

A Logical Geo-Ontology Design Pattern for Quantifying over Types

David Carral Martínez
Kno.e.sis Center
Wright State University
Dayton OH, USA
carral.2@wright.edu

Krzysztof Janowicz
Department of Geography
University of California,
Santa Barbara
Santa Barbara, CA, USA
jano@geog.ucsb.edu

Pascal Hitzler
Kno.e.sis Center
Wright State University
Dayton OH, USA
pascal.hitzler@wright.edu

ABSTRACT

Ontology design patterns ease the engineering of ontologies, improve their quality, foster reusability, and support the alignment of ontologies by acting as common building blocks or strategies for reoccurring modeling problems. This makes ontology design patterns key enablers of semantic interoperability and, hence, a crucial technology for representing the body of knowledge of such heterogeneous domains as the geosciences. While different types of patterns can be distinguished, existing work on geo-ontology design patterns has solely focused on content patterns, i.e., design solutions for domain classes and relationships. In this work, we propose a logical pattern that addresses a frequent modeling problem that has hampered the development of sophisticated geo-ontologies in the past, namely how to model the quantification over types. We argue for the need for such a pattern, explain why it is difficult to model, demonstrate how to implement it using the Web Ontology Language OWL, and finally show how it can be applied to modeling concepts such as biodiversity.

Categories and Subject Descriptors

I.2.12 [Intelligent Web Services and Semantic Web]: [Ontology design]; I.2.4 [Knowledge Representation Formalisms and Methods]: [Representation languages]; I.2.13 [Knowledge Management]: [Knowledge reuse]

General Terms

Theory

Keywords

Geo-Ontology Design Patterns, Geospatial Semantics, Interoperability, Heterogeneity

1. INTRODUCTION

Knowledge engineering is increasingly recognized as a successful approach to improve the publication, discovery, reuse, and integration of heterogeneous data. Examples include the success of Linked Data [1] for publishing and reusing data as well as Google's new Knowledge Graph with its *things not strings* slogan offering a lightweight approach to semantic search. Typically, systems and

infrastructures that employ knowledge engineering methods consist of two parts, domain vocabularies represented in some formal knowledge representation language, i.e., ontologies, and software that exploits the encoded facts, e.g., to improve search beyond simple keywords. Consequently, the added value of knowledge engineering depends on the strengths of the used representation language as well as the represented knowledge. Numerous formal languages have been developed over the last decades, the W3C standardized Web Ontology Language (OWL) [15, 16] being the most successful language in terms of adoption. These languages differ in their expressivity, i.e., in the ability to assign a formal semantics to language constructs, and computational characteristics such as their complexity and decidability [10].

While expressivity determines which domain facts can be encoded in a way exploitable by software, the added value largely depends on what knowledge is actually captured, i.e., the quality and quantity of formalization. To give a simplified example, a knowledge representation language that does not support transitivity cannot be adequately used to model meronomies [34] and, therefore, will not be suitable for geographic information. Transitivity alone, however, does not add any value without introducing relations such as *part-of*. Therefore, the constant progress in developing knowledge representation languages does not automatically translate to more or better ontologies.

Engineering high-quality ontologies is a difficult task. Several methodologies, such as Methontology [5], have been presented over the last years. They support the process of ontology engineering by specifying phases ranging from the initial planing to long-term maintenance cycles and detail the role of different kinds of actors, such as domain experts, users, and knowledge engineers. Originally, researchers envisioned a stacked approach in which local, application-centric ontologies are developed by refining the terminology from domain ontologies which, in turn, use top-level ontologies as common and domain-independent foundation [12]. Besides advantages such as reusability and quality control at the upper levels, the stacked approach has some major drawbacks. First, it does not support a *staged* introduction of semantics and ontologies as the development of application-centric ontologies requires the existence of commonly agreed domain and top-level ontologies. Second, domain experts and knowledge engineers have to understand and agree to the hundreds of statements encoded at the upper-levels, while they can hardly influence them or request changes.

Ontology design patterns [8] offer a flexible alternative to the stacked approach. They are best thought of as strategies or building blocks that support domain experts and knowledge engineers

in modeling recurring problems. While they are analogous to the highly successful software engineering patterns [7], different kinds of ontology design patterns can be distinguished. Examples include logical patterns, content patterns, or alignment patterns. Logical patterns are workarounds to problems which are a consequence of the formal semantics of the used knowledge representation language. Consequently, they are independent of specific domains or applications. Content patterns, in contrast, are strategies or building blocks applied to model frequently occurring domain facts. While patterns may vary in size and complexity, they are usually self-contained and minimal in terms of their ontological commitments. Many successful ontology design patterns have been developed in recent years. For instance, most parts of the DOLCE Ultra Light foundational ontology have been repackaged into a collection of patterns. One example of a pattern relevant for GIScience is the Stimulus-Sensor-Observation (SSO) pattern that forms the core of the W3C SSN-XG semantic sensor network ontology and has been used in several applications [2]. Other examples include a pattern for referential qualities introduced by Ortmann and Daniel [27].

For a highly diverse research community such as the geosciences, a pattern based approach to employing semantics for data publishing, discovery, reuse, and integration is promising for several reasons. First, and even within highly specialized disciplines, it is difficult to reach agreement on a common definition for domain vocabulary. Aligning these definitions with abstract classes from top-level ontologies such as endurants and perdurants is even more challenging. As discussed by Galton [6] many geographic feature types are multi-aspect phenomena which can be classified as events or physical objects at the same time – lenticular clouds or waves being common examples. Similarly, hills can be defined as *physical objects*, *features*, or *amount of matter* [32] which are among the core distinctions proposed by the DOLCE foundational ontology [9] for modeling physical endurants. Consequently, while many patterns have been developed and are actively applied, domain ontologies for the geosciences are largely missing – SWEET being a rare exception [28]. Second, it is a common misconception that interoperability can only be established (similar to interlingual machine translation) by aligning local, application centric ontologies with a common and global top-level ontology. Instead, ontologies can be directly combined using alignment methods and servers such as AgreementMaker [3], BLOOMS [18], or the Alignment API [4]. Patterns improve the quality of alignments by fostering horizontal integration. Third, patterns foster semantic interoperability without restricting the heterogeneity brought in by local ontologies that reflect multiple perspectives on geographic space [19]. This is especially important for knowledge infrastructures such as NSF’s Earth-Cube [14] that exploit the variety component of Big Data [20].

In this work, we introduce a logical pattern that allows ontology engineers to model the quantification over types. This is a frequent modeling problem for geo-ontologies and cannot be directly approached in any of the standardized Semantic Web knowledge representation languages. As a logical pattern, our work is independent of a specific application and is, therefore, highly reusable. The proposed pattern can be applied to model concepts such as cartographic scale or biodiversity. For instance, it can capture the effect that during the process of cartographic abstraction, and depending on the selected scale, certain types of features, e.g., sights, are removed while others, e.g., hospitals, are introduced [24, 29].

