further, for $N \oplus M$, it is assumed that at least one element is in the intersection.

begins by converting the ontologies to axioms in a first-order language (Φ_{O_1}, Φ_{O_2} and Φ_{O_1,O_2}) and using a reasoner to calculate the RCC-5 closure of the union

$$\Phi_M = \Phi_{O_1} \cup \Phi_{O_2} \cup \Phi_{O_1,O_2}$$

of the logic axioms describing the source ontologies and the articulations.

We then create a merged ontology by defining, if necessary, a new concept for each class of equivalent concepts, and rewriting the articulations determined by the RCC-5 closure with the new concept terms. C_M represents the set of predicate names in Φ_M. We define an equivalence relation on C_M such that:

$$a \sim b \text{ if } \Phi \models \forall x.a(x) \leftrightarrow b(x),$$

where the equivalence class of $a \in C$ is $[a] = \{x \in C \mid x \sim a\}$. We say that ontology O has *synonyms* if for some $a, b \in C$ with $a \neq b$ we have that $a \sim b$; otherwise O is called *synonym-free*. Using this definition we can construct a unique, synonym-free version of the initial merged ontology. We call this simplified version a *quotient ontology* $O_{/\sim}$ such that:

$$C_{/\sim} = \{[a] \mid a \in C\},$$
$$\Phi_{/\sim} = \{[\varphi] \mid \varphi \in \Phi\}.$$

Here for every FO formula φ, we define its quotient [φ] to be the formula where each atom a(x) has been replaced by the atom [a](x).

Data Set Merge Result and World Sets. The result of merging two data sets $M = Merge(D_1, D_2, \mathcal{A})$ is often a set of possible worlds. Each world represents an unambiguous data set that has as its metadata the merge of the source data sets' ontologies, and furthermore *respects* the observations in the source data sets. One data set D_1 respects the observations of another D_2 ($D_1 \prec D_2$) if $D_1 \models D_2$. For example, a data set derived from a possible world D_M respects the observations of one of its sources D_S if for every tuple t in D_S, we have that $D_M \models t$.

The main challenge addressed in this paper is (efficiently) determining the possible worlds. Once they have been found, the worlds can be conveniently represented using a single relation W [12]. We start with a set of possible worlds P, where each world $p \in P$ is an instance of a relation following the D(b,s,t,o) schema, where $|p|$ is the number of tuples in p. For each tuple in each possible world, we apply a function $f()$ to create

Table 2. A monadic logic encoding of articulations of the form $A \circ B$ where $\circ \in \{\equiv, \subsetneq, \supsetneq, \oplus, !\}$. This encoding applies when translating data sets into logic. When translating ontologies and articulations into logic for the purpose of checking their consistency or merging the ontologies, use the encoding in [6].

$\equiv$: $\forall x : A(x) \leftrightarrow B(x)$.	$!$: $\forall x : A(x) \rightarrow \neg B(x)$.
$\subsetneq$: $\forall x : A(x) \rightarrow B(x)$.	$\oplus$: $\exists x : A(x) \rightarrow ((A(x) \wedge B(x)) \mid (A(x) \wedge \neg B(x)))$.
$\supsetneq$: $\forall x : B(x) \rightarrow A(x)$.	$\exists x : B(x) \rightarrow ((A(x) \wedge B(x)) \mid (\neg A(x) \wedge B(x)))$.

a symbol representing the concatenation of the context attributes. For example, for the tuple $D(b_1, s_1, t_1, \mathtt{P})$, create a symbol $b_1 s_1 t_1$. We call the set of such symbols T. The attributes of the schema of W are the symbols in T, and its arity is $|T|$. We index each attribute in W with values $1 \leq i \leq |T|$.

The tuples in W are created as follows. For a given world $p \in P$ with tuples $\{t_1, ..., t_n\}$, let t_p be a tuple following the schema of W where for $1 \leq i \leq |T|, t_p(W_i) = 1$ if $\exists x \in p$ such that $f(x) = W_i$ and $o(x) = \mathtt{P}$; $t_p(W_i) = 0$ if $\exists x \in p$ such that $f(x) = W_i$; and $o(x) = \mathtt{N}$ and $p(W_i) = \perp$ otherwise.

Translation into Logic. To determine whether or not two data sets may be merged, to ensure the consistency of data sets, and to validate the result of the merge requires reasoning about the data sets, their ontologies, and the relationships between the ontologies. To provide this reasoning, we translate each of these elements into sets of first-order logic formulas.

