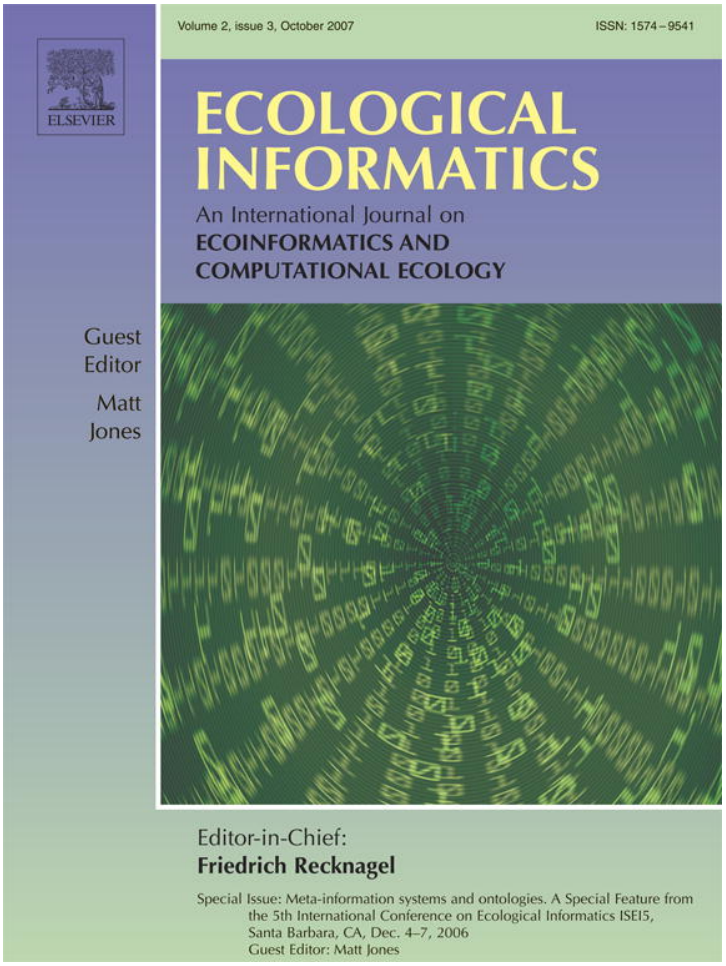


ECOLOGICAL INFORMATICS 2 (2007) 195–209

available at www.sciencedirect.com

www.elsevier.com/locate/ecolinf

Reasoning about taxonomies in first-order logic☆

*David Thau, Bertram Ludäscher**

Department of Computer Science, University of California, Davis, United States

ARTICLE INFO

Article history:
Received 7 February 2007
Received in revised form
19 July 2007
Accepted 30 July 2007

Keywords:
Taxonomy
Automated deduction (reasoning)
First-order logic constraints

ABSTRACT

Experts often disagree about the organization of biological taxa. The shifting definitions of taxonomic names complicate otherwise simple queries concerning these taxa. For example, a query such as "how many occurrences of specimens in genus *G* are recorded in database *D*" will return different answers depending on whose definition of genus *G* is used. In our proposed framework, taxonomic classifications of multiple experts are captured using first-order logic (FOL). Specifically, taxonomies, and relationships between them, are viewed as sets of first-order formulas, constraining the possible interpretations of names and concepts in the taxonomies. The formalization of taxonomies and the relationships between them via our FOL language $\mathcal{L}_{\text{tax}}$ allows us to clarify (a) what it means for a taxonomy to be consistent, (b) to be inconsistent, (c) whether a new relationship between two taxa (e.g., a congruence $A \equiv B$) is implied, thus "filling logic gaps", and (d) whether two taxonomies from different authorities, together with a taxonomy mapping (articulation) from a third authority, are mutually consistent.

We illustrate our logic-based formalization and the resulting opportunities for automated reasoning support for biological taxonomies using examples involving the classification of a genus of plants. We elaborate on (a–d) above and give some example derivations in logic. We also show that while reasoning in $\mathcal{L}_{\text{tax}}$ is decidable, it might still be computationally hard (e.g., NP-complete) and thus infeasible over large taxonomies and articulations. By employing results from the spatial algebra RCC-5, we identify an important class of efficient taxonomy articulations, i.e., whose satisfiability can be checked in polynomial time.

1. Introduction

Experts often disagree on the definition of terms within their fields. This can make discovery and integration of information using these terms difficult. In biology, e.g., experts may disagree upon which species are part of a given genus. For example, Benson, 1948 (Benson, 1948) claims that the genus *Ranunculus* contains a species named *Ranunculus cooleyae*. A second expert, Kartesz, 2004 (Kartesz, 2004) argues that this species belongs in the genus Kumlienia and should therefore be named Kumlienia cooleyae. Somebody querying a database about species may want to know how many examples of plants in the genus *Ranunculus* can be found in the database. Unfortunately, the answer depends on which taxonomy the user chooses.

The variability in definitions of taxonomic names is a well-known problem. A number of researchers have attempted to quantify the differences between taxonomic opinions using set-theoretic notions, e.g., see (Berendsohn, 1995; Franz et al., 2006; Koperski et al., 2000, and Weakley, 2006). A given taxon, e.g. *Ranunculus*, circumscribes a set of organisms. The sets of organisms described by two taxa A and B may be equal ($A \equiv B$), contained in one another ($A \subsetneq B$ or $A \supsetneq B$), disjoint ($A \,!\, B$), or they may partially overlap ($A \oplus B$). These set-theoretic descriptions

☆ Work supported by NSF award DBI-0533368: Science Environment for Ecological Knowledge (SEEK).

* *Corresponding author.*
E-mail addresses: thau@cs.ucdavis.edu (D. Thau), ludaesch@cs.ucdavis.edu (B. Ludäscher).

1574-9541/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.ecoinf.2007.07.005

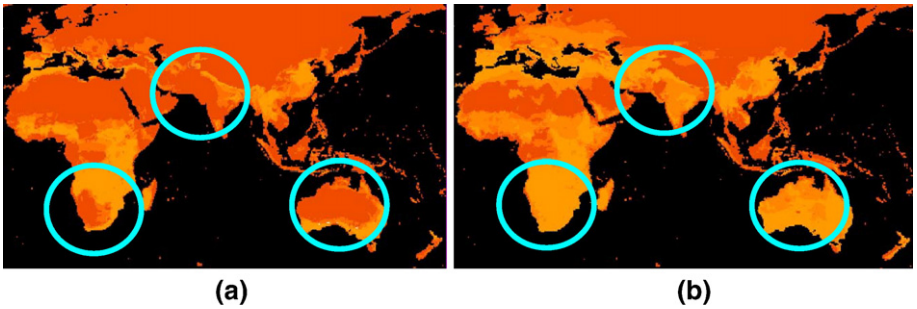


Fig. 1 – Predicted distribution for Anhinga melanogaster according to (a) Clement's Birds of the World, 4th edition (Clement, 1991) and (b) Clement's Birds of the World, 5th edition (Clement, 2001).

of taxa have been used to analyze differences in taxonomic opinions. Geoffroy and Berendsohn (Geoffroy and Berendsohn, 2003) analyzed Koperski's data (Koperski et al., 2000), comparing 11 treatments of German mosses and found that only 207 of the 1548 taxa analyzed have not changed in some way since 1927. Franz, Peet, and Weakley (Franz et al., 2006) Performed a similar analysis of five treatments of weevils and found that only 52% of taxa with the same name circumscribed the same set of weevils, and 26% of taxa with different names actually circumscribe the same set of weevils.

These studies highlight the limitations of using taxon names as identifiers for species. If two ecologists provide abundance data for the taxon *Ranunculus*, it is unclear whether they both have the same definition of *Ranunculus* in mind when conducting their studies. For this reason, it is important for researchers to state specifically which definition of *Ranunculus* was used when attaching the name *Ranunculus* to their dataset. This observation has contributed to the advocacy of using taxonomic concepts rather than scientific names only when identifying taxa (Berendsohn, 2003; Franz et al., 2006; Kennedy et al., 2005). A taxonomic concept is (minimally) a combination of a taxonomic name and a citation (authority) specifying which version of the taxonomic name is meant.

1.1. *Example 1*

Imagine attempting to predict the geographic distribution of a taxon, for example the species *Anhinga melanogaster*. The Global Biodiversity Information Facility (GBIF) provides a good initial source of locality information for this taxon. Fig. 1 (a) shows a species range prediction for *Anhinga melanogaster* generated using 100 iterations of the GARP algorithm run within a Kepler workflow (Ludascher et al., 2006). Lighter shades represent greater probability of presence.

Note that there appears to be a low probability of finding *Anhinga melanogaster* in South Africa, central Australia, western India, Pakistan and Afganistan. However, a recent version of Clement's Birds of the World lumps the species *Anhinga rufa* and *Anhinga novaehollaniae* into *Anhinga melanogaster*. An adherent to this taxonomy would merge the locality data of these three species, resulting in the species range prediction shown in Fig. 1 (b). Knowing the relationships between different versions of the same taxon has a clear impact on predictions about that taxon. Our logic-based approach can help determine when data classified using different taxonomies may be merged.

1.2. *Example 2*

Consider two datasets involving abundances of various species of the plant genus *Ranunculus* (which contains the buttercups), in two transects of a given locality. The first dataset (Table 1) represents observations of Ranunculi taken by a person O_1 who used a field guide based on Benson, 1948. The second dataset (Table 2) represents observations of Ranunculi taken in the same location, by a different person O_2, 6 months later. Observer O_2 used a field guide based on Kartesz, 2004. Assume that both studies were interested in documenting all species of *Ranunculus* observed in each locality.

We can ask a number of queries over these datasets, e.g., "What was the average number of *Ranunculus arizonicus* observed in transect 1 of LTER site 1?" Observer O_1 (using Benson, 1948) found 30 examples of *R. arizonicus* in transect 1, while O_2 (using Kartesz, 2004) found 6 examples of *R. arizonicus* in the same spot. Given no information about the relationship between Benson's and Kartesz's concepts of *R. arizonicus*, we cannot safely average over these two datasets:

(i) It may be that both O_1 and O_2 would have agreed that every observed *R. arizonicus* was in fact an *R. arizonicus*. We could assume this if Benson's and Kartesz's concepts of *R. arizonicus* were known to be equivalent. In other words, every plant classified by Benson, 1948 as

Table 1 – Dataset with abundance counts by observer O_1

Species	Abundance	Locality	Transect	Date
Ranunculus arizonicus (Benson, 1948)	30	LTER site 1	1	January 1, 2005
Ranunculus acriformis (Benson, 1948)	12	LTER site 1	1	January 1, 2005
Ranunculus arizonicus (Benson, 1948)	8	LTER site 1	2	January 1, 2005

Table 2 – Dataset with abundance counts by observer O_2

Species	Abundance	Locality	Transect	Date
Ranunculus arizonicus (Kartesz, 2004)	6	LTER site 1	1	June 22, 2005
Ranunculus aestivalis (Kartesz, 2004)	18	LTER site 1	1	June 22, 2005
Ranunculus glabberimus (Kartesz, 2004)	3	LTER site 1	2	June 22, 2005

R. arizonicus would have also been classified by Kartesz, 2004 as *R. arizonicus* and vice versa. In this case, the answer to our query is that there were on average 18 ($=\frac{30+6}{2}$) *R. arizonicus* seen in transect 1.

(ii) Alternatively, Benson's and Kartesz's concepts of *R. arizonicus* could be so different that O_2 would have classified every *R. arizonicus* of O_1 as some other species, and similarly, O_1 would have classified every *R. arizonicus* of O_2 as another species. In this case, the abundance data for *R. arizonicus* in the two datasets refer to two distinct plant species, rendering an average of the two abundances meaningless.

(iii) Finally, assume that based on scientific literature, we may ascertain that all examples classified as *R. arizonicus* by Kartesz, 2004 would have also been classified as *R. arizonicus* by Benson, 1948, but not vice versa, denoted[1] *R. arizonicus* $^{K04} \subsetneq$ *R. arizonicus* ^{B48}. Furthermore, assume there are no other types of *Ranunculus* considered by Kartesz, 2004 to be *R. arizonicus*. In this case, we could consider the numbers in the two datasets to be comparable according to the definition of Benson, 1948. Thus, we could answer the query by concluding that on average 18 examples of *R. arizonicus* ^{B48} were spotted in 2005. However, we cannot state an average in terms of the Kartesz, 2004 definition of *R. arizonicus* because some of the 30 *R. arizonicus* ^{B48} observed by O_1 may not be considered *R. arizonicus* ^{K04}.

