Using Preference Order in Ontologies

S. Heymans D. Vermeir

Dept. of Computer Science
Free University of Brussels, VUB
{*sheymans,dvermeir*}*@vub.ac.be*

Abstract

The latest ontology languages can be translated into a description logic (DL), thus providing them with a formal semantics and associated reasoning procedures. We introduce the ordered description logic $\mathcal{OSHOQ}(\mathbf{D})$ as a simple decidable extension of $\mathcal{SHOQ}(\mathbf{D})$ that supports the direct definition of a preference order on defeasible axioms, thus allowing for a succinct and intuitive expression of defeasible ontologies, containing e.g. exceptions for certain axioms.

We demonstrate the usefulness of $\mathcal{OSHOQ}(\mathbf{D})$ for solving inconsistencies that may appear e.g. when merging existing ontologies. We present an algorithm that, based on concrete examples of facts that should be true, produces minimal preference orderings on the axioms, in order to make an otherwise inconsistent knowledge base consistent.

```
ontology-definitions
  slot-def eats
  class-def animal
  class-def plant
    subclass-of NOT animal
  class-def defined carnivore
    subclass-of animal
    slot-constraint eats
      value-type animal
  class-def defined herbivore
    subclass-of animal
    slot-constraint eats
      value-type plant
  class-def herbivore
    subclass-of NOT carnivore
  covered-by(one-of dog) carnivore
  covered-by(one-of grass) plant
```

Figure 1. An example ontology

1. Introduction

The "Semantic Web" [5] will improve on the current Internet by providing facilities for the sharing, interpretation and processing of information, not only between humans but also between software agents. Ontologies play an important role in the building of this new Web. Their main purpose is to bring a *shared understanding* [15] to the web and, by this understanding, allow a more intelligent approach to information exchange.

In order to describe ontologies, one needs ontology languages, such as DAML+OIL or OIL [4, 10]. Figure 1 shows a fragment of an example ontology [10], expressed in the language (Standard) OIL [9].

The example describes a number of concepts and their relationships: animals that only eat other animals are *carnivores* while *herbivores* only eat plants (which themselves are not animals). My dog is put forward as a carnivore while grass is a plant. The resulting ontology is easily seen to be consistent since there are no apparent contradictions. More formally, we say that the corresponding description logic knowledge base is consistent, see later. Imagine, however, that you add the following axiom: **covered-by**(**one-of** dog)(**slot-constraint** *eats* **has-value** (**one-of** grass)).

The new axiom says that your dog (sometimes) eats grass (this corresponds to reality: dogs that feel sick often eat grass). Informally, this makes the ontology inconsistent because your dog is supposed to be a carnivore and carnivores only eat animals, whereas grass is a plant and plants are not animals. Thus the new ontology does not support our intuition that "this dog is a carnivore, although, from time to time, it eats grass".

We can reconcile the example ontology with our intuition by *defeating* the rule saying that carnivores only eat animals with the rule that this dog also eats plants. Informally, this means that one is willing to accept that the carnivore rule is not absolute and may be ignored if one sees a carnivore eating grass. Of course, we will need to formalize this notion of defeat by extending the underlying description logic.

A description logic can be used to express the formal semantics of an ontology written in an ontology language like OIL, but it also provides some basic reasoning services such as satisfiability checking and related algorithms [12]. The correspondence between OIL and the DL $\mathcal{SHIQ}$ has been shown in [9]. As explained in [11] this mapping is incomplete with respect to *concrete data types* and *named individuals*, two features that are present in current ontology languages. A DL that overcomes these two deficiencies is $\mathcal{SHOQ}(\mathbf{D})$[11], which includes support for data types ($\mathbf{D}$) and named individuals ($\mathcal{O}$, see also [14] for reasoning with individuals).