Each record of a data set D induces a first-order logic formula as follows. A presence observation denoted by a record of the form $D(b, s, t, \mathtt{P})$ is represented by a formula

$$(\exists xyz) \;\; b(x) \wedge s(y) \wedge t(z) \wedge present(x, y, z)$$

where the relation $present(x, y, z)$ holds whenever the biological entity x was present at location y and time z.[4] The formula above states that a biological organism x of type b was observed within location y of type s and at time z of type t. Similarly, an absence observation denoted by a record $D(b, s, t, \mathtt{N})$ is represented by a formula

$$(\forall xyz) \;\; b(x) \wedge s(y) \wedge t(z) \rightarrow \neg present(x, y, z)$$

stating that for each biological entity x of type b, location y of type s, and time z of type t, x was not found within location y at time z. Note that this encoding of absence asserts the complete absence of entities of the given biological type throughout the given spatial and temporal contexts. We refer to the set of axioms reflecting the observations of a data set as Φ_{DI}.

We encode the constraints over the concepts in the ontologies using monadic logic. More specifically, we restrict the ontology constraints in Σ_O to relations from the RCC-5 algebra, plus an additional type of constraint called *coverage*. The coverage constraint states that one concept can be defined as the union of a set of concepts (e.g. $(\forall x) \;\; P(x) \leftrightarrow C_1(x) \vee \cdots \vee C_n(x)$.) We define Φ_O as the combined set of formulas generated by translating the RCC-5 constraints in Σ_O into monadic logic using the rules in

[4] Where the formula includes the S and T terms only if these are part of the presence-absence schema.

Table 2, plus additional coverage constraints. The RCC-5 based articulations between ontology concepts are also represented as monadic logic formulas Φ_A.

A complete data set, then, is defined as

$$\Phi_{DS} = \Phi_{DI} \cup \Phi_{O_1} \cup \cdots \cup \Phi_{O_n}$$

where n ranges over the ontologies referenced by the data set.

Merge-Compatible Data Sets. To determine whether or not two data sets may be merged, we calculate the absence closure for each data set, if required, and then translate the data sets into the first-order logic representation above, along with their ontologies and the alignment axioms relating the ontologies. We then apply a first-order reasoner to determine whether or not the combined axioms are consistent. The merge of two data sets Φ_M is the union of the formulas for each data set combined with the formulas derived from the RCC-5 articulations between the data set ontologies

$$\Phi_M = \Phi_{DS_1} \cup \Phi_{DS_2} \cup \Phi_{A_1} \cup \cdots \cup \Phi_{A_n}$$

where n ranges over the context attributes in the data sets.

Example (Merge-Compatible). Consider Fig. 1 without absence closure, and ontology alignment set $\mathcal{A} = \{\{A \equiv E; B \equiv F; C \equiv G\}, \{J \equiv M; K \equiv M; L \equiv M\}, \{R \equiv S\}\}$. In this simple example, merging the two data sets is straightforward, where the single merge result shown in Table 1(c) contains no BRU or DRU and represents all the observed data. Typically, however, merging two data sets does not result in a combined data set that is free of uncertainty, due to non-trivial ontologies and articulation constraints. In the following section we describe an approach for merging data sets when the merge cannot be satisfied by a single data set, and instead must be represented as a set of possible merges.

4 Merging Data Sets

Merging two data sets results in a set of possible merges, each representing an unambiguous data set that respects the observations in the source data sets. Before carrying out the merge, we determine the input data sets' merge compatibility. If the sets are merge compatible, we perform one of two types of merge. *Basic relation merges* (BRM) are those in which all the relations between concepts in the two data sets are drawn from the basic set relations. *Disjunctive relation merges* (DRM) are those that involve at least one disjunctive relation (e.g., $A\{\equiv, \supsetneq\}B$). This section proceeds by first describing how to check for merge compatibility. We then describe a naive algorithm for merging data sets, followed by two BRM algorithms, and then a description of how to perform a DRM.

4.1 Merge Compatibility and Absence Closure

For two data sets to be merge compatible, they must follow our schema, their ontologies must be consistent, the data must be consistent with the ontologies, the alignments

Algorithm 1: Merge Compatible

Input: Two data sets and a set of articulations between the ontologies
Output: `true` if the data sets are merge compatible, `false` otherwise

1. Determine consistency.
   (a) For each data set
      i. Calculate Φ_O for each ontology and check its consistency.
      ii. Calculate Φ_{DS} for the data set and check its consistency.
2. If each data set is consistent, check the alignment $\Phi_{O_1} \cup \Phi_{O_2} \cup \Phi_{A_{12}}$ between each pair of data set ontologies for consistency.
3. If each alignment is consistent, check the full merge Φ_M for consistency, applying absence closure if required.