Clearly, if we do not know the relationship between Benson's and Kartesz's concepts of *R. arizonicus*, the only accurate answer to the query is that the average number is uncertain because the observers collecting the datasets used different identification guides. However, if we knew the relationship between these concepts, we could give a more precise answer.

1.3. *Example 3*

Now consider a second query: "What is the average number of *Ranunculus acriformis* in transect 1 in 2005?" Without additional information, we only know that O_1 saw 12 examples in January while O_2 saw none. If we assume that Benson's and Kartesz's concepts of *R. acriformis* are equivalent, then the answer would be that an average of 6 examples were seen. However, if we believe that every example of *Ranunculus aestivalis* (Kartesz, 2004) is also an example of *R. acriformis* (Benson, 1948), we could aggregate the 18 observations of *R. aestivalis* ^{K04} in Table 2 with the 12 observations of *R. acriformis* ^{B48} in Table 1 to state that there were an average of 15 ($=\frac{12+18}{2}$) observations of *R. acriformis* ^{B48} in 2005. Again, because some examples of *R. acriformis* ^{B48} may not be examples of *R. aestivalis* ^{K04}, we cannot aggregate in that direction. Here, then, is another example where information about the relationship between taxonomic concepts is necessary to give a precise answer to the query.

Note that the datasets above included information about which taxonomic concept was used when collecting the data. In a less fortuitous (and not uncommon) setting, the datasets may use only taxonomic names without specifying which definition or authority of the name was meant. For example, all the species in the two datasets might be simply labeled by their species names. In this case, determining how the datasets may be merged becomes even more difficult since it is unclear how the taxa relate to each other. On the other hand, if a database curator can identify which taxonomic concept (i.e., which name plus definition or authority) was used when collecting the data, a logic-based framework can incorporate established relations between different taxonomies to guide the data integration process.

1.4. *Example 4*

Knowing how taxa in different taxonomies relate may not be enough to determine whether or not data classified using those taxonomies may be combined. Consider the abstract taxonomies in Fig. 2. Fig. 2a depicts two taxonomies, T_1 and T_2. In T_1, *B* and *C* are child taxa of *A*, and in T_2, *E* is a child of *D*. Fig. 2b shows these same two taxonomies, but with additional cross-taxonomy assertions. These assertions use set-theoretic notation to state that *A* and *D* are equivalent ($A \equiv D$ means all elements of *A* are also elements of *D* and vice versa), and *B* is a subset of *E* ($B \subset E$ means all elements of B are also elements of E, but not necessarily the other way around). Given these taxonomies and the relations between them can we deduce that $C \subset E$ is implied (Fig. 2c)? Is it consistent to assert $C \equiv D$, or $E \equiv A$ (Fig. 2d and e, respectively)? The answers to these questions depend on various (usually latent) assumptions about T_1 and T_2. Once we spell out these latent taxonomic assumptions (LTAs) in logic, all such questions have unambiguous answers (see Appendix A.3).

1.5. *Related work*

The notion of taxonomic concepts has been applied in a number of databases and data sharing schemes. For example, in 2005 TDWG (Taxonomic Data Working Group, 2006), an international not-for-profit organization which develops standards for biodiversity data, ratified the Taxonomic Concept Schema (TCS) (Taxonomic Concept Schema, 2006).

[1] The superscript KO4 and B48 indicate the authorities KARTESZ, 2004 and BENSON, 1984, respectively.

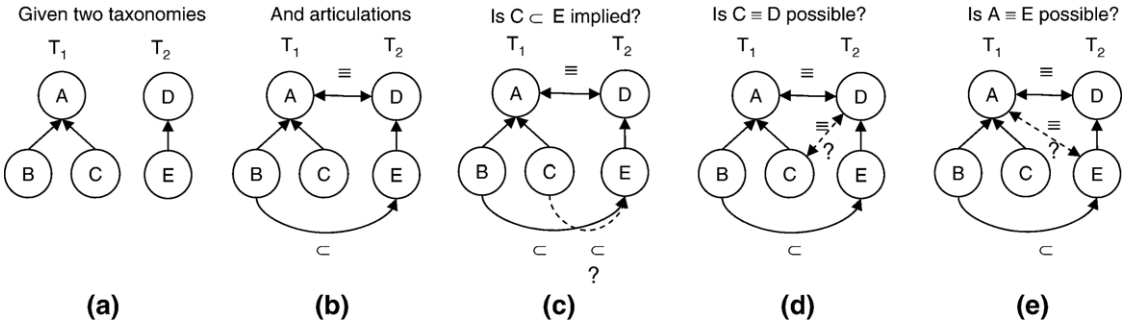


Fig. 2 – Potential questions about articulations between taxonomies.

TCS is an XML Schema which defines a syntax for describing taxonomic concepts. The TCS includes a list of terms which may be used to define the relationships between two different taxonomic concepts. This list includes set-theoretic terms, such as "is congruent to" and "excludes". However, some of the terms are not well defined. For example, it is unclear whether "is included in" means proper subset ($\subsetneq$), or if it means subset ($\subseteq$), which includes the possibility that both sets are equal. It also includes more vague relationships such as "has synonym" (e.g., imported from NCBI sources). These vague relationships are needed in TCS, as its goal is to provide a standard for information providers to communicate information about their data. If an information provider wants to use a mathematically imprecise relationship such as has synonym, TCS must support that relationship. Unless the meaning of such relations is specified more precisely, however, their utility for automated reasoning will be diminished.

Beach et al. (Beach et al., 1993) introduced and Berendsohn (Berendsohn, 1995) elaborated the notion of potential taxon, which identifies a taxonomic concept by referencing the context in which the name is used; e.g., *Hypnum flagellare* Dicks. sec. Mönkemeyer 1927. This notion is central to the MoReTaX project (Berendsohn, 2003), where potential taxa are considered as sets of objects, and the relationships between them are described in precise set-theoretic terms. The five so-called basic relations which may hold between any two potential taxa (or, in fact, any two non-empty sets) A and B are: (i) congruence ($A \equiv B$), (ii) proper inclusion ($A \subsetneq B$), (iii) proper inverse inclusion ($A \supsetneq B$), (iv) partial overlap ($A \oplus B$), and (v) exclusion (disjointness) ($A \,!\, B$). Geoffroy and Güntsch (Geoffroy and Güntsch, 2003) study the problem of propagating knowledge about such binary relationships between taxa: e.g., what can we say about the relationship between potential taxa A and C, provided we only know that $A \supsetneq B$ and $B \oplus C$? Inspection of all possibilities allows one to deduce that $A \supsetneq C$ or $A \oplus C$, but none of the other three options $\equiv$, $\subsetneq$, or ! is possible between A and C. Thus, the authors study combined relationships (i.e., disjunctions of basic relations: e.g., $\{\supsetneq \oplus\}$ denotes the disjunction $X \supsetneq Y \vee X \oplus Y$) and demonstrate how these may be composed to propagate taxonomic knowledge in a potential taxon graph. A generalized path in such a graph bundles all existing simultaneous paths between two nodes (say A and C) and can employ 'strong agreement' (conjunction of expert knowledge on simultaneous paths) or 'weak agreement' (disjunction of such paths). Rules for knowledge propagation in a taxon graph are given as if-then rules, embedded in the MoReTaX system; thus, computing with taxon relations is handled programmatically.

1.6. *Contributions*

We develop a first-order logic (FOL) framework for expressing set-theoretic relationships between taxa and for reasoning with those relationships. Our work can be seen as an "FOL grounding" of the framework by Berendsohn et al. In particular, we consider taxonomies as sets of constraints in a first-order language $\mathcal{L}_{\text{tax}}$. Then, questions related to 'navigating the potential taxon graph', 'concatenating relationships', etc. (Geoffroy and Güntsch, 2003) can be recast in a standard FOL framework. This formalization has many advantages, e.g., it allows us to precisely state what it means for a taxonomy to be consistent or inconsistent, and to recast the problem of 'closing gaps the experts left' (Geoffroy and Güntsch, 2003) as a logic implication problem. Most importantly, it allows us to apply a rich set of results and techniques from computer science (logic, automated deduction, complexity theory). In particular, we show that reasoning in $\mathcal{L}_{\text{tax}}$ is decidable but hard in general. For a special case, i.e., reasoning with constraints that are of a certain form only, a tractable (i.e., polynomial) and practical class of taxonomy constraints (called $\mathbb{R}_5^{28}$, a subalgebra of RCC-5; see Section 4.2.1) is identified. Our FOL framework also naturally accomodates mappings (articulations) between different taxonomies, turning "conflict resolution" among several expert opinions into a special case of reasoning in $\mathcal{L}_{\text{tax}}$. We also show that being able to state the usual pairwise set-theoretic relationships between taxa does not resolve all reasoning problems with taxonomies. Also of critical importance are latent taxonomic assumptions (LTAs), e.g., whether sibling disjointness is assumed for children of the same taxon, or whether coverage of a taxon with respect to its children is complete.

1.7. *Organization*

Section 2 presents preliminaries on taxonomies and some basic first-order logic terminology. In Section 3 we provide the basic framework for modeling taxonomies as logic constraints in a fragment $\mathcal{L}_{\text{tax}}$ of FOL. Section 4 considers some of these constraints in more detail, including common (latent) taxonomic assumptions. Mapping constraints between taxonomies (articulations) and efficient reasoning with them are

the focus of Section 5. Conclusions and future work are discussed in Section 6.

2. Preliminaries

2.1. *Taxonomies as trees and isa-hierarchies*

Taxonomy as a discipline is the practice and science of classification[2]. As a mathematical structure, a taxonomy $T=(\mathbf{N},\mathbf{E})$ is a rooted tree, consisting of a set of nodes $\mathbf{N}=\{N_1,\ldots,N_k\}$ and a set of directed edges $\mathbf{E}$ of the form $N \overset{\text{isa}}{\dashrightarrow} M$[3], where $N\in\mathbf{N}$ is the child and $M\in\mathbf{N}$ the parent of the edge. When drawing taxonomies, we usually omit the isa label. Intuitively, a directed edge $(N \overset{\text{isa}}{\dashrightarrow} M)\in E$ means that in the given taxonomy T , any object of type N also is a type M object. Hence, the objects classified as N also belong to (are of type) M. When viewed as sets of objects, N and M thus satisfy the containment relation $N\subset M$. Taxonomies are therefore also called containment or isa-hierarchies. In a taxonomy (tree) $T=(\mathbf{N},\mathbf{E})$, any $N\in\mathbf{N}$ (including every leaf node), has a unique path to the root N_0 of T. We can view a taxonomy as a special potential taxon graph (Berendsohn, 1995; Geoffroy and Güntsch, 2003) (also cf. (Beach et al., 1993)), i.e., one which forms a rooted tree and which only uses the containment relation $N\subset M$ between nodes.

2.2. *First-order logic (FOL)*

Here we present a brief introduction to some potentially unfamiliar first-order predicate logic (FOL) terminology and notation. Appendix A.1 presents this material in greater detail.

2.2.1. *Syntax*

The language $\mathcal{L}_{\text{FOL}}$ of first-order logic is built from an alphabet consisting of (i) a set of variables $V=\{x, y,z,...\}$, (ii) connectives $\neg,\wedge,\vee,\rightarrow,\leftrightarrow$ (not, and, or, if-then, if-and-only-if), (iii) quantifiers $\forall,\exists$ (for all and there exists), and (iv) a signature, $\mathbf{S}=\mathbf{R}\cup\mathbf{F}\cup\mathbf{C}$, involving sets of relation symbols $\mathbf{R}$, function symbols $\mathbf{F}$, and constants C.

A $\mathcal{L}_{\text{FOL}}$ formula is either an atomic formula $R(t_1,\ldots,t_k)$, where $t_1,\ldots,t_k$ are either variables or constants, or of the form $(\varphi\wedge\psi)$, $(\varphi\vee\psi)$, $(\varphi\rightarrow\psi)$, $(\varphi\leftrightarrow\psi)$, $(\neg\varphi)$, $\forall x{:}\varphi$, or $\exists x{:}\varphi$, where φ, ψ are $\mathcal{L}_{\text{FOL}}$ formulas, and x is a variable.