The remainder of this paper is structured as follows. First, we set the preliminaries relevant to the understanding of the proposed

work by recalling key aspects of the OWL knowledge representation language. Next, in section 3, we introduce our new language primitive to model the quantification over types in OWL. Besides defining the logical pattern, we also present a macro to ease ontology engineering. As the long-term success of an ontology design pattern is evaluated by its applicability to model different domain facts, section 4 describes how to use the pattern. While we focus on the case of biodiversity, other examples are highlighted in addition. We also point out existing limitations of our current implementation, namely reasoning in the presence of ambiguous data. We conclude the paper with summarizing the presented work and pointing to further research. Note that section 3 also contains a technical theorem which supports the feasibility of our proposed pattern. For reasons of readability, the formal proof of this theorem is moved to the appendix.

2. PRELIMINARIES

We start by introducing the description logic (DL) $SR\mathcal{OIQ}$, the logic underpinning the Web Ontology Language’s profile $OWL\ 2\ DL$ [15]. This definition is included to make the paper self-contained. Due to space constraints we only recall key aspects here. For a more detailed definition see [17]; a gentle introduction is given in [16].

The description logic language $SR\mathcal{OIQ}$ is based on a *signature* $\Sigma = \langle N_I, N_C, N_R, N_V \rangle$, where N_I , N_C , N_R , and N_V are finite and pairwise disjoint sets of *individual names*, *concept names*, *role names*, and *variables*. The set N_R is partitioned into disjoint sets N_R^s of *simple role names* and N_R^n of *non-simple role names* as defined in [17]. For the rest of this paper, we assume that a signature Σ has been fixed and so omit further references to it.

Definition 1. The sets \mathbf{C} of $SR\mathcal{OIQ}$ concepts (or classes) and \mathbf{R} ($\mathbf{R}^s/\mathbf{R}^n$) of (simple/non-simple) $SR\mathcal{OIQ}$ roles (called *properties* in the OWL 2 specification) are defined by the grammar

$$\begin{aligned} \mathbf{R}^s &::= N_R^s \mid (N_R^s)^- \mid U \\ \mathbf{R}^n &::= N_R^n \mid (N_R^n)^- \mid U \\ \mathbf{R} &::= \mathbf{R}^s \mid \mathbf{R}^n \\ \mathbf{C} &::= \top \mid \perp \mid N_C \mid \{N_I\} \mid \neg\mathbf{C} \mid \mathbf{C} \sqcap \mathbf{C} \mid \mathbf{C} \sqcup \mathbf{C} \mid \\ &\quad \exists\mathbf{R}.\mathbf{C} \mid \forall\mathbf{R}.\mathbf{C} \mid \exists\mathbf{R}^s.\text{Self} \mid \leq_k \mathbf{R}^s.\mathbf{C} \mid \geq_k \mathbf{R}^s.\mathbf{C} \end{aligned}$$

where k is any non-negative integer. U is the *universal role*, and \top and \perp are the top and bottom concepts. Concepts $\{a\}$ with $a \in N_I$ are called *nominals*. O denotes the set of nominals.

Roles of the form R^- with $R \in N_R^s \cup N_R^n$ are called *inverse roles*. We define a function $\text{Inv} : \mathbf{R} \rightarrow \mathbf{R}$ as follows: For $R \in N_R$, set $\text{Inv}(R) := R^-$ and $\text{Inv}(R^-) := R$.

$SR\mathcal{OIQ}$ knowledge bases (KB) are constructed from axioms as follows:

Definition 2. Given roles $R, S_i \in \mathbf{R}$, a *generalized role inclusion axiom (RIA)* is a statement of the form $S_1 \circ \dots \circ S_k \sqsubseteq R$, with either $R \notin \mathbf{R}^n$, or $k = 1$ and $S_1 \in \mathbf{R}^s$. A set of RIAs is *regular* if there is a strict partial order \prec on \mathbf{R} such that

- if $R \notin \{S, \text{Inv}(S)\}$, then $S \prec R$ if and only if $\text{Inv}(S) \prec R$; and

Table 1: Semantics of *SRQLQ*: \mathcal{I} an interpretation; $A, B \in N_C$; $C, D \in \mathbf{C}$; $V \in N_R$; $R_{(i)}, S \in \mathbf{R}$; $a \in N_I$; $x \in N_V$; $t, u \in \mathbf{T}$.

Name	Syntax	Semantics
concept name	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
role name	V	$V^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
individual name	a	$a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
top	\top	$\Delta^{\mathcal{I}}$
bottom	\perp	\emptyset
existential restriction	$\exists R.C$	$\{\delta \mid \text{there is } \epsilon \text{ with } \langle \delta, \epsilon \rangle \in R^{\mathcal{I}} \text{ and } \epsilon \in C^{\mathcal{I}}\}$
universal restriction	$\forall R.C$	$\{\delta \mid \text{for all } \epsilon \text{ with } \langle \delta, \epsilon \rangle \in R^{\mathcal{I}} \text{ we have } \epsilon \in C^{\mathcal{I}}\}$
self restriction	$\exists R.\text{Self}$	$\{\delta \mid \langle \delta, \delta \rangle \in R^{\mathcal{I}}\}$
concept complement	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
concept conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
concept disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
qualified number restrictions	$\leq n R.C$	$\{\delta \mid \#\{\langle \delta, \epsilon \rangle \in R^{\mathcal{I}} \mid \epsilon \in C^{\mathcal{I}}\} \leq n\}$
	$\geq n R.C$	$\{\delta \mid \#\{\langle \delta, \epsilon \rangle \in R^{\mathcal{I}} \mid \epsilon \in C^{\mathcal{I}}\} \geq n\}$
universal role	U	$\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	V^-	$\{\langle \delta, \epsilon \rangle \mid \langle \epsilon, \delta \rangle \in V^{\mathcal{I}}\}$
concept assertion (ABox)	$A(t)$	$t^{\mathcal{I}} \in A^{\mathcal{I}}$
role assertion (ABox)	$V(t, u)$	$\langle t^{\mathcal{I}}, u^{\mathcal{I}} \rangle \in V^{\mathcal{I}}$
TBox axiom	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
RBox axiom (RIA)	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
	$R_1 \circ \dots \circ R_n \sqsubseteq S$	$R_1^{\mathcal{I}} \circ \dots \circ R_n^{\mathcal{I}} \subseteq S^{\mathcal{I}}$ where ‘ \circ ’ denotes the usual composition of binary relations

- every RIA has the form $R \circ R \sqsubseteq R$, $\text{Inv}(R) \sqsubseteq R$, $R \circ S_1 \circ \dots \circ S_k \sqsubseteq R$, $S_1 \circ \dots \circ S_k \circ R \sqsubseteq R$, or $S_1 \circ \dots \circ S_k \sqsubseteq R$, with $R, S_i \in \mathbf{R}$ and $S_i \prec R$ for each $i \in \{1, \dots, k\}$.