Algorithm 2: Calculate Absence Closure Axioms

Input: A data set
Output: A set of logic axioms representing absence axioms

1. Create logic absence axioms $A = \{a_1, \cdots a_n\}$ for each possible combination of context attribute values $B \times S \times T$
2. For each row in the data set r_i, for each created absence axiom a_i:
   (a) if $r_i \rightarrow a_i$ remove a_i from A
   (b) if $a_i \rightarrow r_i$ remove a_i from A
3. Return A

between their ontologies must be consistent, and finally, the union of the logic axioms for each data set, their ontologies, and the ontology alignments must be consistent. These steps are outlined in Algorithm 1.

Consistency in the last step may be violated by contradictions introduced by explicit absence statements, as well as axioms introduced in absence closure. For example, in Fig. 1, if $D \equiv X$ where X is some known, but unreported rodent in data set 2's taxonomy, absence closure leads to a direct contradiction; data set 2 would state explicitly that X is absent, conflicting with the observed D in data set 1. Algorithm 2 provides a straightforward way of calculating these absence axioms. This algorithm first determines all possible cases in which presence might be observed within the given attribute contexts, and then rules out those cases that are implied by known observations, and also those that imply known observations.

4.2 The Naive BRM Algorithm

The most straightforward way to calculate the possible worlds is to create an initial world set (IWS) as described in Section 2, encode each world in logic, and test whether or not it is consistent with the formulas in Φ_M. This method, however, is both intractable and inefficient. A somewhat more efficient approach is to initially rule out impossible conditions in the IWS. For example, if an articulation holds that $A \subsetneq B$, any world in which the combined concept $A\bar{B}$ is either present or absent would be inconsistent with the articulation. Removing conditions containing such concepts reduces the size of the IWS and, as will be shown in Section 5, can generate possible worlds for small data sets. Table 3 lists the monadic logic formulas generated to test the possible world in Fig. 2(d) in which instances of taxa that are both A and B are present, as well as instances of taxa that are B but not A. The complexity of naive BRM algorithm comes primarily from the need to perform many (up to 2^n) monadic logic proofs, each of which is NEXPTIME [13].

Table 3. Monadic logic rules demonstrating the possibility of the data set in Fig. 2(d)

Axioms:			Conjecture:
$\forall x : A(x) \rightarrow B(x).$	$\exists x : AB(x).$	$\forall x : AB(x) \rightarrow (A(x) \wedge B(x)).$	$(\forall x : (A(x) \rightarrow B(x))) \wedge$
$\forall x : B(x) \leftrightarrow (AB(x) \vee \bar{A}B(x)).$	$\exists x : \bar{A}B(x).$	$\forall x : \bar{A}B(x) \rightarrow (\neg A(x) \wedge B(x)).$	$(\exists x : A(x)) \wedge (\exists x : B(x)).$
$\forall x : A(x) \leftrightarrow AB(x).$	$\neg\exists x : \bar{A}\bar{B}(x).$	$\forall x : \bar{A}\bar{B}(x) \rightarrow (\neg A(x) \wedge \neg B(x)).$	
$\forall x : A(x) \vee B(x).$			

Algorithm 3: General Basic Relation Merge (BRM-G)

Input: A naively merged data set.
Output: A possible world set representing each possible merge.