In this paper, we extend $\mathcal{SHOQ}(\mathbf{D})$ to include a preference relation on axioms. This relation can be used as a justification for defeating certain axioms with more preferred ones, thus capturing "subtleties" like the grass-eating dog example in OIL. A similar approach has been proposed for e.g. ordered logic programming in [7, 6]. Besides being often more intuitive, allowing an explicit preference relation between axioms also has the advantage that it can be derived algorithmically, in contrast with the default logic approach [2, 3] where default rules have to be manufactured virtually by hand.

We provide such a learning algorithm that proposes a preference order on the set of axioms in a knowledge base, thus reconciling input example facts with existing axioms. Because we assume that reasoning with ontologies is done through a description logic, this also provides an ontology learning mechanism.

The remainder of this paper is organized as follows: in Section 2 we extend $\mathcal{SHOQ}(\mathbf{D})$ to ordered $\mathcal{SHOQ}(\mathbf{D})$ (denoted $\mathcal{OSHOQ}(\mathbf{D})$) by providing an extra component (an *Obox*) that defines an order on distinguished sets of defeasible (both terminological and role) axioms. The notion of defeat between axioms is used to define a semantics for an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base (KB) that respects the order in the Obox. An algorithm that learns the order among the axioms from examples provided by the designer is presented in Section 3, where we argue that this may be a useful tool for synthesizing new ontologies from existing (but overlapping) ones. Finally, Section 4 contains conclusions and directions for further research.

2. Basic Definitions

We summarize the syntax and semantics of $\mathcal{SHOQ}(\mathbf{D})$ as in [11]. We assume that we have a set of data types $\mathbf{D}$ and associate with each $d \in \mathbf{D}$ a set $d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$, where $\Delta_{\mathbf{D}}$ is the domain of all data types (the *concrete domain*, see [1]).

Let $\mathbf{C}$ be the set of *concept names*, $\mathbf{R}$ the disjoint union of *abstract role names* $\mathbf{R}_A$ and *concrete role names* $\mathbf{R}_{\mathbf{D}}$. A *role box* $\mathcal{R}$ is a finite set of *role axioms* $R \sqsubseteq S$ where $R, S \in \mathbf{R}_A$ or $R, S \in \mathbf{R}_{\mathbf{D}}$ and *transitivity axioms* $\mathsf{Trans}(R)$ for $R \in \mathbf{R}_A$. An abstract role R is called *transitive* if $\mathsf{Trans}(R) \in \mathcal{R}$. A *simple role* R for a role box $\mathcal{R}$ is a role that is not transitive nor does it have any transitive subroles. Let $\mathbf{I}$ be a set of *individual names*. $\mathbf{C}$, $\mathbf{R}$ and $\mathbf{I}$ are mutually disjoint. The set of $\mathcal{SHOQ}(\mathbf{D})$-*concept expressions* is defined such that every concept name $A \in \mathbf{C}$ is a concept expression and for every $o \in \mathbf{I}$, $\{o\}$ is a concept expression. Moreover, for C and D concept expressions, $R \in \mathbf{R}_A$, $T \in \mathbf{R}_{\mathbf{D}}$, S a simple role and $d \in \mathbf{D}$, the constructors in Table 1 can be used to form complex concept expressions.

Table 1. Syntax and semantics of $\mathcal{SHOQ}(\mathbf{D})$-concept expressions