An *RBox axiom* is a RIA. A *TBox axiom* (or *general concept inclusion axiom*, *GCI*) is an expression $C \sqsubseteq D$ where $C, D \in \mathbf{C}$. An *ABox axiom* is any expression of the form $C(a)$ or $R(a, b)$ where $C \in N_C$, $R \in N_R$, and $a, b \in N_I$. A *SRQLQ axiom* is any ABox, TBox, or RBox axiom, and a *SRQLQ knowledge base* is a set of *SRQLQ axioms* such that the subset of RBox axioms is regular.

The semantics of *SRQLQ* knowledge bases is defined in model-theoretic terms as usual, following Table 1.

All role assertions but role disjointness $\text{disj}(S, \text{Inv}(S))$ can be expressed using the set of presented constructors and therefore we do not explicitly include them in the definition: asymmetry ($\text{disj}(S, \text{Inv}(S))$), reflexivity ($\top \sqsubseteq \exists S_{\text{aux}}.\text{Self}$, $S_{\text{aux}} \sqsubseteq R$), irreflexivity ($\exists S.\text{Self} \sqsubseteq \perp$), symmetry ($\text{Inv}(R) \sqsubseteq R$), transitivity ($R \circ R \sqsubseteq R$).

The *empty (bottom) role* N can also be defined as syntactic sugar via the axiom $\exists N.\top \sqsubseteq \perp$.

For ease of notation, we will make use of a restricted form of the *concept product* construct in DLs [30]. We restrict this to axioms of the form $R \sqsubseteq D \times D$, where R is a role and D is a concept. For the purpose of this paper (and this restricted use), we can simply

think of this as a macro: $R \sqsubseteq D \times D$ expands to the two axioms

$$\begin{aligned} \exists R.\top \sqsubseteq D & \quad \text{i.e., } D \text{ is the domain of } R \\ \top \sqsubseteq \forall R.D & \quad \text{i.e., } D \text{ is the range of } R. \end{aligned}$$

Recall, that the semantics of the Web Ontology Language OWL adheres to the Open World Assumption (OWA): statements which are not logical consequences of a given knowledge base are not necessarily considered false. The OWA is a reasonable assumption to make in the World Wide Web context (and thus for Semantic Web applications). However, for the definition of the pattern in the next section we require some form of Closed World Assumption (CWA) for some concepts. That is, some statements which are not logical consequences of a given knowledge base should be considered *false*.

There are several prominent ways how to add closed world reasoning to OWL, see e.g. [21, 22] for comprehensive overviews. For the purpose of our specific pattern we make use of the *grounded circumscription* approach presented in [31] that allows us to include a form of *Local Closed World Assumption* (LCWA) [11] for some classes in the knowledge base. We explain this in more detail later.

3. THE PATTERN

In this section, we introduce the proposed logical pattern to model the quantification over types and discuss an OWL macro to make the new language construct available to ontology engineers and domain experts. We also briefly discuss a proposal for a corresponding extension of the standardized OWL syntax, and prove a theoretical result concerning algorithmization of reasoning with this pattern. Application examples of the pattern will be given in section 4.

3.1 Introducing the Pattern

We first present the formal definition for the logical pattern which we propose, and will then discuss the rationale behind it from a bird's eye perspective.

The pattern which we introduce is used for *type-count comparison* – we correspondingly call it the *type-count comparison pattern*. Syntactically, we write it as

$$R \equiv D \times D|_{C_1, \dots, C_n},$$

where R is a role name, and D and the C_i are concepts.

Intuitively, the semantics of this pattern is as follows: Two individuals x and y shall be connected by the role R if and only if x is contained in strictly more different classes C_i than y . E.g., say x is contained in C_1 and C_5 (but not in any other C_i , while y is contained in C_2 (but not in any other C_j), then we would like to infer $R(x, y)$. The notation using a vertical bar is borrowed from a very common mathematical notation used for restricting functions to subsets of their domains.

While this informal explanation of the pattern is intuitively appealing, it is not entirely straightforward to cast it into a formal semantics. While we will give a formal semantics in section 3.3 below, let us try to shed some light on the difficulties by contemplating an abstract example.

Consider the knowledge base consisting of the following statements.

$$\begin{aligned} R &\equiv T \times T|_{C_1, C_2} \\ C_1(a) \\ C_2(a) \\ C_1(b) \end{aligned}$$

From this knowledge base we would like to infer $R(a, b)$, since a is known to be contained in the two classes C_1 and C_2 , while b is only known to be contained in C_1 .

However, now assume we add the axiom $C_2(b)$ to the knowledge base. Under this new knowledge base, we would no longer infer $R(a, b)$, since both a and b are contained in two of the classes. Note that the *addition* of the axiom $C_2(b)$ means that a previously drawn inference, namely $R(a, b)$, is no longer a valid inference. This observation shows that we are in fact considering a so-called *non-monotonic* semantics. Such non-monotonic semantics usually arise in the context of some kind of (local) world closure as discussed at the end of section 2.

In our formal semantics we will therefore have to reflect this, and introduce some non-monotonic semantic construct. We will discuss this further in section 3.3.

Before we continue with our formal treatment, however, let us first introduce a much more convenient macro. Indeed, the notation

$$R \equiv D \times D|_{C_1, \dots, C_n}$$

introduced above is cumbersome if n , i.e., the number of classes used in the type count, is large. It would be preferable to have a notation which does not require the explicit listing of all n classes as part of the pattern.

To do this, we make use of the so-called *punning* feature available

in OWL 2, which makes it possible to use classes also as individuals in the same knowledge base. More precisely, we introduce the notation

$$R \equiv|_C D \times D$$

as a macro for

$$R \equiv D \times D|_{C_1, \dots, C_n},$$

where C_1, \dots, C_n are exactly all individuals contained in the class C . The C_i are in this case used both as individuals and as concepts, which is allowed due to OWL 2 punning. The semantics of punning in OWL 2 furthermore guarantees that there are no unwanted inferences (see, e.g., the explanations in [16]). To make sure that there are no other additional – unwanted – inferences, and to ensure that the expansion of the macro is straightforward, we have to impose two very mild conditions: On the one hand, we impose that we have ABox statements $C(C_i)$, for all $i = 1, \dots, n$. On the other hand, we impose that the concept name C is not used for any other purpose in the knowledge base. This way, we can easily retrieve all C_i by a simple SPARQL query [33] over the knowledge base.

3.2 Functional Syntax

We define a syntax for the pattern as an extension of the OWL 2 Functional Syntax [25]. We do this by minimally modifying the existing grammar specification, and by adding a new reserved word to express the type-count comparison pattern.