1. Create a new concept $c_1c_2\cdots c_n$ for those concepts that are equivalent according to the articulations. Replace all concepts contributing to the new concept with the new concept in Φ_M. Remove redundant formulas.
2. For each attribute $A \in \{B,S,T\}$:
   (a) Create an empty set P_A.
   (b) For each pair of rows (r_i, r_j) in the data set
      i. Let $c_i = A(r_i)$ and $c_j = A(r_j)$
      ii. If $c_i \subsetneq c_j$, add c_ic_j and $\bar{c}_ic_j$ to P_A.
      iii. if $A(c_i) \oplus A(c_j)$, add c_ic_j, $c_i\bar{c}_j$, $\bar{c}_ic_j$ to P_A.
      iv. if $A(c_i)\,!A(c_j)$ add $c_i\bar{c}_j$ and $\bar{c}_ic_j$ to P_A.
   (c) Repeat $|A|-2$ times, where $|A|$ is the number of concepts of attribute A in the data sets.
      i. Create empty set E_A
      ii. For each pair of concepts $c_i, c_j, i \neq j \in P_A$, add $compress(c_i, c_j)$ to E_A
      iii. set $P_A = E_A$
3. For each data set row r, for each attribute $A \in \{B,S,T\}$, for each term $p \in P_A$, if $A(r)$ appears positively in p then add p to $V_{A(r)}$.
4. Create a propositional logic statement that will generate the possible worlds: Create an empty sets A and H. For each observation r in each data set:
   (a) For each attribute $A \in \{B,S,T\} : D_A = \bigvee V_{A(r)}$
   (b) $C = \bigvee D_B \times D_S \times D_T$
   (c) If $O(r) = \mathtt{P}$, add C to A
   (d) If $O(r) = \mathtt{N}$ add the negation of C to A
   (e) Add C to H
5. Conjoin the elements in set A - this will be a propositional logic statement - the possible worlds are the models of this statement.
   (a) H contains the conditions in the header of the table
   (b) Create the rows: For each model, add a new row to the table where for each condition in H, if the condition holds in the model, put 1in the appropriate column, and add 0 otherwise.

4.3 General Basic Relation Merge (BRM-G)

The general basic relation merge (BRM-G) presented in Algorithm 3 applies when the data sets to be merged have no DRM, but may have BRM. The key steps to the BRM-G algorithm are calculating the columns of the PWS H, and the propositional formula Φ, the models of which represent the possible worlds in the PWS. The compress function in step 2(c)ii takes two combined concepts, both of length n. If the two combined concepts agree on $n-1$ concepts, the result is the concepts they agree on, plus the concepts they disagree on. For example, $compress(A\bar{B}\bar{C}, \bar{B}\bar{C}D)$ results in A$\bar{B}\bar{C}$D. The compress function also makes sure to not create any impossible combined concepts, such as ones that contain a term and its negation (e.g., $A\bar{A}$).

Example (Only one context domain). Consider a simplified version of Fig. 1 with only the biological attribute context, the observation data context, and the following alignment between the biological ontologies of the data sets: $\mathcal{A} = \{A \equiv E; B \equiv F; C \oplus$

Table 4. Possible worlds for Fig. 1 with just its biological attribute context and its data context. (a) shows a merge representing the (ambiguous) straightforward union of the data sets. (b) shows the PWS of unambiguous worlds. Tables (c) and (d) represent unambiguous merged data sets derived from the PWS.

$$H = \{AE, BF, C\bar{G}, CG, \bar{C}\bar{D}G, DG\}$$
$$\Phi = AE \wedge \neg BF \wedge (C\bar{G} \vee CG) \wedge (\bar{C}\bar{D}G \vee CG \vee DG) \wedge \neg DG$$

(a)

Taxon	O
A	P
B	N
$C\bar{G}$	P
D	N
E	P
F	N

(b)

World	AE	BF	C$\bar{\text{G}}$	CG	$\bar{\text{C}}\bar{\text{D}}$G	DG
1	1	0	1	1	1	0
2	1	0	1	0	1	0
3	1	0	1	1	0	0
4	1	0	0	1	1	0
5	1	0	0	1	0	0

(c)

Taxon	O
AE	P
BF	N
$C\bar{G}$	P
CG	N
$\bar{C}\bar{D}G$	P
DG	N

(d)

Taxon	O
AE	P
BF	N
$C\bar{G}$	N
CG	P
$\bar{C}\bar{D}G$	N
DG	N

$G; D \subsetneqq G\}$. A straightforward union of the biological concepts in this situation shown in Table 4(a) contains several problems. First, listing both A and E is redundant, as $A \equiv E$. More importantly, D and G have a $\subsetneqq$ relation between them, so the result in Table 4(a) still contains BRU. Finally, although C and G are both named pack rat, they are not equivalent terms as represented in Table 4(a).

Running the BRM-G against this example results in the H and Φ shown at the top of Table 4.[5] The PWS that results from these formulas is show in Table 4(b). The enumeration of all possible worlds shown in Table 4(b) indicates that the combined concept AE is present in all possible worlds (certainly present), while BF and DG are absent in all possible worlds (certainly absent). The situation is more complicated for concepts C and G. Table 4(c) and (d) give two of the possible merged data sets showing different possible configurations of C and G .