2.2.2. *Semantics*

An *interpretation* $\mathcal{I}$ maps the symbols in a signature $\mathbf{S}=\mathbf{R}\cup\mathbf{F}\cup\mathbf{C}$ to relations, functions, and objects in a modeled "real world." A formula φ without free variables is called a *constraint* and corresponds to a yes/no (boolean) query. We use Φ to represent a set of such constraints. If an *interpretation* makes true all constraints in Φ, we write $\mathcal{I}\models\Phi$, and say "$\mathcal{I}$ *satisfies* (or *is a model of*) Φ". If a formula φ is a logical consequence of a set of constraints[4] Φ, we write $\Phi\models\varphi$. See Example 5 in Section 3.1 for a concrete example of interpretations and constraints.

[2] For the importance of taxonomy in biology see, e.g.,(Wheeler, 2004, Franz et al., 2006).

[3] To avoid confusion with logical implication '→', we use '⇢' to depict graph edges in text mode.

[4] φ is a logical consequence of Φ if every model of Φ is also a model of φ.

2.2.3. *Automated deduction*

The semantic consequence relation $\Phi\models\varphi$ can be "mechanized" using the rules of a FOL *calculus*. A calculus is based on a *provability relation* $\Phi\vdash\varphi$, stating that φ can be derived (formally proven) from the formulas in Φ and the derivation rules of the calculus. Φ is called *consistent* if there is no formula φ such that both φ and its opposite $\neg\varphi$ can be derived from Φ; otherwise Φ is *inconsistent*. Appendix A.3 contains automatically generated proofs for some examples discussed in this paper.

3. Formalizing taxonomies as logic constraints

As described above, a taxonomy $T=(\mathbf{N},\mathbf{E})$ is a *rooted tree*, consisting of *nodes* $\mathbf{N}$ and *directed edges* $\mathbf{E}$. In Section 2.1 we indicated that an edge $M \overset{\text{isa}}{\dashrightarrow} N$ corresponds to a containment relation $N\subset M$.

3.1. *Formalizing isa-edges*

More formally, we can associate with every edge $N \overset{\text{isa}}{\dashrightarrow} M$ in $\mathbf{E}$, a first-order formula (or logic constraint) $\forall x{:}N(x)\rightarrow M(x)$, stating that if x is in N, then x is also in M. Since N and M are relation symbols with an *arity* of 1 (*i.e.*, they have a single argument), they are called *unary* relation symbols. An interpretation $\mathcal{I}$ satisfies this constraint iff $N^{\mathcal{I}}\subseteq M^{\mathcal{I}}$. With this logic formalization, the containment relation defined by $N \overset{\text{isa}}{\dashrightarrow} M$ is true if either $N^I\subsetneq M^I$ (proper containment) or $N^I=M^1$ (set equality).

In this way, we can associate with each taxonomy $T=(\mathbf{N},\mathbf{E})$ a set of FOL constraints Φ_T^{isa} which capture the meaning of the directed edges in $\mathbf{E}$:

$$\Phi_T^{isa} := \left\{\forall x : N(x)\rightarrow M(x) | N \overset{\text{isa}}{\dashrightarrow} M\in\mathbf{E},\ T=(\mathbf{N},\mathbf{E})\right\}$$

Once we have formalized a taxonomy T as a set of logical constraints Φ_T^{isa}, we can ask whether or not a given labeling $\mathcal{I}$ of elements (here: specimen) agrees with the taxonomy. Formally: does $\mathcal{I}\models\Phi_T^{\text{isa}}$ hold?

3.1.1. *Example 5*

Consider the following oversimplified taxonomy, created by a taxonomist named Carl:

$$T_{\text{Carl}} = \left(\underbrace{\{\text{Fox}, \text{Dog}, \text{Canis}\}}_{N}, \quad \underbrace{\left\{\text{Fox} \overset{\text{isa}}{\dashrightarrow} \text{Canis}, \text{Dog} \overset{\text{isa}}{\dashrightarrow} \text{Canis}\right\}}_{E} \right)$$

Carl's taxonomy T_{Carl}, interpreted as a set of FOL formulas $\Phi_{T_{Carl}}^{\text{isa}}$ constrains the interpretation of the names $\mathbf{N}=\{\text{Fox,Dog, Canis}\}$. Now consider a museum collection organized by a curator called Ed who has labeled the specimens in his collection with taxonomic names. We can view this labeling as a logic interpretation $\mathcal{I}_{\text{Ed}}$ that assigns to each node (taxon name) $N\in\mathbf{N}$ a subset $N^{\mathcal{I}_{\text{Ed}}}\subseteq D$ of elements from the underlying domain (here: specimens). Let Ed's specimen collection include $\{s_1, s_2\}$, and let his interpretation of Carl's names be $\text{Fox}^{\mathcal{I}_{\text{Ed}}}=\{s_1\}$, $\text{Dog}^{\mathcal{I}_{\text{Ed}}}=\{s_2\}$, $\text{Canis}^{\mathcal{I}_{\text{Ed}}}=\{s_2\}$, and $\text{Vulpes}^{\mathcal{I}_{\text{Ed}}}=\{s_1\}$.

We can now ask whether Ed's specimen labeling $\mathcal{I}_{\text{Ed}}$ satisfies the constraints imposed by Carl's taxonomy T_{Carl}, or more formally: does $\mathcal{I}_{Ed}\models\Phi_{T_{Carl}}^{\text{isa}}$ hold?

Fig. 3 – The 5 basic relations: (i) N≡M , (ii) N⊊M, (iii) N⊋M , (iv) N⊕M , and (v) N!M.

Given Carl's taxonomy and Ed's interpretation, the answer is no. Carl requires ∀x: Fox(x) → Canis(x), but for Ed's labeling $\mathcal{I}_{Ed}$ we have $\text{Fox}^{\mathcal{I}_{Ed}} \not\subseteq \text{Canis}^{\mathcal{I}_{Ed}}$. Therefore $\mathcal{I}_{Ed} \not\models \Phi^{\text{isa}}_{T_{Carl}}$, i.e., Ed's specimen labeling is *not* a model of (or: does not satisfy) Carl's taxonomy constraints.

3.2. *Model-checking a taxonomy vs. reasoning with taxonomy constraints*

Assume, as in Example 5, that an interpretation $\mathcal{I}$ is given and that we wish to check whether $\mathcal{I}$ is a model of (*i.e., satisfies*) Φ_T, where Φ_T captures constraints about the taxon names in *T*. Model-checking whether $\mathcal{I} \models \Phi_T$ holds can be done efficiently even for large interpretations (here: labeled specimen collections) $\mathcal{I}$, as it precisely corresponds to evaluating the constraints Φ_T against a given database instance $\mathcal{I}$. Thus, for each constraint $\varphi \in \Phi_T$ we can simply run a yes/no (*i.e.*, boolean) SQL query Q_φ against a relational database instance $\mathcal{I}$. Here, Φ_T is not limited to **isa** constraints, but can include *any* finite set of first-order constraints — a much more expressive language than Φ^{isa}_T.

While model-checking $\mathcal{I} \models \Phi_T$ does not require a full-fledged automated first-order reasoner (SQL query evaluation is sufficient), we will now encounter harder *implication problems*, where no interpretation $\mathcal{I}$ is given. Instead we wish to know whether a "piece of taxonomic knowledge" φ follows from a set of taxonomy constraints Φ_T, *independent* of $\mathcal{I}$. Determining this, *i.e.*, whether $\Phi_T \models \varphi$ holds, requires FOL reasoning.

3.3. $\mathcal{L}_{tax}$: *the language of first-order taxonomy constraints*

The set of constraints Φ^{isa}_T of a taxonomy *T* captures only the basic **isa**-hierarchy aspect of *T*. We can obtain a more powerful language of general *first-order taxonomy constraints* $\mathcal{L}_{tax}$ by allowing *any* FOL constraint over the signature **S=N**, *i.e.*, whose schema consists only of unary relations (denoting taxa). Since $\mathcal{L}_{\text{tax}}$ constraints contain only unary (*or monadic*) relation symbols, they correspond to a monadic first-order logic fragment (*cf.* Bachmair et al., 1993). It follows that reasoning with taxonomy constraints in $\mathcal{L}_{\text{tax}}$ is decidable, *i.e.*, there is a terminating algorithm which checks whether a given formula $\varphi \in \mathcal{L}_{\text{tax}}$ is a logical consequence of a finite set of first-order constraints $\Phi_T \subseteq \mathcal{L}_{\text{tax}}$ used to formalize *T*. Deciding the implication (reasoning) problem $\Phi_T \models \varphi$ is tantamount to filling possible "knowledge gaps" in *T*.

Corollary 1. The implication problem for taxonomies (formalized as constraints $\Phi_T \subseteq \mathcal{L}_{\text{tax}}$) is decidable.

Let $\Phi_T \subseteq \mathcal{L}_{\text{tax}}$ be a finite set of FOL constraints formalizing a taxonomy *T*. Then $\Phi_T \models \varphi$ is decidable. Since there are sound and complete first-order calculi (implemented by automated first-order reasoners), we can check whether φ can be *proven* from Φ_T using an automated reasoner, denoted $\Phi_T \vdash \varphi$. Similarly, the corollary implies that there is an algorithm to check whether a set of taxonomy constraints Φ_T is *consistent*. However, automated reasoning over large sets of taxonomy constraints may still be infeasible in practice. Fortunately, there is a large class of such constraints for which consistency checking can be done efficiently, i.e., in polynomial time (see Corollary 2 in Section 4.2.1).

4. Different types of taxonomy assumptions and constraints

A taxonomy *T* is usually seen as an **isa**-hierarchy of inclusion dependencies N⊂M, where the subconcept (or subclass) *N* is a child of the parent *M*. Whether $N \subset M$ means $N \subseteq M$ or rather $N \subsetneq M$ is often ambiguous. In addition, there are many other possible binary relations between pairs of sets (potential taxa) which we may want to assume, assert, or check in a given taxonomy. We describe these in the following. For any taxonomy *T*, we may assume none or some of these constraints. By formalizing *T* and its often latent assumptions as first-order constraints Φ_T, we can distinguish different types of taxonomies.

4.1. *Taxonomy graphs with basic relations between nodes*

Let *N*, *M* denote two non-empty sets.[5] Then exactly one of the following five basic relations must hold between them (*cf.* Fig. 3): (i) congruence ($N \equiv M$), (ii) proper inclusion ($N \subsetneq M$), (iii) proper inverse inclusion ($N \supsetneq M$), (iv) partial overlap ($N \oplus M$), and (v) exclusion (disjointness) ($N ! M$).

Let $\mathbb{B}_5 = \{\equiv, \subsetneq, \supsetneq, \oplus, !\}$ denote this set of basic relations. The importance of the $\mathbb{B}_5$ relations (and their combinations) for capturing and reasoning with constraints between (potential) taxa has been noted before, *e.g.*, see (Geoffroy and Berendsohn, 2003; Jonsson and Drakengren, 1997; Taxonomic Concept Schema, 2006; Franz et al., 2006).

Let $T = (\mathbf{N}, \mathbf{E})$ be a $\mathbb{B}_5$-*taxonomy graph* (not necessarily a rooted tree), *i.e.*, a labeled directed graph whose edges are labeled using only the five basic relations (instead of **isa** used above). Edges in such a taxonomy graph *T* are of the form

[5] Thus, the extents of *N* and *M* have been fixed as $N^{\mathcal{I}}$ and $M^{\mathcal{I}}$ via an (otherwise unimportant) interpretation $\mathcal{I}$.