Consider the pattern macro

$$R \equiv|_C D \times D.$$

We first need to add classes C_i as instances of the class C , so we just list them as instances of the class using the existing syntax:

ClassAssertion($C C_i$)

We reserve a new word *TypeCountComparison* to express the actual pattern:

SubObjectPropertyOf(TypeCountComparison($D C$) R)

To include this type of axioms in the functional syntax we make the following modifications to the grammar. We modify the production rule of the non-terminal **SubObjectPropertyOf** to:

```
SubObjectPropertyOf :=
'SubObjectPropertyOf' '(' axiomAnnotations
subObjectPropertyExpression
superObjectPropertyExpression ')' |
'SubObjectPropertyOf' '(' axiomAnnotations
TypeCountComparison superObjectPropertyExpression ')'
```

We also include a new production rule for the new non-terminal **TypeCountComparison**:

```
TypeCountComparison :=
'TypeCountComparison' '(' ClassExpression ClassExpression ')'
```

We have chosen to use “SubObjectPropertyOf” to represent the pattern, because (i) reusing an existing keyword makes for a light-weight modification to the existing grammar, and because (ii) the pattern can very well also be understood as a sub-property construct of sorts.

Table 2: Expansion of $R \equiv D \times D|_{C_1, \dots, C_n}$

$$\text{Close}(C_i) \quad (1)$$

for all $1 \leq i \leq n$.

$$R \sqsubseteq D \times D \quad (2)$$

$$D \sqsubseteq N_{0,0} \quad (3)$$

$$N_{m-1,k} \sqcap \neg C_m \sqsubseteq N_{m,k} \quad (4)$$

$$N_{m-1,k} \sqcap C_m \sqsubseteq N_{m,k+1} \quad (5)$$

where all $N_{i,j}$ are freshly introduced classes where $m = 1, \dots, n$ and $k = 1, \dots, m-1$ for every m .

$$N_{n,i} \sqsubseteq \exists S_i. \text{Self} \quad (6)$$

where $i = 0, \dots, n$.

$$S_i \circ R \circ S_j \sqsubseteq R_{\text{typeCountViolation}} \quad (7)$$

for all $i \leq j$ where $j = 0, \dots, n$.

$$S_i \circ U \circ S_j \sqsubseteq R \quad (8)$$

for all $i > j$ where $i = 1, \dots, n$.

3.3 Formal Semantics and Reasoning

We now turn to specifying the formal semantics of our type-count comparison pattern

$$R \equiv D \times D|_{C_1, \dots, C_n}.$$

In fact, we provide the formal semantics by means of understanding the pattern as a macro, which expands to a set of *SRQIQ* axioms together with a number of closure specifications in the sense of grounded circumscription [31]. Indeed, the pattern is expanded to the set of axioms listed in Table 2. Let us try to give some intuitive explanations for the axioms.

Axiom (1) lists *closure* predicates in the sense of grounded circumscription. According to [31], a statement such as $\text{Close}(C_i)$ means two things. On the one hand, it means that the class C_i contains only *known* individuals, i.e., it contains only individuals which are named entities in the knowledge base. On the other hand, it means that the class C_i contains *only* individuals which are *necessarily* contained in it – in the sense that there is no model for the knowledge base where only a subset of these is contained in C_i .

Of course, the explanations just given need to be formally specified, and this is done in [31] – and we refrain from repeating formal details. However, let us return to the example from section 3.1, where the non-monotonic nature of our pattern was first discussed. Due to the statement $\text{Close}(C_2)$ we indeed obtain that – before the addition of $C_2(b)$ to the knowledge base – we can infer $\neg C_2(b)$ from the knowledge base, as desired. Of course, after addition of $C_2(b)$ we infer $C_2(b)$, also as desired, and we see that this kind of closure captures non-monotonicity as required for our purpose.

Let it be remarked here that the use of grounded circumscription for the purpose of this paper is by no means the only choice. However, as argued in [31], grounded circumscription appears to be a rather intuitive approach to world closure, which also appears to be compatible with OWL design choices. In fact, for simple cases such as the ones in which our pattern may be used, the semantics of the different approaches for non-monotonicity will likely coincide anyway. Investigating the most useful ways of introducing world

closure to description logics is a topic currently under investigation, and while the basic (and, arguably, practically relevant) cases are reasonably well understood, theoretical foundations are still being worked out (see, e.g., [21]).

Returning to Table 2, axiom (2) is simply the expected domain and range restriction for R .

Axioms (3) to (5) are more tricky. The key idea behind them is that of counting, for each x in D , how many of the classes C_i contain x . This is done in such a way that x is contained in exactly m of the classes C_i if x is contained in the class $N_{n,m}$. Axiom (6) is then a sort of typecasting, called *rolification* [23], which captures membership of x in the class $N_{n,i}$ in terms x being related to itself by the freshly introduced role S_i . This typecasting is required for later use in axioms (8) and (9).

The axioms (7) can be understood as constraints which capture violations in the data. To understand this, consider the knowledge base consisting of the following statements.

$$\begin{aligned} R &\equiv \top \times \top|_{C_1, C_2, C_3, C_4} \\ C_1(a) \\ C_1(b) \\ C_2(b) \\ R(a, b) \end{aligned}$$

In this case, we would obtain $R_{\text{typeCountViolation}}(a, b)$ as logical consequence of the knowledge base. Thus, if desired, the knowledge base can be queried for such violations in the data, in order to take appropriate action. We have chosen this axiomatization as a service to the user, who can utilize this to inspect the knowledge base in order to search for incomplete or contradictory information. This approach to capturing data violations is indeed very common in rule-based systems, including logic programming.

Let us dwell on this a little longer, and briefly discuss an alternative axiomatization, which is realized by adding the statement $R_{\text{typeCountViolation}} \equiv N$, which declares $R_{\text{typeCountViolation}}$ to be equivalent to the bottom (i.e., empty) role. At first sight, one might expect that the resulting knowledge base is inconsistent. However, it is not. Rather, in any model a would have to be contained in additional classes C_i , e.g., we may have a also contained in C_3 and C_4 . This means that additional class memberships are now *inferred* due to the non-monotonic nature of grounded circumscription. Note that our base pattern can be used either way, by either additionally including, or not, the axiom $R_{\text{typeCountViolation}} \equiv N$.

The axioms (8), which utilize the top role U , specify that two individuals x and y are indeed related through R if x is contained in strictly more classes C_i than y is.

The explanations just given already provide a clear understanding why our treatment of the pattern captures the intended meaning. We have also already given some examples which show how the pattern can be put to work. As a sort of formal support for our proposed pattern, we now also provide the following theorem.