Example (Two context domains). Consider the taxonomic and spatial dimensions of the running example with the alignment $\mathcal{A} = \{\{A \equiv E; B \equiv F; C \oplus G; D \subsetneqq G\}, \{J \subsetneqq M; K \subsetneqq M; L \subsetneqq M\}\}$. Below are the columns of the PWS (each given a number), followed by the propositional formula describing the possible worlds.

$$\begin{aligned} H = \{&1 : AEJM, 2 : BFKM, 3 : CGJM, 4 : C\bar{G}JM, 5 : DGCM, 6 : AEKM, 7 : AELM, 8 : AEJ\bar{K}\bar{L}M, \\ &9 : BFJM, 10 : BFKM, 11 : BFLM, 12 : BFJ\bar{K}\bar{L}M, 13 : CGKM, 14 : CGLM, 15 : CGJ\bar{K}\bar{L}M \\ &16 : \bar{C}\bar{D}GJM, 17 : \bar{C}\bar{D}GKM, 18 : \bar{C}\bar{D}GLM, 19 : \bar{C}\bar{D}GJ\bar{K}\bar{L}M, 20 : DGJM, 21 : DGKM, 22 : DGJ\bar{K}\bar{L}M\} \end{aligned}$$

$$\Phi = 1 \wedge \neg 2 \wedge (3 \vee 4) \wedge \neg 5 \wedge (1 \vee 6 \vee 7 \vee 8) \wedge \neg(9 \vee 10 \vee 11 \vee 12) \wedge (3 \vee 13 \vee 14 \vee 15 \vee 16 \vee 17 \vee 18 \vee 19 \vee 20 \vee 21 \vee 5 \vee 22)$$

Φ has 24576 ($< 2^{15}$) models, each of which is a possible merged data set. This may seem like a large number, but it is considerably smaller than the number of possible worlds in the initial world set ($2^{2^{(7+5)}} = 2^{4096}$). The BRM-G is considerably more efficient than the naive algorithm because it involves a single NP-complete SAT proof,

[5] To save space and improve legibility, the complete combined concepts are not given in the table. For each abbreviated concept in Table 4, the full combined concept can be determined by adding the negated form of all the concepts in the data sets not mentioned in the combined concept. For example, the abbreviated combined concept AE in Table 4 stands for $A\bar{B}\bar{C}\bar{D}E\bar{F}\bar{G}$.

rather than up to 2^n NEXPTIME monadic logic proofs. The algorithm itself, however is $O(2^n)$ due to the need to run the compress function multiple times. Each time compress is run, the size of changes P, and in the worst case, when all the concepts overlap, $|P| = 2^n$ the final time compress is run.

4.4 The Basic Relation Merge for Unambiguous Data Sets (BRM-U)

The BRM-G algorithm works when source data sets contain BRU. The BRM-U algorithm presented here is far more efficient, but only works when the source data sets have no BRU (or DRU). The only difference between the BRM-G and BRM-U algorithms is in how the compress function creates combined concepts. In the BRM-G algorithm, compress is run $n-2$ times where n is the number of distinct concepts in the source data sets and the input can be as large as 2^n combined concepts. In the BRM-U algorithm, on the other hand, compress is only run once on $\binom{n}{2}$ combined concepts. This is possible because when a data set has no BRU, and the equivalent concepts have been combined into a single concept (step 1 in Algorithm 3), any combined concept can contain at most one pair of non-negated concepts (one concept from each data set). After a single run of compress, each combined concept in P_A will be three concepts long, and all the feasible pairs of non-negated concepts will have been found. After this single run of compress, each combined concept is then padded with the negated version of all the $n-3$ concepts that are not yet in that combined concept. The resulting compress algorithm is $O(n^2)$ as it involves a single pass through $\binom{n}{2}$ combined concepts determined in step 2b of Algorithm 3. The entire BRM-U algorithm is $O(n^2)$ except for the single SAT proof at the end, which is NP-complete.

4.5 Merging under Disjunctive Relation Uncertainty

The algorithm described here applies to merges involving both BRU and DRU. The strategy is to divide alignments containing disjunctions into several alignments containing no disjunctions, determine the PWS for each BRM situation, and combine the results. Dividing disjunction containing alignments into several basic alignments is an expensive process. Consider, e.g., the taxonomy alignment in Fig. 4, which contains two disjunctive relationships $\{\mathrm{A}\ \{\equiv,\subsetneq\}\ \mathrm{E}\ ;\ \mathrm{B}\ \{\equiv,\supsetneq\}\ \mathrm{F}\}$ and represents "isa" relations as $\subsetneq$. To decompose this disjunction-containing alignment into alignments containing only basic relations, one might try simply multiplying out the disjunctive relationships, creating four possible alignments. If, however, the following additional constraints hold in the alignment: $X \equiv A \vee B \vee C \vee D$; $Y \equiv E \vee F \vee G$, and sibling concepts are disjoint, two of the four possible alignments ($\{A \subsetneq E; B \equiv F\}$ and $\{A \equiv E; B \supsetneq F\}$) are ruled out.