syntax	semantics
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
T	$T^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$
$\{o\}$	$\{o\}^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}, \#\{o\}^{\mathcal{I}} = 1$
d	$d^{\mathbf{D}} \subseteq \Delta_{\mathbf{D}}$
$\neg d$	$(\neg d)^{\mathbf{D}} = \Delta_{\mathbf{D}} \setminus d^{\mathbf{D}}$
$C \sqcap D$	$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$C \sqcup D$	$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\neg C$	$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
$\exists R.C$	$(\exists R.C)^{\mathcal{I}} = \{x \mid \exists y : (x,y) \in R^{\mathcal{I}}, y \in C^{\mathcal{I}}\}$
$\forall R.C$	$(\forall R.C)^{\mathcal{I}} = \{x \mid \forall y : (x,y) \in R^{\mathcal{I}} \Rightarrow y \in C^{\mathcal{I}}\}$
$\geq nS.C$	$(\geq nS.C)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in S^{\mathcal{I}}, y \in C^{\mathcal{I}}\} \geq n\}$
$\leq nS.C$	$(\leq nS.C)^{\mathcal{I}} = \{x \mid \#\{y \mid (x,y) \in S^{\mathcal{I}}, y \in C^{\mathcal{I}}\} \leq n\}$
$\exists T.d$	$(\exists T.d)^{\mathcal{I}} = \{x \mid \exists y : (x,y) \in T^{\mathcal{I}}, y \in d^{\mathbf{D}}\}$
$\forall T.d$	$(\forall T.d)^{\mathcal{I}} = \{x \mid \forall y : (x,y) \in T^{\mathcal{I}} \Rightarrow y \in d^{\mathbf{D}}\}$

The semantics of $\mathcal{SHOQ}(\mathbf{D})$ is defined using an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ is a nonempty domain (the *abstract domain*) and $\cdot^{\mathcal{I}}$ is an interpretation function, defined on concept expressions and roles as in Table 1.

A *Tbox* $\mathcal{T}$ is a finite set of *terminological axioms* $C \sqsubseteq D$ with C and D $\mathcal{SHOQ}(\mathbf{D})$-concept expressions. An interpretation $\mathcal{I}$ *satisfies* a role axiom $R_1 \sqsubseteq R_2$, in a role box $\mathcal{R}$ iff $R_1^{\mathcal{I}} \subseteq R_2^{\mathcal{I}}$, and it satisfies a transitivity axiom $\mathsf{Trans}(R) \in \mathcal{R}$ iff $R^{\mathcal{I}} = (R^{\mathcal{I}})^+$, where $(\cdot)^+$ is the transitive closure operator. An interpretation $\mathcal{I}$ satisfies a role box $\mathcal{R}$ iff it satisfies every role and transitivity axiom in $\mathcal{R}$. An interpretation $\mathcal{I}$ satisfies a terminological axiom $C_1 \sqsubseteq C_2$ from a Tbox iff $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$. $\mathcal{I}$ satisfies a Tbox $\mathcal{T}$ iff $\mathcal{I}$ satisfies every axiom in $\mathcal{T}$. An interpretation $\mathcal{I}$ that satisfies $\mathcal{T}$ and $\mathcal{R}$ is said to be a *model* for $\mathcal{T}$ and $\mathcal{R}$. A $\mathcal{SHOQ}(\mathbf{D})$-concept expression C is *satisfiable* w.r.t a Tbox $\mathcal{T}$ and a role box $\mathcal{R}$ if there exists a model $\mathcal{I}$ for $\mathcal{T}$ and $\mathcal{R}$ such that $C^{\mathcal{I}} \neq \emptyset$. A concept expression C is *subsumed* by a concept expression D (notation: $C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each model $\mathcal{I}$ for $\mathcal{T}$ and $\mathcal{R}$.

Translating, as in [9], the example in the introduction to $\mathcal{SHOQ}(\mathbf{D})$ we get a $\mathcal{SHOQ}(\mathbf{D})$-knowledge base $\langle \mathbf{C}, \mathbf{R}, \mathbf{D} = \emptyset, \mathbf{I}, \mathcal{R} = \emptyset, \mathcal{T}\rangle$ with concept names $\mathbf{C} = \{\mathsf{animal}, \mathsf{plant}, \mathsf{carnivore}, \mathsf{herbivore}\}$, $\mathbf{R} = \{\mathit{eats}\}$, $\mathbf{I} = \{\mathit{dog}, \mathit{grass}\}$ and the Tbox $\mathcal{T}$ contains the following axioms (numbered for easy reference):