THEOREM 1. *Let K be a *SRQIQ* knowledge base (not containing concepts C_1, \dots, C_n or any sub-formulas of these concepts in the *TBox*), and let K' be the knowledge base which results from*

adding the type-count comparison pattern

$$R \equiv \top \times \top |_{C_1, \dots, C_n}$$

to K . Furthermore, for any individual a in K let

$$c(a) = |\{i \mid K \models C_i(a)\}|,$$

i.e., $c(a)$ is the number of different classes C_i of which a can be inferred to be an element.

(i) Then $c(a) > c(b)$ implies $K' \models R(a, b)$.

Now assume furthermore that there is no pair (a, b) of individuals such that $K \models R_{\text{TypeCountViolation}}(a, b)$.

(ii) Then $K' \models R(a, b)$ also implies $c(a) > c(b)$.

A formal proof of this theorem can be found in the appendix. The assumptions made for the statement of the theorem are rather strong, in order to avoid an extensive formal treatment which would feel out of place in this paper. We will discuss some of these assumptions at the end of section 4.1.

The expansion from Table 2 does not only provide a formal semantics for our pattern, but also shows how automated reasoning with the pattern can be realized. Indeed, it was shown in [31] that adding grounded circumscriptive closure axioms to OWL retains decidability, and that this can be handled algorithmically.

Furthermore, note that the number of axioms into which

$$R \equiv \top \times \top |_{C_1, \dots, C_n}$$

is expanded is in fact polynomial (more precisely, quadratic) in n . So adding the pattern to grounded circumscriptive \mathcal{SROIQ} does not cause any increase in the computational complexity of the language. Note, however, that grounded circumscription itself in general causes a rise in complexity compared to the base language. But this is avoidable in specific cases which may often apply to use cases of our pattern.

4. APPLICATION OF THE PATTERN

In this section we outline how the presented logical pattern can be applied to model challenging domain facts that could not be specified in any Semantic Web knowledge representation language before. While we focus on biodiversity as example, other application areas will be highlighted as well. It is important to note that our examples are simplified for demonstration purpose, we do *not* claim that they capture all of the relevant scientific work, complex interplay, and nuances. The definition of domain knowledge is up to the respective experts, while our task as ontology engineers is to develop the *methods* that enable the logical representation of this knowledge [19].

4.1 Application to Biodiversity

Biodiversity is a major concept in ecology and evolutionary biology. For instance, it is studied in ecological forecasting to estimate the future development of ecosystems in response to environmental change. Due to the range of involved scientific disciplines, different measures, and indices, it is difficult to agree on a common, 'simple, comprehensive, and fully operational' [26] definition of biodiversity. This makes the proposed pattern approach in which communities can maintain their own definitions more attractive than trying

to arrive at a common and stable global definition. Simplifying, '[b]iodiversity is the totality of genes, species, and ecosystems in a region' [35]. Species diversity is one type of biodiversity and can be expressed in terms of species richness and species evenness [13]. We will restrict our example to the first component and just focus on species richness as the *count of different species in a given region*.

This apparently trivial definition highlights two challenges for the engineering of ontologies.¹ First, the term *count* is difficult to capture and unclear in terms of its logical consequences. Second, the quantification is done over different types, not individuals. Each single species should be modeled as a class to ensure that the formal semantics of OWL can be employed for reasoning, e.g., to automatically select all species of the *Squamata* Order such as snakes and lizards. Strictly speaking, this calls for a second-order language and cannot be directly expressed in OWL DL, thus, requiring our proposed logical pattern.

The first mentioned difficulty is a classic in communicating about ontologies between domain experts interested in conceptual modeling and knowledge engineers interested in the inferential power of their knowledge representation languages. Natural language descriptions such as *count of different species in a given region* do not translate well to a set-theoretic semantics. To address this, we propose to translate such definitions to binary relationships. The relation will hold between two regions and could be labeled *is more biodiverse than*, or short *imbt*. Put simply, given two habitats h_1 and h_2 , $imbt(h_1, h_2)$ holds if, and only if, h_1 has a greater count of different species than h_2 . Strictly speaking, from a GIS perspective, we need to introduce a topological *inside* relation between a region represented as a polygon and point-like observations of individuals. For each individual of a distinct type, we conclude that this species occurs in the given region.² The *imbt* relation allows us to employ the inferential semantics of OWL by using reasoning services, e.g., to infer *imbt* relations between regions. This was not possible with the original definition given before. The second part, i.e., the quantification over types, will be addressed using our proposed pattern as follows. It will also ensure that we only count distinct species.

We work out a simple example for the pattern to improve the understandability of the approach. We will use the biodiversity case previously described. Assume we have a knowledge base with only three different species, S_1 , S_2 , and S_3 , where we want to model the property

$$imbt \equiv E \times E |_{\exists \text{habitatOf}.S_1, \exists \text{habitatOf}.S_2, \exists \text{habitatOf}.S_3},$$

where E stands for the class of ecosystems.

We list all the new axioms that the transformation of the primitive will produce:

$$\text{Close}(\exists \text{habitatOf}.S_1)$$

$$\text{Close}(\exists \text{habitatOf}.S_2)$$

$$\text{Close}(\exists \text{habitatOf}.S_3)$$

These closure statements play a central role for our pattern. Intuitively, their effect is as follows. Assume there is an ecosystem e , and that there is no positive evidence or data whatsoever which

¹which will, thus, also appear in more complex definitions of biodiversity and related concepts.

²In practice, we often know which species to expect.

would suggest that e is a habitat of species S_1 . Then, the closure axioms above make it possible to conclude $\neg\exists\text{habitatOf}.S_1(e)$, which would not be possible under the open world assumption.

$$\begin{aligned}\text{imbt} &\sqsubseteq E \times E \\ E &\sqsubseteq N_{0,0}\end{aligned}$$

Axiom $\text{imbt} \sqsubseteq E \times E$ is a macro for axioms $\top \sqsubseteq \forall\text{imbt}.E$ (range of imbt is E) and $\exists\text{imbt}.\top \sqsubseteq E$ (domain of imbt is E) as stated in section 2. With these two we set the domain and range of property imbt to the class E .

$$\begin{aligned}N_{0,0} \sqcap \neg\exists\text{habitatOf}.S_1 &\sqsubseteq N_{1,0} \\ N_{0,0} \sqcap \exists\text{habitatOf}.S_1 &\sqsubseteq N_{1,1}\end{aligned}$$

Note that concepts $N_{1,0}$ and $N_{1,1}$ are disjoint and that for every individual e in E we can conclude $N_{1,0}(e)$ (due to the closure predicate) unless we have evidence to the contrary.

$$\begin{aligned}N_{1,0} \sqcap \neg\exists\text{habitatOf}.S_2 &\sqsubseteq N_{2,0} \\ N_{1,0} \sqcap \exists\text{habitatOf}.S_2 &\sqsubseteq N_{2,1} \\ N_{1,1} \sqcap \neg\exists\text{habitatOf}.S_2 &\sqsubseteq N_{2,1} \\ N_{1,1} \sqcap \exists\text{habitatOf}.S_2 &\sqsubseteq N_{2,2}\end{aligned}$$