$N \overset{\circ}{\dashrightarrow} M$, with $o \in \mathbb{B}_5$. We associate with T a set of constraints Φ_T^{B5} by formalizing each edge $N \overset{\circ}{\dashrightarrow} M \in \mathbf{E}$ using a FOL constraint:

- Congruence: for each $N \overset{\equiv}{\dashrightarrow} M$ in T, add the formula:

  $\forall x : N(x) \leftrightarrow M(x)$

- Proper inclusion: for each $N \overset{\subsetneq}{\dashrightarrow} M$ in T, add the formula:

  $\forall x : N(x) \rightarrow M(x) \wedge \exists a : M(a) \wedge \neg N(a)$

- Proper inverse inclusion: for each $N \overset{\supsetneq}{\dashrightarrow} M$ in T, add the formula:

  $\forall x : M(x) \rightarrow N(x) \wedge \exists a : N(a) \wedge \neg M(a)$

- Partial overlap: for each $N \overset{\oplus}{\dashrightarrow} M$ in T, add the formula:

  $\exists a \exists b \exists c : N(a) \wedge M(a) \wedge N(b) \wedge \neg M(b) \wedge \neg N(c) \wedge M(c)$

- Exclusion for each $N \overset{!}{\dashrightarrow} M$, add the formula:[6]

  $\neg \exists x : N(x) \wedge M(x)$

Unlike Φ_T^{isa} above, the constraints $\Phi_T^{\mathbb{B}_5}$ for a $\mathbb{B}_5$-taxonomy graph T may be inconsistent. For example, consider $T = \{B \overset{\equiv}{\dashrightarrow} A, C \overset{\subsetneq}{\dashrightarrow} A, B \overset{!}{\dashrightarrow} C\}$. The resulting set of constraints $\Phi_T^{\mathbb{B}_5}$ is inconsistent: Since $B \equiv A$ (first edge) it follows that $C \subsetneq B$ (using the second edge), contradicting $B \,!\, C$ (the third edge).

4.2. *Taxonomy graphs with combined relations between nodes*

We noted in the previous subsection that exactly one of the five basic relations $\mathbb{B}_5$ must hold between any two sets N^I and M^I, *i.e.*, provided an interpretation $\mathcal{I} = (D, I)$ is given for all taxa N, M under consideration. However, sometimes one does not have complete knowledge corresponding to an interpretation I of taxa. In this case, i.e. one does not have complete knowledge corresponding to an interpretation $\mathcal{I}$ of taxa. Instead, one often has to reason with *partial* taxonomic knowledge given in the form of constraints that are to be satisfied by *any* interpretation $\mathcal{I}$. For our constraint language (a suitable subset of $\mathcal{L}_{tax}$) this means that we should be able to express not only the basic relations $\mathbb{B}_5$, but the larger set of possible combinations between them. By disjunctively combining the basic relations $\mathbb{B}_5$ in all possible ways, we obtain 32 ($=2^5$) relations, one for each subset of $\mathbb{B}_5$, we denote this set of *combined relations* by $\mathbb{R}_{32}$. For example, $\mathbb{R}_{32}$ contains $\{\subsetneq, \equiv\}$ which is read as the disjunction $N \subsetneq M \vee N \equiv M$. This is equivalent to our original **isa** relation, *i.e.*, $N \subseteq M$. $\mathbb{R}_{32}$ also contains extreme cases of relations, *e.g.*, $\{\equiv, \subsetneq, \supsetneq, \oplus, !\}$, the disjunction of all basic relations, stating that nothing about the relationship between two taxa is known.

Similar as above, we define an $\mathbb{R}_{32}$-*taxonomy graph* T to be a labeled directed graph whose edges are of the form $N \overset{\circ}{\dashrightarrow} M$, but now with $o \in \mathbb{R}_{32}$. In the terminology of Berendsohn *et al.* this graph is also known as a *potential taxon graph* (Berendsohn, 2003). Thanks to our FOL formalization, reasoning with an $\mathbb{R}_{32}$-taxonomy graph T can now be reduced to traditional logic deduction with a set of constraints $\Phi_T^{\mathbb{R}_{32}}$. In particular, we can test whether some taxonomic knowledge φ is implied by T by checking whether $\Phi_T^{\mathbb{R}_{32}} \models \varphi$ holds. Similarly, we can test whether $\Phi_T^{\mathbb{R}_{32}}$ is consistent. Just as $\Phi_T^{\mathbb{B}_5}$ above, $\Phi_T^{\mathbb{R}_{32}}$ is a subset of $\mathcal{L}_{tax}$, so implication and satisfiability are decidable. An important remaining question is: How hard or easy is reasoning with $\Phi_T^{\mathbb{R}_{32}}$? In other words, is 'filling the gaps' in a potential taxon graph feasible for large sets of constraints $\Phi_T^{\mathbb{R}_{32}}$? We can answer this question by employing a theorem from Jonsson and Drakengren (Jonsson and Drakengren, 1997) derived in the context of the spatial algebra RCC-5.

4.2.1. *Computational complexity of reasoning with combined relations*

Consider the powerset (set of all subsets) $2^{\mathbb{B}_5}$ of $\mathbb{B}_5$. It contains 32 elements, corresponding to the 32 relations in $\mathbb{R}_{32}$. Loosely following notation and terminology in (Jonsson and Drakengren, 1997), an RCC-5 *formula* ψ is an expression of the form $N \{o_1,\ldots,o_n\} M$, where N, M are concept names or taxa,[7] and where $\{o_1,\ldots,o_n\}$ denotes a relation from $\mathbb{R}_{32}$ (*i.e.*, all o_i are basic relations from $\mathbb{B}_5$, for $1 \le i \le n$ and some $n \ge 0$). Equivalently, we can write ψ as an edge $N \overset{o_1,\ldots,o_n}{\dashrightarrow} M$ of an $\mathbb{R}_{32}$-taxonomy graph or as a disjunction of the form $(N\ o_1\ M) \vee \ldots \vee (N\ o_n\ M)$, emphasizing that ψ is equivalent to a disjunctive formula in $\mathcal{L}_{tax}$ (and thus in FOL).

Let $\Psi^{\mathbb{R}_{32}}$ be a finite set of such formulas φ. In our terminology, $\Psi^{\mathbb{R}_{32}}$ corresponds to an $\mathbb{R}_{32}$-taxonomy graph T and hence to a set of constraints $\Phi_T^{\mathbb{R}_{32}}$. We are interested in the following questions:

- Is $\Psi^{\mathbb{R}_{32}}$ satisfiable, *i.e.*, is there a model $\mathcal{I} \models \Psi^{\mathbb{R}_{32}}$? (1)
- Does a given $\Psi^{\mathbb{R}_{32}}$ *imply a given* ψ, *i.e.*, does $\Psi^{\mathbb{R}_{32}} \models \psi$ hold? (2)

Question (1) allows us to check whether a taxonomy T represented as $\Phi_T^{\mathbb{R}_{32}}$ (or equivalently $\Psi^{\mathbb{R}_{32}}$) is consistent, while (2) corresponds to deciding whether some information ψ follows from the given constraints (thus possibly 'filling a knowledge gap' in a potential taxon graph T). Since (1) and (2) correspond to implication problems in (a subset of) $\mathcal{L}_{tax}$, it follows from Corollary 1 that (1) and (2) are decidable.

The following theorem by Jonsson and Drakengren (Jonsson and Drakengren, 1997) gives a precise account of the complexity of deciding these questions by identifying all maximal tractable subalgebras of $\mathbb{R}_{32}$.

Theorem 1. *(Jonsson and Drakengren, 1997) Let $\mathbb{R} \subseteq \mathbb{B}_5$ be any of the 32 subsets of $\mathbb{B}_5$. Then deciding whether $\Psi^{\mathbb{R}}$ is satisfiable is polynomial iff $\mathbb{R}$ is a subset of one of $\mathbb{R}_5^{28}$, $\mathbb{R}_5^{20}$, $\mathbb{R}_5^{17}$, or $\mathbb{R}_5^{14}$, and NP-complete otherwise.*

Thus, in general we can expect reasoning with combined relations from $\mathbb{R}_{32}$ to be efficient for large sets of constraints only if we consider combined relationships that correspond to one of the four maximal classes $\mathbb{R}_5^{28}$, $\mathbb{R}_5^{20}$, $\mathbb{R}_5^{17}$, or $\mathbb{R}_5^{14}$ (or subsets of those); for other constraint sets over $\mathbb{R}_{32}$, not falling under those four, reasoning is NP-complete and thus infeasible in general.

[6] Two equivalent, alternative formalizations are: (i) $\forall x$: $\neg N(x) \vee \neg M(x)$ and (ii) $\forall x$: $N(x) \rightarrow \neg M(x)$.

[7] Since RCC-5 is derived from a spatial algebra RCC-8, N and M are called region variables in (Jonsson and Drakengren, 1997).

For our purposes, the subalgebra $\mathbb{R}_5^{28}$ (see Appendix A.2) is the most interesting: it contains 28 out of the 32 possible relations in $\mathbb{R}_{32}$. The only excluded relations are $\{\subsetneq, \supsetneq\}$, $\{\subsetneq, \supsetneq, \equiv\}$, $\{\subsetneq, \supsetneq, !\}$, and $\{\subsetneq, \supsetneq, \equiv, !\}$. Note that all four of these contain $\{\subsetneq, \supsetneq\}$, *i.e.*, the disjunction $(N \subsetneq M) \vee (N \supsetneq M)$. In practical settings it does not appear to be common that one is uncertain whether $(N \subsetneq M)$ or rather $(N \supsetneq M)$. For example, if the taxonomic ranks of N and M are known, then one of these cases can always be eliminated.[8]

Using the results from Theorem 1, we have established for which subsets $\mathbb{R}$ of $\mathbb{R}_{32}$ deciding question (1) is tractable. Moreover, question (2) can be reduced to (1) for subalgebras $\mathbb{R}$ which are closed under negation:

To check $\Psi^{\mathbb{R}} \models \psi$ simply check whether $\Psi^{\mathbb{R}} \cup \{\neg\psi\}$ is satisfiable. Summarizing, we obtain:

Corollary 2. Reasoning with arbitrary $\mathcal{L}_{\text{tax}}$ taxonomy constraints can be infeasible (NP-hard) in general. Reasoning with $\mathbb{R}_5^{28}$ taxonomy constraints is efficient (i.e., deciding satisfiability is in polynomial time).

4.3. *Capturing latent taxonomic assumptions*

So far we have considered fragments of our taxonomy language $\mathcal{L}_{\text{tax}}$ for capturing simple isa-hierarchies (Section 3.1), the five basic relations $\mathbb{B}_5$ (Section 4.1), and the 32 combined relations $\mathbb{R}_{32}$ (Section 4.2). When reasoning about taxonomies in logic, latent taxonomic assumptions (LTAs) can often play an important role. We consider such LTAs in the following.

4.3.1. *LTA: non-emptiness*

A given taxon $N \in \mathbf{N}$ of a taxonomy $T = (\mathbf{N}, \mathbf{E})$ may or may not have actual instances. If we want to express that some nodes $\mathbf{N}_\exists \subseteq \mathbf{N}$ cannot be empty, *i.e.*, have at least one instance, then we can express this using a logic constraint: for each $N \in \mathbf{N}_\exists$, add the constraint:

- $\exists x\colon N(x)$

The results in Section 4.2 for RCC-5 are based on the assumption that all nodes are non-empty, *i.e.*, $\mathbf{N}_{\exists=}\mathbf{N}$.[9]

4.3.2. *LTA: sibling disjointness*

Most biological taxonomies require disjointness of *sibling* elements, *i.e.*, among the children $\{N_1, N_2, \ldots\}$ of a parent M. For example, if a wolf is a kind of Canis, and a fox is a kind of Canis, then an animal cannot be both a wolf and a fox. The formulas enforcing sibling disjointness are: for each pair of edges $N_1 \overset{\text{isa}}{\dashrightarrow} M$ and $N_2 \overset{\text{isa}}{\dashrightarrow} M$ with $N_1 \neq N_2$, add the constraint:

- $\neg\exists x\colon N_1(x) \wedge N_2(x)$

4.3.3. *LTA: coverage*

Let M be a node whose children are $\{N_1, \ldots, N_\ell\}$. We know at least that $N_i \overset{\text{isa}}{\dashrightarrow} M$ for all $i = 1, \ldots, \ell$ (and maybe even $N_i \overset{\subsetneq}{\dashrightarrow} M$ for some or all i). So the union of all children is contained in the parent M. However, we may or may not know whether the converse is true, *i.e.*, that the union of children *covers* the parent. To enforce this, let $\mathbf{M}_\forall \subseteq \mathbf{N}$ be the set of parents who are completely covered by their children (*i.e.*, there are no "surprise" children): for all $M \in \mathbf{M}_\forall$ having children $N_1, \ldots, N_\ell$ of M, add:

- $\forall x : M(x) \rightarrow N_1(x) \vee N_2(x) \vee \ldots \vee N_\ell(x)$

4.3.4. *Further constraints*

The above LTAs are just a few of the many possible constraints that may be assumed for or required of a particular taxonomy. Completeness and rank are two additional constraints that can have an impact on the types of reasoning and inferences that a taxonomy can support:

A taxonomy $T = (\mathbf{N}, \mathbf{E})$ is a *complete tree* (short: *complete*) if all leaf nodes have the same distance from the root. Some taxonomies, such as strictly traditional seven-leveled biological taxonomies are complete. Others, such as phylogenetic trees are not.