With this in mind, the disjunction containing alignment above can be divided into two consistent alignments containing only basic relations: one equivalent to the one described in Section 4.3, and the other following the alignment: $\mathcal{A} = \{A \subsetneq E; B \supsetneq F; C \oplus G; D \subsetneq G\}$. Table 5(a) shows the complete PWS for the disjunction containing alignment. The column to the side of the table records which alignment applies to the given row: alignment 1 is where $A \equiv E$ and $B \equiv F$, and alignment 2 is where $A \subsetneq E$ and $B \supsetneq F$. This additional information may be considered provenance; the actual metadata

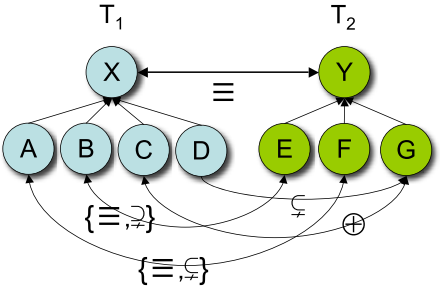

Fig. 4. When sibling concepts are disjoint and parents contain no instances not found in their children, this disjunctive relation containing alignment has two basic relation interpretations

Table 5. PWS for Section 4.5(a) and two data sets derived from the PWS: world 5 in (b) and world 15 in (c)

(a)

World	AE	BF	$\bar{A}E$	$B\bar{F}$	$C\bar{G}$	CG	$\bar{C}\bar{D}G$	DG	P
1	1	0	⊥	⊥	1	1	1	0	1
2	1	0	⊥	⊥	1	0	1	0	1
3	1	0	⊥	⊥	1	1	0	0	1
4	1	0	⊥	⊥	0	1	1	0	1
5	1	0	⊥	⊥	0	1	0	0	1
6	1	0	1	0	1	1	1	0	2
7	1	0	1	0	1	0	1	0	2
8	1	0	1	0	1	1	0	0	2
9	1	0	1	0	0	1	1	0	2
10	1	0	1	0	0	1	0	0	2
11	1	0	0	0	1	1	1	0	2
12	1	0	0	0	1	0	1	0	2
13	1	0	0	0	1	1	0	0	2
14	1	0	0	0	0	1	1	0	2
15	1	0	0	0	0	1	0	0	2

(b)

Taxon	O
AE	P
BF	N
$C\bar{G}$	N
CG	P
$\bar{C}G$	N
DG	N

(c)

Taxon	O
AE	P
BF	N
$\bar{A}E$	N
$B\bar{F}$	N
$C\bar{G}$	N
CG	P
$\bar{C}G$	N
DG	N

for the merged data sets is still the merged ontologies of the original data sets. The ⊥ seen in worlds 1 through 5 indicates that the combined concept does not exist in that world.

Table 5 contains some subtly different merges. For example, in merge 15 (shown in Table 5(c)), no instances of $\bar{A}E$ were seen, while in merge 5 (shown in Table 5(b)) there is no such thing as an instance of $\bar{A}E$.

5 Evaluation

Here we evaluate the efficiency of the basic relation merge, which is the core of our data set merging methodology. We implemented the naive, BRM-G, and BRM-U data set merge algorithms in Python, and compared them using two types of data sets: those containing no BRU or DRU (the *unambiguous inputs* condition), and those that contained BRU (the *ambiguous inputs* condition).

Table 6. Average run times in seconds for the naive algorithm and two versions of the BRM algorithm using data sets of between 3 and 9 concepts in two conditions: (a) where the data set contains basic relation uncertainty, and (b) where the input data sets do not contain basic relation uncertainty. Run times in seconds for larger data sets using the BRM-U algorithm are shown in (c). The average number of worlds generated by data sets with mixed relations is shown in (d).