The observation just given carries through the chain of $N_{i,j}$ predicates.

$$\begin{aligned}N_{2,0} \sqcap \neg\exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,0} \\ N_{2,0} \sqcap \exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,1} \\ N_{2,1} \sqcap \neg\exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,1} \\ N_{2,1} \sqcap \exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,2} \\ N_{2,2} \sqcap \neg\exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,2} \\ N_{2,2} \sqcap \exists\text{habitatOf}.S_3 &\sqsubseteq N_{3,3}\end{aligned}$$

We now have that, for a knowledge base containing this set of axioms, every individual $e \in E$ is contained in at most one $N_{3,x}$ where the x determines the number of different species inhabiting the ecosystem.

$$\begin{aligned}N_{3,0} &\sqsubseteq \exists S_0.\text{Self} \\ N_{3,1} &\sqsubseteq \exists S_1.\text{Self} \\ N_{3,2} &\sqsubseteq \exists S_2.\text{Self} \\ N_{3,3} &\sqsubseteq \exists S_3.\text{Self}\end{aligned}$$

Roles S_i can be used in role chains to check the number of different species that inhabit a given ecosystem.

$$\begin{aligned}S_0 \circ \text{imbt} \circ S_0 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_0 \circ \text{imbt} \circ S_1 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_0 \circ \text{imbt} \circ S_2 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_0 \circ \text{imbt} \circ S_3 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_1 \circ \text{imbt} \circ S_1 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_1 \circ \text{imbt} \circ S_2 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_1 \circ \text{imbt} \circ S_3 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_2 \circ \text{imbt} \circ S_2 &\sqsubseteq R_{\text{typeCountViolation}} \\ S_2 \circ \text{imbt} \circ S_3 &\sqsubseteq R_{\text{typeCountViolation}}\end{aligned}$$

The previous set of axioms flags a possible violation. This occurs when the imbt property has been defined to hold between two individuals while the data does not support this. Typically, this functionality will be exploited during ontology engineering, data entry,

or cleaning of the knowledge base.

$$\begin{aligned}S_1 \circ U \circ S_0 &\sqsubseteq \text{imbt} \\ S_2 \circ U \circ S_0 &\sqsubseteq \text{imbt} \\ S_2 \circ U \circ S_1 &\sqsubseteq \text{imbt} \\ S_3 \circ U \circ S_0 &\sqsubseteq \text{imbt} \\ S_3 \circ U \circ S_1 &\sqsubseteq \text{imbt} \\ S_3 \circ U \circ S_2 &\sqsubseteq \text{imbt}\end{aligned}$$

This last set of axioms automatically produces the imbt connections between individuals $x \in E$. Note that, as stated in the proof, the whole set of axioms added to the knowledge base is polynomial with respect to the number of $\exists\text{habitatOf}.S_i$ classes in the knowledge base.

Consider now the following example knowledge base.

$$\begin{array}{ccc}E(e1) & E(e2) & E(e3) \\ S_1(s11) & S_2(s21) & S_3(s31) \\ S_1(s12) & S_2(s22) & S_3(s32) \\ \text{habitatOf}(e1, s11) & & \text{habitatOf}(e1, s21) \\ \text{habitatOf}(e1, s31) & & \text{habitatOf}(e2, s11) \\ \text{habitatOf}(e2, s12) & & \text{habitatOf}(e2, s21) \\ \text{habitatOf}(e2, s22) & & \text{habitatOf}(e3, s21) \\ \text{habitatOf}(e3, s31) & & \text{habitatOf}(e3, s32) \\ & & \text{imbt}(e2, e3) \\ \text{imbt} \equiv E \times E |_{\exists\text{habitatOf}.S_1, \exists\text{habitatOf}.S_2, \exists\text{habitatOf}.S_3}\end{array}$$

The last of these axioms is expanded as just described. The expanded knowledge base now implies, amongst others, the following statements.

$$\begin{array}{ccc}N_{0,0}(e1) & N_{0,0}(e2) & N_{0,0}(e3) \\ N_{1,1}(e1) & N_{1,1}(e2) & N_{1,0}(e3) \\ N_{2,2}(e1) & N_{2,2}(e2) & N_{2,1}(e3) \\ N_{3,3}(e1) & N_{3,2}(e2) & N_{3,2}(e3)\end{array}$$

Note that these inferences require the closure axioms as described. We furthermore obtain as logical consequences

$$\exists S_3.\text{Self}(e1) \quad \exists S_2.\text{Self}(e2) \quad \exists S_2.\text{Self}(e3)$$

and therefore also

$$\begin{aligned}\text{imbt}(e1, e2) \\ \text{imbt}(e1, e3) \\ R_{\text{typeCountViolation}}(e2, e3)\end{aligned}$$

as expected.

The example just given shows how the pattern can be used. However, in case of ambiguous data, there are also cases in which the pattern may not support the full range of logical consequences. This can happen, for example, if a field scientist is uncertain about species membership of an observed plant or animal. To understand these difficulties, consider the following knowledge base.

$$\begin{aligned}
& E(e1) \quad E(e2) \\
& S_{1,2} \equiv S_1 \sqcup S_2 \\
& \quad S_{1,2}(s) \\
& \quad \text{habitatOf}(e1, s) \\
\text{imbt} \equiv E \times E |_{\exists \text{habitatOf}. S_1, \exists \text{habitatOf}. S_2, \exists \text{habitatOf}. S_3}
\end{aligned}$$

Some logical consequences of this knowledge base are $N_{0,0}(e1)$ $N_{0,0}(e2)$ $N_{1,0}(e2)$ $N_{2,0}(e2)$ $N_{3,0}(e2)$. However, we can infer neither $N_{1,0}(e1)$ nor $N_{1,1}(e1)$ due to the ambiguity of the data, and consequently we do not have a count for the number of species in $e1$ – and thus fail to infer $\text{imbt}(e1, e2)$ while the data would suggest this. The assumptions made in Theorem 1 serve to focus on the basic settings (although, in general, the theorem will carry over to more general settings, provided data ambiguity is avoided).

4.2 Application to Further Use Cases

As a logical pattern, our work is domain independent and can be applied to a variety of modeling problems in the geosciences and beyond. While we focused on the species richness aspect of biodiversity in the previous section, the pattern can also be adapted to model species evenness. It can also be applied to model other diversities as well as *variety* which is one of the key characteristics of data-intensive science. To give another example, one could model neighborhoods in cities based on the different composition of types of Points Of Interest (POI). Following the same argumentation, the pattern could also be expanded to meta-model the dynamics of semantic heterogeneity within knowledge infrastructures such as NSF's Earthcube. Further application areas include modeling the changing relation between feature types and cartographic scale discussed in the introduction.