A taxonomy may also be ranked.[10] If $T = (\mathbf{N}, \mathbf{E})$ is ranked then there is a function $\varrho\colon \mathbf{N} \rightarrow \mathbf{R}$, where $\mathbf{R}$ is a set of ranks (*e.g.*, $R = \{\text{species, genus, family, order, class, phylum, kingdom}\}$ is common). These ranks are ordered and every edge $N \overset{\text{isa}}{\dashrightarrow} M$ must maintain that order, *i.e.*, $\varrho(N) < \varrho(M)$.

4.4. *Dealing with different taxonomy types*

A given type of taxonomy may include some or all of the constraints described above. Different types of taxonomy are relevant for different circumstances:

For example, in a published expert taxonomy, each taxon (e.g., Ranunculus according to Benson, 1948) has at least one instance (e.g., a species holotype). Each taxon in such a taxonomy has a rank, and the child taxa of a given taxon are disjoint. Depending on the taxa considered, the taxonomy may or may not be complete; some species have subspecies or varieties, while others do not. Finally, for any given taxon, the expert usually has tried to capture the full extent of the taxon (perhaps within a geographic scope); the set of a taxon's elements is defined as the combined elements of that taxon's child taxa. Taken together, we would say that a published taxonomy conforms to the following constraints or LTAs: *non-emptiness, sibling disjointness, coverage, and ranks*.

As another example, consider a museum's specimen collection. In this situation, there may be specimens that are not identified down to the species level. If this is the case, a specimen might be an element of a genus-level node, but not an element of any child species-level node (Blum, 2007). Therefore, a genus may contain specimens not contained in any of the species under that genus. We would say that the taxonomy does not adhere to the coverage constraint. The taxonomy is otherwise similar to the one described above: all taxa in the collection have at least one example (*non-emptiness*), child taxa are disjoint, and the taxa are *ranked*.

[8] On the other hand, the excluded cases could potentially arise as (intermediate) results of a larger reasoning problem.

[9] Some investigations of mappings between ontologies have shown that populating ontologies with instances may be useful when relating concepts in those ontologies (Kalfoglou and Schorlemmer, 2002, Kent, 2000).

[10] The term "rank" is heavily overloaded: In graph theory, the rank of a node is the number of children of that node. In set theory, the rank of a set is the number of elements in the set. Here, a rank is a label attached to the height of a taxonomic tree. All nodes at the same height are considered to be of the same rank.

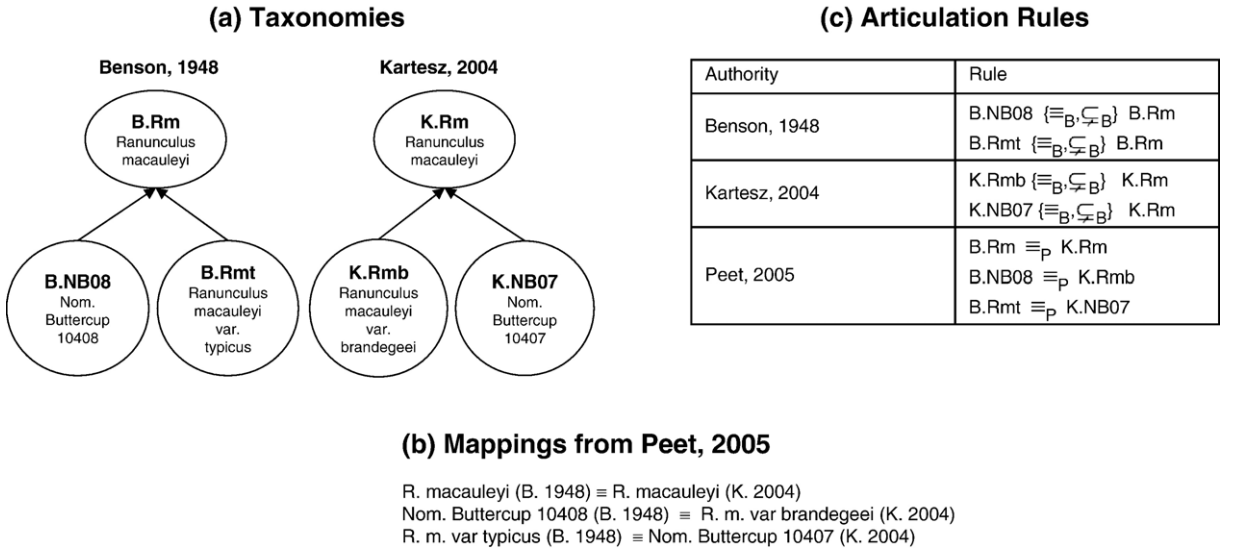


Fig. 4 – Equivalence between two taxonomies (a), given a mapping (b), and articulations (c).

Thus, when formalizing a taxonomy, it is important to know what type of taxonomy is intended. For example, a taxonomist who wishes to produce a traditional taxonomic revision of a given genus would provide a *ranked, sibling-disjoint, covered* taxonomy, requiring *non-emptiness* of nodes. In addition to using the constraints to ensure that an instance of a taxonomy conforms to the rules of the specified type of taxonomy, the constraints provide additional formulae which may contribute extra deductive power when deciding whether or not a mapping between two taxonomies is logically consistent. The additional formulas may also help a logic-based reasoner deduce additional mappings between taxonomies. Before demonstrating this, we must first define how one can express mappings between different taxonomies as constraints.

Table 3 – $\mathcal{L}_{tax}$ rules for Fig. 4 plus non-emptiness, sibling-disjointness and coverage constraints

Authority	Rule
Benson, 1948	$(\forall x\colon B.NB08(x) \leftrightarrow B.Rm(x)) \vee$ $(\forall x\colon B.NB08(x) \rightarrow B.Rm(x) \wedge \exists a\ B.Rm\ (a)$ $\wedge \neg B.N\ B08(a))$ $(\forall x\colon B.Rmt(x) \leftrightarrow B.Rm(x)) \vee$ $(\forall x\colon B.Rmt(x) \rightarrow B.Rm(x) \wedge \exists a\ B.Rm\ (a)$ $\wedge \neg B.Rmt\ (a))$
Kartesz, 2004	$(\forall x\colon K.Rmb \leftrightarrow K.Rm(x)) \vee$ $(\forall x\colon K.Rmb(x) \rightarrow K.Rm(x) \wedge \exists a\ K.Rm(a)$ $\wedge \neg K.Rmb(a))$ $(\forall x\colon K.NB07(x) \leftrightarrow K.Rm(x)) \vee$ $(\forall x\colon K.NB07(x) \rightarrow K.Rm(x) \wedge \exists a\ K.Rm(a)$ $\wedge \neg K.N\ BO7(a))$
Peet, 2005	$\forall x\colon B.Rm(x) \leftrightarrow K.Rm(x)$
	$\forall x\colon B.NB08(x) \leftrightarrow K.Rmb(x)$
	$\forall x\colon B.Rmt(x) \leftrightarrow K.NB07(x)$
LTA: Sibling disjointness	$\forall x\colon B.NB08(x) \rightarrow \neg B.Rmt(x)$
	$\forall x\colon K.Rmb(x) \rightarrow \neg K.NB07(x)$
LTA: coverage	$\forall x\colon B.Rm(x) \leftrightarrow B.NB08(x) \vee B.Rmt(x)$
	$\forall x\colon K.Rm(x) \leftrightarrow K.Rmb(x) \vee K.NB07(x)$
LTA: non-emptiness	$\exists x\colon B.Rm(x)$
	$\exists x\colon B.NB08(x)$
	$\exists x\colon B.Rmt(x)$
	$\exists x\colon K.Rm(x)$
	$\exists x\colon K.Rmb(x)$
	$\exists x\colon K.NB07(x)$

5. Formalizing mappings between taxonomies as articulations

In the previous sections, we have mainly considered individual taxonomies, but not relationships between two or more *different* taxonomies. Borrowing terminology from the area of knowledge representation and formal ontologies, we call a logic constraint relating names or concepts from different taxonomies an articulation (Mitra et al., 2000). Informally, we may also speak of a *mapping* between (concepts from) different taxonomies. Our formal framework, i.e., the language $\mathcal{L}_{tax}$ and the important RCC-5 fragment $\mathbb{R}_5^{28}$, can be easily extended to represent and *reason* with articulations across taxonomies. Automated reasoning support for inter-taxonomy constraints (articulations) is even more important than being able to reason with intra-taxonomy constraints, since it is very easy to make subtle mistakes when mapping between taxonomies.

For example, let T_1 and T_2 be taxonomies, and α an articulation between them, denoted $T_1 \sim_\alpha T_2$. We can represent all three artifacts as logic constraints and check, using an automated reasoner, whether a specific formula φ is a consequence, *i.e.*, whether $\Phi_{T1} \cup \Phi_{T2} \cup \Phi_\alpha \models \varphi$ holds. Some consequences φ may be unintended, *e.g.*, if φ=False can be deduced,[11] then $\Phi_{T1} \cup \Phi_{T2} \cup \Phi_\alpha$ is inconsistent, indicating that the taxonomies and/or the articulation α "have problems".

It is easy to see how to extend the formal $\mathcal{L}_{tax}$ and RCC-5 reasoning framework to include articulations: First, we assume there are no name clashes between names N_1 and N_2 from any two different taxonomies T_1 and T_2. This can be achieved, *e.g.*, by qualifying each node N with the *authority* for

[11] Instead of False, sometimes the empty clause □ is used.

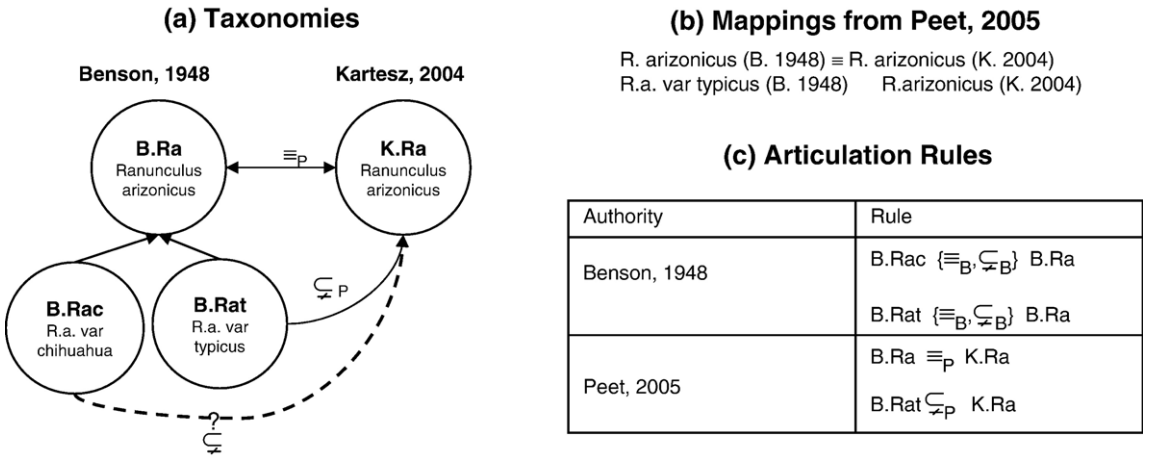


Fig. 5 – Inference of new articulations from taxonomies (a), a mapping (b), and articulations (c).

the taxonomy from which N is taken. For example, in Section 1 we distinguished between N^{K04} and N^{B48}, depending whether with N we meant the name or concept according to Kartesz, 2004 or according to Benson, 1948.

In the following, let A be an authority for taxonomy T, and N a name from T. Then we can write either N^A (as in Section 1) or $A.N$ (as in the following) to indicate that name N is used in the sense of A.

5.1. *A simple articulation example*

Throughout the remainder of this paper, we apply $\mathcal{L}_{tax}$ to taxonomies and articulations taken from a real-world dataset. The dataset contains nine different expert taxonomies of the genus *Ranunculus* — a flowering plant genus which includes buttercups. In addition to the seven taxonomies, an expert, Peet (Peet, 2005) has created inter-taxonomy mappings for a large number of the taxa. These mappings use the inter-taxon relationships taken from the TCS standard (Taxonomic Concept Schema, 2006) for representing taxonomic information. We have translated a subset of those relationships into the basic RCC-5 relations $\mathbb{B}_5$.