(a) Ambiguous Inputs

Data Items	3	4	5	6	7	8	9
naive	8.83	23.57	32.49	> 60	> 60	> 60	> 60
BRM-G	0.03	0.04	0.08	0.34	1.86	16.88	23.91

(b) Unambiguous Inputs

Data Items	3	4	5	6	7	8	9
naive	1.73	6.46	18.09	> 60	> 60	> 60	> 60
BRM-G	0.03	0.04	0.05	0.11	0.37	2.06	7.24
BRM-U	0.03	0.04	0.04	0.05	0.05	0.06	0.06

(c) BRM-U with larger unambiguous input data sets

Data Items	25	50	75	100	200	300	400	500
BRM-U	0.19	0.37	0.86	2.32	24.54	121.25	359.24	824.92

(d) Worlds generated by mixed relation data sets

Data Items	3	4	5	6	7	8	9
Ambiguous Inputs	22	12	26	81	266	173	180
Unambiguous Inputs	2	3	7	86	58	165	224

Pairs of data sets and the alignments between their concepts were generated randomly. Each data set had only one context attribute, and each observation of the data sets was recorded as present. Generating interesting large consistent alignments between data set concepts proved challenging. While it is simple to generate large alignments in which all the relations are of one type (e.g., all equivalent, all overlapping, all disjoint), generating consistent alignments that mix relations is computationally expensive. To address this issue, we generated alignments of up to 9 concepts in which the non-disjoint relationships were either all-$\oplus$, all-$\subsetneq$, or had mixed relations, including $\equiv$, $\oplus$, $\subsetneq$, and $\supsetneq$. We found the same patterns of results held in the all-$\oplus$, all-$\subsetneq$, and mixed conditions, so for data sets of fewer than 10 concepts, we report the average results of these three types of data sets. Results of 10 or more concepts are the average of the all-$\oplus$ and all-$\subsetneq$ conditions. Each condition was run three times, and only marginal variance was found between runs.

The naive algorithm runs employed the first-order reasoner iProver 0.7 [15] to test whether a given world qualifies as a possible merge. Comparisons between iProver and several other available first-order reasoners showed iProver to be the fastest to solve our class of problem. The BRM algorithm tests employed the c2d [14] reasoner to check the satisfiability of the propositional statement that determines the possible merges, and to generate and count models of the statement. c2d has the advantage of providing polynomial-time model counting.

As may be seen in Table 6, the naive implementation performs poorly, taking over a minute to generate possible worlds for data sets with more than 6 concepts. In the ambiguous input condition, the BRM-G algorithm performs considerably better. However, the time to generate the possible worlds still grows exponentially with the size of the input. The unambiguous inputs condition shows the same pattern for the naive and general BRM algorithms. However, the BRM-U algorithm performs comparatively well, providing both a feasible and efficient method for generating the possible data set merges. Table 6(c) shows how the BRM-U algorithm scales to up to 500 concepts. The presence data sets we have seen have listed fewer than 300 concepts, and the largest pair

of articulated biological taxonomies we have seen to date [17] has comprised 360 concepts, so the algorithm scales well to the currently available real-world data. Table 6(d) gives the average number of worlds generated by the data sets with mixed relations.

6 Related Work and Conclusion

This paper has described a framework and algorithms for merging data sets when the domains of attributes overlap and contain uncertainty. We have shown that no single merge, except in trivial cases, can satisfy all the requirements of a data set merge, and multiple merges must be represented. We have given a possible worlds semantics for such data sets, and algorithms for constructing these possible worlds when ambiguity arises during the merge. We have also presented an efficient algorithm for performing the merge when ambiguity is due to articulations (i.e., source data sets do not contain ambiguity).

The three areas most similar to the current work are traditional data integration, data fusion, and ontology merging. In traditional data integration [16], two or more databases with different schemas are combined through the definition of a common (i.e., global) schema. The current work, on the other hand, focuses on merging data sets when the schemas of data sets are the same, but the domains of the schema attributes may be different. Another difference is that in traditional data integration, the data themselves are generally not considered; integration happens at the schema level. In the current work, however, the alignments between the domains of the data set attributes impact the interpretation of the data itself. Data fusion [18] tasks typically involve integrating multiple types of information about the same objects. The data fusion setting differs from the current one in that we are merging data sets that contain the same type of data: presence data, in this case. Furthermore, our observations are about sets of objects rather than individuals. Ontology merging [19,20,21], like traditional data integration tasks, focuses on the schema level rather than the instance level. The work in [7], which describes how to merge taxonomies that have been aligned with RCC-5 relations, is more similar to ontology merging. As we have seen here, merging taxonomies is just the first step in merging taxonomically organized data sets.