In general, the pattern can be applied to those definitions that can be translated to binary relationships and require a (total or relative) numerical sizing of different classes (or individuals). In practice, logical pattern are not used alone but will typically be combined with content patterns such as the SSO and referential qualities patterns mentioned before.

5. CONCLUSIONS & OUTLOOK

In this paper, we argued for a logical pattern to quantify over types and explained why this cannot be modeled in any Semantic Web language directly. As a logical pattern, our work is independent of a specific use case and can be applied equally well to geographic features types of Points of Interest as well as to species in studying biodiversity. In combination with content patterns, our work will support the creation of local, purpose-driven ontologies, and, thus, support a staged approach of introducing semantics to knowledge infrastructures such as NSF's EarthCube, instead of hoping for commonly agreed domain ontologies to come. As argued before, a pattern-based approach can foster semantic interoperability without restricting the diversity of interdisciplinary research. We showed how the pattern is formally constructed, made it available to ontology engineers and domain experts as an OWL macro, and specified reasoning strategies. We discussed how the pattern can be applied to model species richness as part of biodiversity and also highlighted other application areas to show generalizability.

Our work also shows that the expressivity argument often brought forward as limiting factor for modeling complex ontologies can be

approached by developing ready-to-use and packaged strategies. Besides continuing the development of geo-ontology design patterns, we also plan to work on methods that assist engineers and domain experts in finding appropriate patterns and understanding which logical and content patterns can be combined. So-called *GeoVoCamps* that bring domain expert and knowledge engineers together to work on concrete, purpose-driven ontologies and semantic technologies, offer the environment to refine existing patterns, identify needs for further patterns, and understand the different viewpoints on geospatial semantics.

Finally, as follow-up work, we intend to refine the pattern to realize a better coverage of cases with ambiguous and uncertain data. We also intend to explore the benefits and drawbacks of using alternative modeling approaches to world closure, such as MKNF description logics [21].

Acknowledgements. This work was partially supported by the National Science Foundation under award 1017225 "III: Small: TROn – Tractable Reasoning with Ontologies." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

6. REFERENCES

- [1] Bizer, C., Heath, T., Berners-Lee, T.: Linked Data – The Story So Far. *International Journal on Semantic Web and Information Systems* 5(3), 1–22 (2009)
- [2] Compton, M., Barnaghi, P., Bermudez, L., Garcia-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., Huang, V., Janowicz, K., Kelsey, W.D., Phuoc, D.L., Lefort, L., Leggieri, M., Neuhaus, H., Nikolov, A., Page, K., Passant, A., Sheth, A., Taylor, K.: The SSN Ontology of the W3C Semantic Sensor Network Incubator Group. *Web Semantics: Science, Services and Agents on the World Wide Web* (2012; in press), <http://www.websemanticsjournal.org/index.php/ps/article/view/292>
- [3] Cruz, I.F., Antonelli, F.P., Stroe, C.: AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proc. VLDB Endow.* 2(2), 1586–1589 (2009)
- [4] David, J., Euzenat, J., Scharffe, F., Trojahn does Santos, C.: The Alignment API 4.0. *Semantic Web* 2(1), 3–10 (2011)
- [5] Fernandez-Lopez, M., Gomez-Perez, A., Juristo, N.: Methontology: from ontological art towards ontological engineering. In: *Proceedings of the AAAI97 Spring Symposium*. pp. 33–40. Stanford, USA (1997)
- [6] Galton, A.: Fields and objects in space, time, and space-time. *Spatial Cognition and Computation* 1, 39–68 (2004)
- [7] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: *Design Patterns: Elements of Resusable Object-Oriented Software*. Addison-Wesley Professional (1995)
- [8] Gangemi, A.: Ontology design patterns for semantic web content. In: Gil, Y., Motta, E., Benjamins, R., Musen, M. (eds.) *The Semantic Web - ISWC 2005*, 4th International Semantic Web Conference. *Lecture Notes in Computer Science*, vol. 3729, pp. 262–276. Springer (2005)
- [9] Gangemi, A., Guarino, N., Masolo, C., Oltramari, A., Schneider, L.: Sweetening Ontologies with DOLCE. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) *Proceedings of the 13th International Conference on Knowledge Engineering*