Fig. 4 shows the equivalence between two taxonomies for the species *Ranunculus macauleyi* and its varieties. Part (a) of the figure shows a taxonomy published by Benson in 1948 (Benson, 1948), and a newer one, published by Kartesz in 2004 (Kartesz, 2004). Part (b) lists the mappings between the taxa, provided by Peet. Part (c) combines the taxonomies and the mappings into a set of articulations. These articulations, combined with the sibling-disjointness, coverage, and non-emptiness constraints are then transformed into $\mathcal{L}_{tax}$ formulas, shown in Table 3.

To prove the consistency of the taxonomies and mapping in Fig. 4, we can apply an automatic reasoner such as MACE4 to the $\mathcal{L}_{tax}$ rules in Table 3. MACE4[12] discovers models that satisfy sets of first-order logic formulas. If MACE4 finds such a model, the formulas must be consistent. In the case of Fig. 4, Mace4 finds a model with two domain elements, a and b: a satisfies $B.NB08(x)$, $B.Rm(x)$, $K.Rmb(x)$, and $K.Rm(x)$, while b satisfies $B.Rmt(x)$, $B.Rm(x)$, $K.NB07(x)$ and $K.Rm(x)$. By discovering a model for the formulas, MACE4 proves the consistency of the taxonomies and the associated articulations.

[12] PROVER9 and MACE4: http://www.cs.unm.edu/~mccune/mace4/.

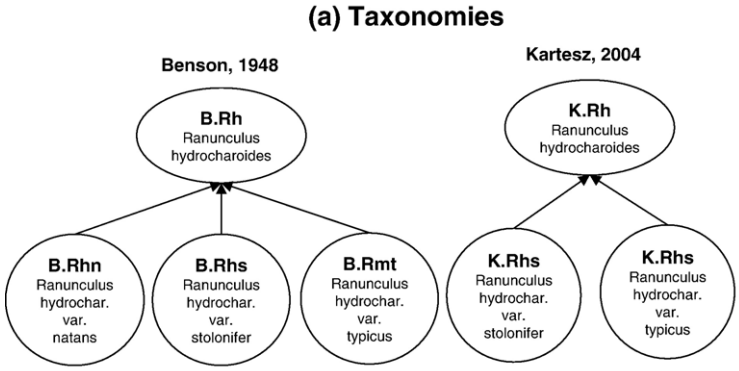
5.2. *Discovering unstated articulations*

Leveraging the machinery of first-order logic, we can use $\mathcal{L}_{tax}$ to discover unstated, but logically implied consequences φ (including new articulations) between taxa.

Peet makes the following claims (Fig. 5): Benson's *R. arizonicus var. typicus* (marked B.Rat) is included in Kartesz's

Table 4 – Prover9's proof of the query in Fig. 5: is Benson's *Ranunculus arizonicus* var. Chihuahua contained in Kartesz's *Ranunculus arizonicus*?

Clause#	Formula or clause	Comment
1	$\forall x : B.Rac(x) \rightarrow B.Ra(x)$	Assumption
3	$\forall x : B.Ra(x) \leftrightarrow K.Ra(x)$	Assumption
4	$\forall x : B.Rat(x) \rightarrow K.Ra(x)$	Assumption
8	$\exists x : B.Rat(x)$	Assumption
10	$\forall x : B.Rac(x) \rightarrow \neg B.Rat(x)$	Assumption
12	$\forall x : B.Rac(x) \rightarrow K.Ra(x) \wedge (\exists y : (K.Ra(y) \wedge \neg B.Rac(y)))$	Goal
14	$\forall x : \neg B.Rac(x) \vee B.Ra(x)$	Clausify 1
15	$\forall x : \neg B.Rac(x) \vee \neg B.Rat(x)$	Clausify 10
17	$B.Rac(c6)$	Deny 12
18	$\forall x : \neg K.Ra(c6) \vee \neg K.Ra(x) \vee \neg B.Rac(x)$	Deny 12
19	$B.Rat(c4)$	Clausify 8
21	$\forall x : \neg B.Rat(x) \vee K.Ra(x)$	Clausify 4
25	$\forall x : \neg K.Ra(c6) \vee \neg K.Ra(x) \vee \neg B.Rat(x)$	Resolve 18 15
27	$\forall x : \neg B.Ra(x) \vee K.Ra(x)$	Clausify 3
30	$B.Ra(c6)$	Resolve 17 14
35	$K.Ra(c4)$	Resolve 19 21
36	$\neg K.Ra(c6) \vee \neg K.Ra(c4)$	Resolve 25 19
37	$\neg K.Ra(c6)$	Copy 36, unit_del 35
40	$K.Ra(c6)$	Resolve 30 27
41	$\square$	Copy 40, unit_del 37



(b) Mappings from Peet, 2005

R. hydrocharoides (B. 1948) ≡ R. hydrocharoides (K. 2004)
R.h.v. stolonifer (B. 1948) ≡ R.h.v. stolonifer (K. 2004)
R.h.v. typicus (B. 1948) ≡ R.h.v. typicus (K. 2004)

(c) Articulation Rules

Authority	Rule
Benson, 1984	B.Rhn $\{\equiv_B, \subsetneq_B\}$ B.Rh B.Rhs $\{\equiv_B, \subsetneq_B\}$ B.Rh B.Rht $\{\equiv_B, \subsetneq_B\}$ B.Rh
Kartesz, 2004	K.Rhs $\{\equiv_B, \subsetneq_B\}$ K.Rh K.Rht $\{\equiv_B, \subsetneq_B\}$ K.Rh
Peet, 2005	B.Rh $\equiv_P$ K.Rh B.Rhs $\equiv_P$ K.Rhs B.Rht $\equiv_P$ K.Rht

Fig. 6 – A mapping/articulation whose inconsistency is not obvious.

R. arizonicus (*K.Ra*), and Bensons's *R. arizonicus* (*B.Ra*) is equivalent to Kartesz's *R. arizonicus* (*K.Ra*). From these facts, it seems plausible that Benson's *Ranunculus arizonus var. chihuahua* (*B.Rac*) must also be contained in Kartesz's *R. arizonicus*. We use our formalization to determine if this is the case.

Intuitively, it is easy to see that *B. Rac* must be contained in *K. Ra*. *B. Rac* is a type of *B. Ra* and Peet says that *B. Ra* equals *K. Ra*. Because those terms are equivalent, *K. Ra* may be substituted wherever *B. Ra* is seen, and therefore, *B. Rac* $\subseteq_P$ *K. Ra*. To formally prove the hypothesis, we use PROVER9 on the first-order formulas created by translating the articulations in Fig. 5 into $\mathcal{L}_{tax}$ along with the non-emptiness, disjoint-children, and coverage constraints. We then ask PROVER9 to prove that *B. Rac* is a proper subset of *K. Ra*. The proof attempt succeeds with the proof shown in Table 4: Prover9 shows that *B. Rac* $\subsetneq$ *K. Ra* by proving that the negation of the hypothesis leads to a contradiction. Although the hypothesis seems intuitively true, the automatically derived formal proof involves many mechanical proof steps.[13]

5.3. *Articulation consistency*

Our formalization can also be used to determine the consistency of a given set of taxonomies and mappings. Consider the mapping from the Ranunculus data set shown in Fig. 6. The taxonomies and articulation derived from this figure are inconsistent[14] if the taxonomies conform to the non-emptiness, sibling-disjointness and coverage constraints. We can see the inconsistency of this articulation by asserting a member of *Ranunculus hydrocharoides var. natans* (Benson, 1948). If Benson's taxonomy adheres to the non-emptiness constraint, then there must be such an element. If there is such an element, then it is also an instance of *R. hydrocharoides var. natans* (Benson, 1948). Following Peet's mapping, the instance must also be an instance of *R. hydrocharoides var. natans* (Kartesz, 2004). According to the coverage constraint, this means that it must also be an instance of either *Ranunculus hydrocharoides var. stolonifer* (Kartesz, 2004) or *Ranunculus hydrocharoides var. typicus* (Kartesz, 2004). However, if this is the case, then the instance of *R. hydrocharoides var. natans* (Benson, 1948) must also be an instance of either *Ranunculus hydrocharoides var. stolonifer* (Benson, 1948) or *Ranunculus hydrocharoides var. typicus* (Benson, 1948). Both cases violate the disjoint-children constraint.

This example is interesting because the inconsistency only appears if the taxonomies adhere to the non-emptiness, sibling-disjointness and coverage constraints. Had any of these constraints not been in effect, the articulation would have been considered consistent. If the sibling-disjointness constraint was not in effect, for example, all the elements of B.Rhn could also be elements of B.Rhs and there would be no inconsistency. If the coverage constraint was not in effect, K.Rh could have elements not contained in K.Rhs or K.Rht, and these elements could be the same as those in B.Rhn. Finally, if the non-emptiness constraint was not in place, then B.Rhn could simply have been empty. The dependence of the proof of inconsistency on the application of all three constraints highlights the importance of clearly stating the type of taxonomies being mapped; the relationships between the nodes in the taxonomies are not enough to fully describe a mapping.

We formally show the inconsistency of the elements in Fig. 6 using PROVER9. To show that $\Phi \models \varphi$ the system adds the negated goal $\neg\varphi$ to the assumptions Φ, trying to find a refutation, i.e., showing that $\Phi \cup \{\neg\varphi\} \vdash \square$ holds.[15] The proof establishing the inconsistency of the articulation in Fig. 6 appears in Appendix A.3.2.

In this example, the explanation for Peet's mapping derives from the geographic scope of the two authorities being mapped. Benson, 1948 provides a world-wide taxonomy for *R. hydrocharoides* while Kartesz's taxonomy is restricted to North America, where there are no known instances of *R. hydrocharoides var. natans*. In one sense, limiting the scope

[13] See (Thau and Ludascher,submitted for publication) for first results on optimizing our framework for thousands of such proof jobs.

[14] More precisely, $\Phi(T_1) \cup \Phi(T_2) \cup \Phi(A) \models \square$.

[15] If Prover9 is simply given a set of formulas, a proof means finding a refutation of the given set of clauses (any one of which could be thought of as the negated theorem to be proven).

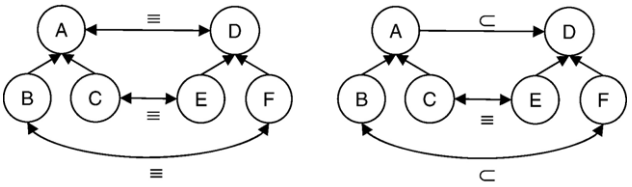


Fig. 7 – Alternative articulations between two taxonomies.

to North America, Benson and Kartesz have equivalent definitions of *R. hydrocharoides*, hence Peet's equivalence mapping. In another sense, when applied globally, Kartesz's notion of *R. hydrocharoides* should be represented as a subset of Benson's *R. hydrocharoides*. This indicates that data classified with Kartesz's North American taxonomy may be correctly merged into data classified according to Benson's world taxonomy, but not the other way around. Merging specimens classified with Benson's world taxonomy into a data set classified with Kartesz's taxonomy, and stating that the combined data are in accordance with Kartesz's taxonomy would lead one to incorrectly believe that Kartesz's taxonomy was global in scope rather than local.

6. Conclusions and future work

This paper has provided a framework for a first-order predicate logic representation of taxonomies and articulations between them. We have discussed several features of taxonomies and shown how to represent these features as logical constraints. These logical constraints may be used to determine whether or not a given taxonomy conforms to a required type of taxonomy. They may also be used to test the consistency of two taxonomies together with articulations between them. We have also shown how different types of taxonomy can provide different degrees of deductive power (intuitively, the more LTAs we can consistently assume, the more articulations we can infer). Finally, the implication problem in our FOL fragment $\mathcal{L}_{\text{tax}}$ is decidable, and by applying results on the RCC-5 algebra, we have shown that in some cases reasoning may be performed in polynomial time. This work can be extended in a number of ways.