This work can be expanded in several directions. First, although we use RCC-5 to describe relations between attribute domains, there are other algebras that may be more suited to specific domains. For example, RCC-8 may be a better language to describe relations between spatial regions. Allen's interval calculus is more suited for the temporal dimension. The types of languages used constrain the questions that may be asked. In this work, we are satisfied to ask questions that are suitable for RCC-5 articulated domains. In the future, other languages should be applied. Second, in the current work, domains are independent. However, in general this may not be the case. For example, one taxonomic alignment may apply in one spatial region, while a second taxonomic alignment may apply in a different region. Extending the algorithms to deal with this extra complexity is not straightforward. Third, we have only considered presence data here. As we have seen, merging data sets with even this limited type of data is complicated. However, data sets typically contain data other than simple presence data, so this work should be extended to include other types of measurements. Finally, the work

must be evaluated by testing its utility for the people who currently spend their time integrating data sets by hand. This evaluation will no doubt generate interesting new avenues of study.

References

1. Cliff, A.D., Haggett, P., Smallman-Raynor, M.: The changing shape of island epidemics: historical trends in icelandic infectious disease waves. J Hist Geogr. 1902–1988 (2009)
2. Berkley, C., Jones, M., Bojilova, J., Higgins, D.: Metacat: a schema-independent xml database system. In: SSDBM, pp. 171–179 (2001)
3. Thau, D.: Reasoning about taxonomies and articulations. In: EDBT Workshops, pp. 11–19 (2008)
4. Brachman, R.: What is-a is and isn't: An analysis of taxonomic links in semantic networks. IEEE Computer 16, 30–36 (1983)
5. Randell, D.A., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: KR, pp. 165–176 (1992)
6. Thau, D., Ludäscher, B.: Reasoning about taxonomies in first-order logic. Ecological Informatics 2(3), 195–209 (2007)
7. Thau, D., Bowers, S., Ludäscher, B.: Merging taxonomies under RCC-5 algebraic articulations. In: Proceedings of the CIKM ONISW Workshop, pp. 47–54 (2008)
8. Lewis, C., Langford, C.: Symbolic Logic, 2nd edn. Dover, New York (1959)
9. Abiteboul, S., Kanellakis, P.C., Grahne, G.: On the representation and querying of sets of possible worlds. In: SIGMOD, pp. 34–48 (1987)
10. Antova, L., Jansen, T., Koch, C., Olteanu, D.: Fast and simple relational processing of uncertain data. In: ICDE, pp. 983–992 (2008)
11. Bowers, S., Madin, J.S., Schildhauer, M.P.: A conceptual modeling framework for expressing observational data semantics. In: Li, Q., Spaccapietra, S., Yu, E., Olivé, A. (eds.) ER 2008. LNCS, vol. 5231, pp. 41–54. Springer, Heidelberg (2008)
12. Antova, L., Koch, C., Olteanu, D.: World-set decompositions: Expressiveness and efficient algorithms. In: Schwentick, T., Suciu, D. (eds.) ICDT 2007. LNCS, vol. 4353, pp. 194–208. Springer, Heidelberg (2007)
13. Bachmair, L., Ganzinger, H., Waldmann, U.: Set constraints are the monadic class. Logic in Computer Science, 75–83 (1993)
14. Darwiche, A.: New advances in compiling cnf into decomposable negation normal form. In: ECAI, pp. 328–332 (2004)
15. Korovin, K.: iProver – an instantiation-based theorem prover for first-order logic (system description). In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 292–298. Springer, Heidelberg (2008)
16. Lenzerini, M.: Data integration: A theoretical perspective. In: PODS (2002)
17. Peet, R.: Taxonomic concept mappings for 9 taxonomies of the genus Ranunculus published from 1948 to 2004. Unpublished data set (2005)
18. Vasseur, P., Mouaddib, E.M., Pégard, C.: Introduction to multisensor data fusion. In: Zurawski, R. (ed.) The Industrial Information Technology Handbook, pp. 1–10. CRC Press, Boca Raton (2005)
19. Noy, N.F., Musen, M.A.: PROMPT: Algorithm and tool for automated ontology merging and alignment, pp. 450–455. AAAI, Menlo Park (2000)
20. McGuinness, D.L., Fikes, R., Rice, J., Wilder, S.: An environment for merging and testing large ontologies. In: ECAI (2000)
21. Stumme, G., Maedche, A.: Ontology merging for federated ontologies on the semantic web. In: FMII, pp. 413–418 (2001)