- and Management, EKAW 2002, Siguenza, Spain, October 1-4, 2002. Lecture Notes in Computer Science, vol. 2473, pp. 223–233. Springer Berlin (2002)
- [10] Grimm, S., Hitzler, P., Abecker, A.: Knowledge Representation and Ontologies. In: Studer, R., Grimm, S., Abecker, A. (eds.) *Semantic Web Services: Concepts, Technology and Applications*, pp. 51–106. Springer, Heidelberg (2007)
- [11] Grimm, S., Hitzler, P.: Semantic Matchmaking of Web Resources with Local Closed-World Reasoning. *International Journal of Electronic Commerce* 12(2), 89–126 (2007)
- [12] Guarino, N.: Formal Ontology and Information Systems. In: Guarino, N. (ed.) *International Conference on Formal Ontology in Information Systems (FOIS1998)*, pp. 3–15. IOS Press, Trento, Italy (1998)
- [13] Hill, M.O.: Diversity and evenness: A unifying notation and its consequences. *Ecology* 54(2), 427–432 (1973)
- [14] Hitzler, P., Janowicz, K., Berg-Cross, G., Obrst, L., Sheth, A., Finin, T., Cruz, I.: Semantic aspects of EarthCube. EarthCube report of the Technology Subcommittee of the EarthCube Semantics and Ontologies Group, May 22, 2012 (2012), available from <http://www.pascal-hitzler.de/>
- [15] Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. (eds.): *OWL 2 Web Ontology Language: Primer*. W3C Recommendation 27 October 2009 (2009), available from <http://www.w3.org/TR/owl2-primer/>
- [16] Hitzler, P., Krötzsch, M., Rudolph, S.: *Foundations of Semantic Web Technologies*. Chapman & Hall/CRC (2009)
- [17] Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible *SR_QIQ*. In: Doherty, P., Mylopoulos, J., Welty, C. (eds.) *Proc. 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*. pp. 57–67. AAAI Press (2006)
- [18] Jain, P., Hitzler, P., Sheth, A.P., Verma, K., Yeh, P.Z.: Ontology alignment for linked open data. In: *Proceedings of the 9th international semantic web conference on The semantic web - Volume Part I*. pp. 402–417. ISWC'10, Springer-Verlag, Berlin, Heidelberg (2010)
- [19] Janowicz, K.: Observation-driven Geo-Ontology Engineering. *Transactions in GIS* 16(3), 351–374 (2012)
- [20] Janowicz, K., Hitzler, P.: The Digital Earth as a knowledge engine. *Semantic Web journal* 3(3), 213–221 (2012)
- [21] Knorr, M., Hitzler, P., Maier, F.: Reconciling OWL and non-monotonic rules for the Semantic Web. In: *Proceedings ECAI2012*. IOS Press (2012), to appear
- [22] Krisnadhi, A., Sengupta, K., Hitzler, P.: Local closed world semantics: Keep it simple, stupid! In: Rosati, R., Rudolph, S., Zakharyashev, M. (eds.) *2011 International Workshop on Description Logics*. *CEUR Workshop Proceedings*, vol. 745. CEUR-WS.org (2011)
- [23] Krisnadhi, A., Maier, F., Hitzler, P.: OWL and Rules. In: Polleres, A., d'Amato, C., Arenas, M., Handschuh, S., Kroner, P., Ossowski, S., Patel-Schneider, P.F. (eds.) *Reasoning Web*. *Lecture Notes in Computer Science*, vol. 6848, pp. 382–415. Springer (2011)
- [24] McMaster, R.B., Shea, K.S.: *Generalization in Digital Cartography*. Assoc. of American Geographers, Washington, D.C. (1992)
- [25] Motik, B., Patel-Schneider, P., Parsia, B. (eds.): *OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax*. W3C Recommendation (27 October 2009), available at <http://www.w3.org/TR/owl2-syntax/>
- [26] Noss, R.F.: Indicators for monitoring biodiversity: A hierarchical approach. *Conservation Biology* 4(4), 355–364 (1990)
- [27] Ortmann, J., Daniel, D.: An ontology design pattern for referential qualities. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) *The Semantic Web – ISWC 2011*, *Lecture Notes in Computer Science*, vol. 7031, pp. 537–552. Springer (2011)
- [28] Raskin, R.G., Pan, M.J.: Knowledge representation in the semantic web for earth and environmental terminology (SWEET). *Comput. Geosci.* 31(9), 1119–1125 (Nov 2005)
- [29] Roth, R., Brewer, C., Stryker, M.: A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives* 68, 29–64 (2011)
- [30] Rudolph, S., Krötzsch, M., Hitzler, P.: All Elephants are Bigger than All Mice. In: Baader, F., Lutz, C., Motik, B. (eds.) *Proceedings of the 21st International Workshop on Description Logics (DL2008)*, Dresden, Germany, May 13-16, 2008. *CEUR Workshop Proceedings*, vol. 353 (2008)
- [31] Sengupta, K., Krisnadhi, A., Hitzler, P.: Local closed world reasoning: Grounded circumscription for OWL. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N.F., Blomqvist, E. (eds.) *The Semantic Web – ISWC 2011 – 10th International Semantic Web Conference*, Bonn, Germany, October 23-27, 2011, *Proceedings, Part I*. *Lecture Notes in Computer Science*, vol. 7031, pp. 617–632. Springer, Heidelberg (2011)
- [32] Sinha, G., Mark, D.: Toward a foundational ontology of the landscape. In: *Extended Abstracts of GIScience 2010* (2010)
- [33] SPARQL query language for RDF. W3C Recommendation 15 January 2008, <http://www.w3.org/TR/rdf-sparql-query/> (2008)
- [34] Winston, M.E., Chaffin, R., Herrmann, D.: A taxonomy of part-whole relations. *Cognitive Science* 11(4), 417–444 (1987)
- [35] World Resources Institute, World Conservation Union, and United Nations Environment Programme. *Global Biodiversity Strategy*. (1992)

APPENDIX

THEOREM 1. *Let K be a $SR_{Q}IQ$ knowledge base (not containing concepts C_1, \dots, C_n or any sub-formulas of these concepts in the $TBox$), and let K' be the knowledge base which results from adding the type-count comparison pattern*

$$R \equiv \top \times \top |_{C_1, \dots, C_n}$$

to K . Furthermore, for any individual a in K let

$$c(a) = |\{i \mid K \models C_i(a)\}|,$$

i.e., $c(a)$ is the number of different classes C_i of which a can be inferred to be an element.

(i) Then $c(a) > c(b)$ implies $K' \models R(a, b)$.

Now assume furthermore that there is no pair (a, b) of individuals such that $K \models R_{\text{typeCountViolation}}(a, b)$.

(ii) Then $K' \models R(a, b)$ also implies $c(a) > c(b)$.

PROOF. We substitute $R \equiv \top \times \top|_{C_1, \dots, C_n}$ by the set of *SROTQ* axioms defined by the transformation in Table 2. We start by proving statement (i) of the theorem, i.e., that $c(a) > c(b)$ implies $K' \models R(a, b)$.

By axiom (1) we have that all concepts C_i , where $1 \leq i \leq n$, are closed. Consequently, we have that either $K \models C_i(a)$ or $K \models \neg C_i(a)$, where $1 \leq i \leq n$, for every individual a . We can assert this due to the fact that the concepts C_i and their sub formulas do not appear in the Tbox – and that the C_i are also closed.

By axiom (2) we have that if $K' \models R(a, b)$, then $K' \models \top(a)$ and $K' \models \top(b)$ also hold. The choice of the \top concept simplifies the definition and the theorem. In case we want to quantify over an arbitrary concept D , it is easy to see that the proof carries on easily just by substituting the appearances of the \top concept in the proof by the class D . That is the reason for the appearance of the \top concept in the proof.

Axiom (3) makes the class \top a subclass of $N_{0,0}$. By axioms (3)-(5), we have that $K \models \top(e)$ and for every $1 \leq j \leq n$ there is a unique $x \leq j$ such that $K \models N_{j,x}(e)$. Due to axioms (4) and (5) we have that $K' \models N_{x,y}(e)$ if and only if e is also contained in y different classes C_i , where $i \leq x$.

Therefore, we have that for all $K' \models \top(e)$ also $K' \models N_{n,y}(e)$, where the y marks the number of different classes C_i s.t. $K \models C_i(e)$ where $1 \leq i \leq n$. By axiom (6) we have that if $K \models N_{n,y}(e)$ then $K' \models S_y(e, e)$. Axiom (8) enforces the creation of the R role according to the given definition in the theorem making use of the roles S_i . Therefore we have that $R(a, b)$ is an inference of K' for all pairs of individuals (a, b) s.t. $c(a) > c(b)$.

Note that, as part of the definition, we make the assumption that there is no $R_{\text{typeCountViolation}}(x, y)$ entailed by K' . We make use of this assumption to prove part (ii) of the theorem. If $K' \models R_{\text{typeCountViolation}}(x, y)$ we have that there is no pair of individuals such that $K' \models (S_i \circ R \circ S_j)(x, y)$ for all $i \leq j$, where $1 \leq j \leq n$, by axiom (7).

As shown in the previous part of the proof we have that roles S_i mark the number of different classes C_i a given individual is in. Combined with the fact $K \not\models (S_i \circ R \circ S_j)(x, y)$, for all $i \leq j$, where $1 \leq j \leq n$, we can say that there is no $R(a, b)$ entailed by K' where $c(a) \leq c(b)$. \square