6.1. *Combining articulations*

This paper has focussed on cases in which only one authority provides a given set of articulations. However, more than one authority may describe sets of articulations between a given pair of taxonomies. In this case, it may be possible to merge the articulations. In the simplest situation, the two sets of articulations will be identical. In a more complex situation, the articulations may disagree with each other. For example, Fig. 7 shows two alternative consistent articulations between two taxonomies. The only difference between the sets of formulas describing these two scenarios is that the first scenario has these additional formulas $D \rightarrow A$, $F \rightarrow B$. The overlap between the formulas in these examples leads to a type of ordering of the sets of articulations, where $A_2 \prec A_1$. Given an ordering like this, we could conclude that any data integration or discovery done using articulation A_2 should be considered valid by someone who believes that articulation A_1 is correct. However, the ordering means that the reverse is not true. Someone who prefers A_2 cannot completely trust any data aggregation performed by a person using articulation A_1 because that person may have cast data labeled by node *F* into data labeled by node *B*.

(Geoffroy and Berendsohn, 2003) present two ways to combine two sets of taxonomic articulations: intersection and union. The intersection of two sets of articulations is created by taking the conjunction of the articulations between each pair of nodes. This will result in an "≡" relation between the nodes *C* and *E* and an "unknown" relation between nodes *A* and *D* above. It may be that certain "unknowns" might be removed by reasoning over the resulting formulas. This is another place to utilize a logic-based formalization.

The union of two sets of articulations is created by taking the disjunction of the articulations between each pair of vertices. For example, the union of A_1 and A_2 above leads to nodes *A* and *D* having an {≡, ⊂} articulation; which should be read as "equivalent or subset". As discussed in Section 4, there are 32 possible combined relationships. According to (Jonsson and Drakengren, 1997) certain subsets of those 32 combined relationships lead to tractable reasoning. Future work will investigate the applicability of the RCC-5 algebra to reasoning about combined relationships.

6.2. *Coping with many alternative opinions*

There may be many possible sets of articulations available to aid in combining data annotated with taxonomic concepts. Determining which articulations should be used may be difficult for someone without taxonomic expertise in the relevant taxa. In this case, a logic-based formalization of the articulations can help a user choose which articulations to invoke when integrating data by elucidating the similarities and differences between the articulations. Given a set of articulations, a data integrator has many options, e.g.:

- Choose a "favorite" articulator.
- Choose some articulations thought to be true, and choose articulators who agree with those.
- Find articulations for taxa of interest which are agreed upon by the largest group of articulators.
- Choose only articulations which maximize the amount of data which can be integrated.
- Choose only articulations which add no uncertainty.

For all but the first option, our formalization should be able to help an integrator decide which articulations to use. The situation is similar to the more common case where taxonomic data are not annotated with taxonomic concepts and are instead simply given names, such as *Ranunculus glabberimus*. This creates a scenario similar to that above, but with additional uncertainty in deciding which taxonomies to choose, as well as which articulations between the taxonomies. Again, our formalization can provide data integrators with information to make a more informed decision about which taxonomies and articulations to follow.

6.3. *Richly defined concepts*

The nodes in the current model are defined by their relations to each other. This is in keeping with many of the descriptions of taxa available online[16] and with the articulations available (Franz et al., 2006; Koperski et al., 2000; Weakley, 2006). In general, however, taxa, or other concepts in a taxonomy, may be defined more richly. A taxon, for example, can be defined by a set of characters describing the organisms circumscribed by the taxon. In addition, a taxon may be defined using a set of type specimens. We have ignored these, and other complications such as taxonomic rank, in the current analysis. We have shown that reasoning in our language $\mathcal{L}_{\text{tax}}$ is decidable, and we have identified when reasoning may be infeasible (at least NP-hard), and when efficient (polynomial time) decisions procedures exist. $\mathcal{L}_{\text{tax}}$ is currently limited to unary predicates. Extending the formalization to handle richly defined concepts like those defined above, and involving taxonomic rank, may involve extending the formalization to binary or *n*-ary predicates. Specialized reasoning procedures to effectively and efficiently handle larger classes is an interesting topic for future research.

6.4. *Conclusion*

As the examples in this paper have demonstrated, our first-order logic-based formalization provides a well-founded mechanism for reasoning about taxonomies and articulations between them. The machinery of first-order logic has been extensively studied, and much of what has been learned can be applied directly to better understanding how taxonomies from different authorities may be compared and reasoned over. Understanding the relationships between different taxonomies is a crucial first step toward automatic merging of ecological data sets which have been indexed with taxonomic concepts or names.

Appendix A

A.1. *First-order logic (FOL)*

For ease of reference, we summarize basic notions from first-order predicate logic (Ebbinghaus et al., 1994).

A.1.1. *Syntax*

The language $\mathcal{L}_{\text{FOL}}$ of first-order logic is built from an alphabet consisting of (i) a set of *variables* $\mathbf{V}=\{x, y, z,...\}$, (ii) *connectives* $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ (*not, and, or, if-then, if-and-only-if*), (iii) *quantifiers* $\forall$, $\exists$ (for all and there exists), and (iv) a signature $\mathbf{S}=\mathbf{R}\cup\mathbf{F}\cup\mathbf{C}$, involving sets of relation symbols $\mathbf{R}$ (R_1, R_2,...), function symbols $\mathbf{F}$ (f_1, f_2,...), and constants $\mathbf{C}$ (c_1, c_2,...). Each $R\in\mathbf{R}$ and $f\in\mathbf{F}$ has a unique *arity*≥ 1.

The set $\mathbf{T}$ of terms is the least set such that (i) $\mathbf{V}$, $\mathbf{C}\subseteq\mathbf{T}$ (constants and variables are terms), and (ii) for every *k-ary* $f\in\mathbf{F}$ and $t_1,..., t_k\in\mathbf{T}$, also $f(t_1,..., t_k)\in\mathbf{T}$.

A $\mathcal{L}_{\text{FOL}}$ *formula* is either an *atomic formula* $R(t_1,..., t_k)$, with *k-ary* $R\in\mathbf{R}$ and $t_1,..., t_k\in\mathbf{T}$, or of the form $(\varphi\wedge\psi)$, $(\varphi\vee\psi)$, $(\varphi\rightarrow\psi)$, $(\varphi\leftrightarrow\psi)$, $(\neg\varphi)$, $\forall x(\varphi)$, or $\exists x(\varphi)$, where φ, ψ are $\mathcal{L}_{\text{FOL}}$ formulas, and $x\in\mathbf{V}$. Parentheses may be omitted when clear from the context.

A.1.2. *Semantics*

Fix a signature $\mathbf{S}=\mathbf{R}\cup\mathbf{F}\cup\mathbf{C}$. A first-order structure $\mathcal{I}=(D, I)$ for $\mathbf{S}$ consists of a domain D and a mapping I, assigning to every constant $c\in\mathbf{C}$, *k-ary* function symbol $f\in\mathbf{F}$, and *k-ary* relation symbol $R\in\mathbf{R}$, a domain element c^I, a *k-ary* function $f^I: D^k\rightarrow D$, and a *k-ary* relation $R^I\subseteq D^k$, respectively. Since I interprets (*i.e.*, assigns meaning to) all symbols in S, terms and formulas over S can be evaluated under $\mathcal{I}$, provided we also map free variables to domain elements via a variable assignment $\beta: \mathbf{V}\rightarrow D$. Let $\mathfrak{I}=(\mathcal{I}, \beta)$ be an interpretation, i.e., a first-order structure $\mathcal{I}$ with variable assignment β. Formula evaluation is defined inductively as a satisfaction relation $\mathfrak{I}\models\varphi$ ("$\mathfrak{I}$ satisfies φ", "$\mathfrak{I}$ is a model of φ", "φ holds in $\mathfrak{I}$"):[17]

$$\mathfrak{I}\models R(t_1,...,t_n) \quad : \text{iff } (\mathfrak{I}(t_1),...,\mathfrak{I}(t_n))\in R^{\mathcal{I}}$$

$$\mathfrak{I}\models\neg\varphi \quad : \text{iff not } \mathfrak{I}\models\varphi$$

$$\mathfrak{I}\models\varphi\wedge\psi \quad : \text{iff } \mathfrak{I}\models\varphi \text{ and } \mathfrak{I}\models\psi$$

$$\mathfrak{I}\models\varphi\vee\psi \quad : \text{iff } \mathfrak{I}\models\varphi \text{ or } \mathfrak{I}\models\psi$$

$$\mathfrak{I}\models\varphi\rightarrow\psi \quad : \text{iff } \mathfrak{I}\models\varphi \text{ implies } \mathfrak{I}\models\psi$$

$$\mathfrak{I}\models\varphi\leftrightarrow\psi \quad : \text{iff } \mathfrak{I}\models\varphi \text{ if and only if } \mathfrak{I}\models\psi$$

$$\mathfrak{I}\models\forall x(\varphi) \quad : \text{iff } \mathfrak{I}\models\varphi \text{ for all } d\in D \text{ holds } \mathfrak{I}\frac{d}{x}\models\varphi$$

$$\mathfrak{I}\models\exists x(\varphi) \quad : \text{iff } \mathfrak{I}\models\varphi \text{ there exists } d\in D \text{ such that } \mathfrak{I}\frac{d}{x}\models\varphi$$

These abstract notions become more tangible when recast in database terminology: A formula $\varphi(x_1,...,x_n)$ over S with free variables[18] $x_1,...,x_n$ defines an *n-ary query q* over the *schema* $\mathbf{S}$. In particular, checking for which tuples $(c_1,...,c_n)$ we have $\mathcal{I}\models\varphi(c_1,...,c_n)$ is exactly the same as running the query q against the database instance $\mathcal{I}$, *i.e.*, $q(\mathcal{I})=\{\ (c_1,...,c_n)\ |\mathcal{I}\models\varphi(c_1,...,c_n)\ \}$. Here, we use $\mathcal{I}$ instead of $\mathfrak{I}$, since all free variables x_i of φ have been substituted by constants c_i (so β is not needed).

A formula φ without free variables is called a sentence or constraint, and corresponds to a yes/no (boolean) query. Let Φ be a set of constraints. We write $\mathcal{I}\models\Phi$, if $\mathcal{I}\models\varphi$ for all $\varphi\in\Phi$ and say "$\mathcal{I}$ *is a model of* Φ". We write $\Phi\models\varphi$ if every model of Φ is also a model of φ, *i.e.*, φ is a (logical) consequence of Φ.[19]

[16] (Integrated Taxonomic Information System, 2006), (Species 2000, 2006), (Universal Biological Indexer and Organizer, 2006).

[17] As usual, 'iff ' means 'if and only if '. The colon ':' indicates that the left-hand side is defined by the right-hand side. $\mathfrak{I}\frac{d}{x}$ is the same as $\mathfrak{I}$, but β is modified to map x to d, *i.e.*, $\beta(x) := d$.

[18] An occurrence of a variable x in j is free if it is not under the scope of a quantifier $\forall x(...)$ or $\exists x(...)$; else it is called bound.

[19] Note the difference between $\mathcal{I}\models\varphi$ and $\Phi\models\varphi$: the former is the satisfaction relation between a structure (database instance) $\mathcal{I}$ and a formula (query) φ; the latter is the consequence relation, stating that all structures $\mathcal{I}$ which satisfy Φ, also satisfy φ. Thus, $\mathcal{I}\models\varphi$ is also called formula evaluation (given $\mathcal{I}$), while the $\Phi\models\varphi$ involves "reasoning" (independent of $\mathcal{I}$).

A.1.1.3. Automated reasoning. The semantic consequence relation $\Phi \models \varphi$ can be "mechanized" using the rules of a FOL calculus. A calculus is based on a provability relation $\Phi \vdash \varphi$, stating that φ can be derived (formally proven) from the formulas in Φ and the derivation rules of the calculus. Φ is called consistent if there is no formula φ such that both $\Phi \vdash \varphi$ and $\Phi \vdash \neg\phi$; otherwise Φ is inconsistent. A calculus is sound if $\Phi \vdash \varphi$ implies $\Phi \models \varphi$ (everything that can be derived is a consequence), and complete if $\Phi \models \varphi$ implies $\Phi \vdash \varphi$ (every consequence can be derived). There are many FOL calculi that are sound and complete, e.g., the Hilbert calculus, the Gentzen calculus, and calculi commonly used in automated reasoning, i.e., resolution and tableaux calculus (Stanford Encyclopedia of Philosophy, 2006).

A.2. *The maximal tractable subalgebra* $\mathbb{R}_5^{28}$

Below are the $\mathbb{R}_{32}$ relations, and the $\mathbb{R}_5^{28}$ subset which leads to polynomial time reasoning.

$\mathbb{R}_{32}$ relations	$\mathbb{R}_5^{28}$	$\mathbb{R}_{32}$ relations	$\mathbb{R}_5^{28}$
{ }	•	{ ≡ }	•
{ ! }	•	{ !, ≡ }	•
{ ⊕ }	•	{ ⊕, ≡ }	•
{ !, ⊕ }	•	{ !, ⊕, ≡ }	•
{ ⊊ }	•	{ ⊊, ≡ }	•
{ !, ⊊ }	•	{ !, ⊊, ≡ }	•
{ ⊕, ⊊ }	•	{ ⊕, ⊊, ≡ }	•
{ !, ⊕, ⊊ }	•	{ !, ⊕, ⊊, ≡ }	•
{ ⊋ }	•	{ ⊋, ≡ }	•
{ !, ⊋ }	•	{ !, ⊋, ≡ }	•
{ ⊕, ⊋ }	•	{ ⊕, ⊋, ≡ }	•
{ !, ⊕, ⊋ }	•	{ !, ⊕, ⊋, ≡ }	•
{ ç, ⊋ }		{ ⊊, ⊋, ≡ }	
{ !, ⊊, ⊋ }		{ !, ⊊, ⊋, ≡ }	
{ ⊕, ⊊, ⊋ }	•	{ ⊕, ⊊, ⊋, ≡ }	•
{ !, ⊕, ⊊, ⊋ }	•	{ !, ⊕, ⊊, ⊋, ≡ }	•

Table 5: The maximal tractable subalgebra $\mathbb{R}_5^{28}$: only relations marked "•" are in $\mathbb{R}_5^{28}$.

A.3. *Automated deduction: overview and examples*

In the following we include several formal, automated proofs mentioned in the paper. We used two automatic reasoners: PROVER9 and MACE4.[20] PROVER9 is a resolution-based first-order logic theorem prover. Thus, in order to prove $\Phi \models \varphi$, *i.e.*, that a formula φ follows from a set of assumptions (or axioms) Φ, PROVER9 adds the negated formula $\neg\varphi$ as a goal to the assumptions, trying to refute the conjunction $\Phi \wedge \neg\varphi$, using (primarily) logic resolution steps: $\Phi \cup \{\neg\varphi\} \vdash \square$. Here, "$\square$" stands for the empty clause, denoting False, and "$\vdash$" denotes the provability relation, based on the legal derivation rules steps of the reasoning calculus at hand (e.g., first-order resolution). If indeed, False can be derived, it follows from the correctness of the calculus that φ is a logical consequence of Φ.

[20] http://www.cs.unm.edu/~mccune/mace4/.

If PROVER9 fails to find a proof, this may be because (a) indeed φ is not a consequence of Φ, or (b) it is, but Prover9 simply could not find it (given the limited time and memory resources). One can then use the Mace4 tool to try and find models of $\Phi \wedge \neg\varphi$, *i.e.*, interpretations $\mathcal{I}$ such that $\mathcal{I} \models (\Phi \wedge \neg\varphi)$ holds. When successful, $\mathcal{I}$ is in fact a counter model for the desired theorem φ, demonstrating that all the assumptions Φ can be satisfied while still falsifying φ.

If Mace4 fails to find a model for $\Phi \wedge \neg\varphi$ after Prover9 also failed to show that $\Phi \wedge \neg\varphi$ is inconsistent, then one cannot discern between the cases (a) and (b). In general, this situation can occur because the logical implication and satisfiability of FOL are undecidable problems. However, as shown in the paper, these questions are decidable for $\mathcal{L}_{\text{tax}}$ and – in the case of using only $\mathbb{R}_5^{28}$ Constraints—even efficiently decidable in polynomial time.

Summarizing, PROVER9 and MACE4 were used in tandem when searching for proofs or for counter-models of $\Phi \models \varphi$, respectively.

A.3.1. Automated reasoning examples for Fig. 2

Fig. 2 presented a number of questions about a pair of taxonomies and a mapping between them. We examined the effects of each of our latent taxonomic assumptions: non-emptiness, disjoint-children, and coverage on each of the questions asked in the figure. Given the three LTAs, there are eight possible LTA combinations (e.g., none of the LTAs, all of the LTAs, just coverage and non-emptiness, etc.).

A.3.1.1. Fig. 2c: Is $C \subset E$ implied? We used Prover9 to find proofs for each of the combinations of possible LTAs. When the hypothesis could not be proven, we used Mace4 to verify that there was a counter example. We found that the hypothesis could only be proven if the coverage LTA was applied. The other LTAs had no effect on the outcome. Below is the proof, in Prover9 syntax, with only the coverage LTA applied. The correspondence between Prover9 syntax and the syntax in Table 4 should be clear.

```
2   c(x)->a(x)      Assumption
4   a(x)<->d(x)     Assumption
7   d(x)->e(x)      Assumption
8   c(x)->e(x)      Goal
12  c(c1)           Deny 8
13  -c(x)|a(x)      Clausify 2
15  -d(x)|e(x)      Clausify 7
17  -e(c1)          Deny 8
19  -a(x)|d(x)      Clausify 4
20  a(c1)           Resolve 12 13
21  d(c1)           Resolve 20 19
22  -d(c1)          Resolve 17 15
23  False           Resolve 21 22
```

A.3.1.2. Fig. 2d: Is $C \equiv D$ possible? Here we used Mace4 to find models of the formulas under various LTA combinations. Mace4 found models for all combinations of LTAs unless non-emptiness and disjoint-children were both assumed. Prover9 could not prove that C≡D does not follow from the other formulas. However, the reason that Mace4 could not find a model for this situation is clear when one looks at the figure: If C≡D, then A, C and D must be identical, i.e., contain the same

elements. If B is non-empty, and B and C are disjoint, then A must contain some element which is not in C . That contradicts the statement that A and C are equivalent.

A.3.1.3. Fig. 2e: Is A≡E possible? We again used Mace4 to find models which satisfy this set of formulas under various combinations of LTAs. In this situation, the assertion is indeed possible regardless of whether or not any LTAs are asserted.

A.3.2. Fig. 5: Inconsistent taxonomies and mappings

The Prover9 theorem prover shows that the non-emptiness, disjoint-children and coverage LTAs render the taxonomies and mappings in Fig. 5 inconsistent. The proof posits an instance of BRhn. It then reasons that the instance cannot be one of BRhs or BRht because of the disjoint-children constraint. The equivalence articulations from these nodes to their counterpart KRhs and KRht nodes imply that the instance of BRhn cannot be an instance of either of these. Through the coverage constraint, the instance cannot be an instance of KRh. Following the equivalence articulation to BRh means that the instance cannot be in BRh. However, this leads to a contradiction, because BRhn implies BRh.

1	BRhn(x)->BRh(x)	Assumption
6	BRh(x)<->KRh(x)	Assumption
7	BRhs(x)<->KRhs(x)	Assumption
8	BRht(x)<->KRht(x)	Assumption
10	(exists x (BRhn(x)))	Assumption
16	BRhn(x)->-BRhs(x)	Assumption
17	BRhn(x)->-BRht(x)	Assumption
21	KRh(x)->KRhs(x) | KRht(x)	Assumption
22	BRhn(c2)	Clausify 10
23	−BRhn(x)|BRh(x)	Clausify 1
24	−BRhn(x)|-BRhs(x)	Clausify 16
25	−BRhn(x)|-BRht(x)	Clausify 17
27	BRhs(x)|-KRhs(x)	Clausify 7
32	−BRhs(c2)	Resolve 24 22
33	BRht(x)|-KRht(x)	Clausify 8
37	−BRht(c2)	Resolve 25 22
43	−KRh(x)|KRhs(x) | KRht(x)	Clausify 21
46	−KRhs(c2)	Resolve 32 27
53	−KRht(c2)	Resolve 37 33
57	−KRh(c2)|KRht(c2)	Resolve 46 43
60	−BRh(x)|KRh(x)	Clausify 6
67	−KRh(c2)	Resolve 57 53
69	BRh(c2)	Resolve 22 23
79	−BRh(c2)	Resolve 67 60
80	False	Copy 79, unit_del 69

REFERENCES

Bachmair, L., Ganzinger, H., Waldmann, U., 1993. Set constraints are the monadic class. In Logic in Computer Science, pp. 75–83.

Beach, J.H., Pramanik, S., Beaman, J.H., 1993. Hierarchic taxonomic databases. In: Fortuner, R. (Ed.), Advances in Computer Methods for Systematic Biology: Artificial Intelligence, Databases, Computer Vision, chapter 15. Johns Hopkins University Press, Baltimore, pp. 241–256.

Benson, L.D., 1948. A treatise on the North American Ranunculi. American Midland Naturalist 40, 1–261.

Berendsohn, W.G., 1995. The concept of "Potential Taxa" in databases. Taxon 44, 207–212.

Berendsohn, W.G., 2003. MoReTax — handling factual information linked to taxonomic concepts in biology. Number 39 in Schriftenreihe fu¨r Vegetationskunde. Bundesamt für Naturschutz.

Blum, S., 2007. Personal Communication.

Clement, J.F., 1991. Birds of the World: A Checklist, fourth edition. Ibis Publishing Co. edition.

Clement, J.F., 2001. Birds of the World: A Checklist, fifth edition. Ibis Publishing Co. edition.

Ebbinghaus, H.-D., Flum, J., Thomas, W., 1994. Mathematical Logic, second edition. Springer.

Franz, N.M., Peet, R.K., Weakley, A.S., 2006. On the Use of Taxonomic Concepts in Support of Biodiversity Research and Taxonomy. Proceedings of the New Taxonomy Symposium. http://academic.uprm.edu/franz/publications-pdf/UseTaxConcepts.pdf.

Geoffroy, M., Berendsohn, W.G., 2003. In: Berendsohn (Ed.), The Concept Problem in Taxonomy: Importance, Components, Approaches, pp. 5–14. 2003.

Geoffroy, M., Güntsch, A., 2003. In: Berendsohn (Ed.), Assembling and Navigating the Potential Taxon Graph, pp. 71–82. 2003.

Integrated Taxonomic Information System, 2006. http://www. itis.gov.

Jonsson, P., Drakengren, T., 1997. A complete classification of tractability in RCC-5. Journal of Artificial Intelligence Research 6, 211–221.

Kalfoglou, Y., Schorlemmer, M., 2002. Information flow based ontology mapping. In Proceedings of the 1st International Conference on Ontologies, Databases and Application of Semantics (ODBASE'02). Irvine, CA, USA, pp. 1132–1151.

Kartesz, J.T., 2004. Synthesis of North American flora. BONAP, North Carolina Botanical Garden.

Kennedy, J., Kukla, R., Paterson, T., 2005. Scientific names are ambiguous as identifiers for biological taxa: their context and definition are required for accurate data integration. In 2nd Intl. Workshop on Data Integration in the Life Sciences (DILS), LNCS 3615, pp. 80–95.

Kent, R., 2000. The information flow foundation for conceptual knowledge organization. In 6th International Conference of the International Society for Knowledge Organization.

Koperski, M., Sauer, M., Braun, W., Gradstein, S., 2000. Referenzliste der Moose Deutschlands. Schrifteneihe für Vegetationskunde 34, 1–519.

Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J., Zhao, Y., 2006. Scientific workflow management and the Kepler system. Concurrency and Computation: Practice & Experience 18 (10), 1039–1065.

Mitra, P., Wiederhold, G., Kersten, M., 2000. A graph-oriented model for articulation of ontology interdependencies. Lecture Notes in Computer Science 1777, 86.

Peet, R.K., 2005. Ranunculus dataset.

Species 2000, 2006. http://www.sp2000.org.

Stanford Encyclopedia of Philosophy, 2006. Automated Reasoning. http://plato.stanford.edu/entries/reasoning-automated/.

Taxonomic Concept Schema, 2006. http://www.tdwg.org/subgroups/tnc/tcs-schema-repository/.

Taxonomic Data Working Group, 2006. http://www.tdwg.org.

Thau, D., Ludäscher, B., submitted for publication. Toward Optimizing CleanTAX: An Automated Reasoning Method for Taxonomies and Articulations.

Universal Biological Indexer and Organizer, 2006. http://www.ubio.org.

Weakley, A.S., 2006. Flora of the Carolinas, Virginia, Georgia, and Surrounding Areas. University of North Carolina Herbarium. http://www.herbarium.unc.edu/flora.htm.

Wheeler, Q.D., 2004. Taxonomic triage and the poverty of phylogeny. Philosophical transactions– Royal Society of London. Biological sciences 359 (1444), 571–583.