		$\mathcal{T}$	
(1)	plant	$\sqsubseteq$	$\neg$ animal
(2)	herbivore	$\sqsubseteq$	$\neg$ carnivore
(3)	$\{$dog$\}$	$\sqsubseteq$	carnivore
(4)	$\{$grass$\}$	$\sqsubseteq$	plant
(5)	carnivore	$\sqsubseteq$	animal $\sqcap \forall eats.$animal
(6)	herbivore	$\sqsubseteq$	animal $\sqcap \forall eats.$plant
(7)	animal $\sqcap \forall eats.$animal	$\sqsubseteq$	carnivore
(8)	animal $\sqcap \forall eats.$plant	$\sqsubseteq$	herbivore
(9)	$\{$dog$\}$	$\sqsubseteq$	$\exists eats.\{$grass$\}$

Take an interpretation $\mathcal{I}$ with domain $\Delta^{\mathcal{I}} = \{d, g, c\}$, and $\mathsf{animal}^{\mathcal{I}} = \{d, c\}$, $\mathsf{carnivore}^{\mathcal{I}} = \{d\}$, $\mathsf{herbivore} = \{c\}$, $\mathsf{plant}^{\mathcal{I}} = \{g\}$, $\{\mathit{dog}\}^{\mathcal{I}} = \{d\}$, $\{\mathit{grass}\}^{\mathcal{I}} = \{g\}$ and $\mathit{eats}^{\mathcal{I}} = \{(d, g), (d, c), (c, g)\}$. Although $\mathcal{I}$ seems reasonable to our human eyes, it does not satisfy the Tbox, because it falsifies axiom (5) since our *dog* sometimes eats grass, contradicting that it is a $\mathsf{carnivore}$ ($\{d\} \not\subseteq \emptyset$).

Thus, we can keep $\mathcal{I}$ as a model, and support our intuition, if we defeat axiom (5) using another "more preferred" axiom. We could e.g. declare that $(3) < (5)$, corresponding to our belief that having been declared a carnivore is sufficient for our dog to be a carnivore, even if he occasionally eats grass. Intuitively, using (3) as a defeater for (5), an object can cause (5) to fail, provided it satisfies (3). In our case, d makes (5) fail, but $\{\mathit{dog}\}^{\mathcal{I}} = \{d\} \subseteq \mathsf{carnivore}^{\mathcal{I}}$, and thus d satisfies (3).

In general, the preference relation between axioms must be provided by the knowledge base designer, depending on the underlying intuition that she wishes to support. However, as will be shown in Section 3, one can "learn" an appropriate order by supplying examples of individuals that must be instances of concepts or play certain roles.

Formally, we define the defeasible logic $\mathcal{OSHOQ}(\mathbf{D})$ as an extension of $\mathcal{SHOQ}(\mathbf{D})$ by adding a preference order between axioms.

Definition 1 *An $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base is a tuple*[1] *$\langle \mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ where $\langle \mathcal{T}, \mathcal{R}\rangle$ is a $\mathcal{SHOQ}(\mathbf{D})$-knowledge base, and $\mathcal{O}$ is an* ***Obox*** *representing a preference order between axioms. Thus $\mathcal{O}$ contains items of the form $C_1 \sqsubseteq C_2 < C_3 \sqsubseteq C_4$ and $R_1 \sqsubseteq R_2 < R_3 \sqsubseteq R_4$ with $C_1 \sqsubseteq C_2$ and $C_3 \sqsubseteq C_4$ in $\mathcal{T}$, $R_1 \sqsubseteq R_2$ and $R_3 \sqsubseteq R_4$ in $\mathcal{R}$. Moreover, R_1, R_2, R_3, R_4 are all abstract or all concrete and R_3 must be simple. For a pair $a_1 < a_2$ in $\mathcal{O}$, a_2 is said to be* ***defeasible*** *while a_1 is a (possible)* ***defeater*** *of a_2.*

[1]For the sake of brevity, we omit the concept names $\mathbf{C}$, role names $\mathbf{R}$, datatypes $\mathbf{D}$ and individual names $\mathbf{I}$ from the notation.

The notion of defeat is formalized in the following definition.

Definition 2 *Let $\Sigma = \langle \mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base. An* ***interpretation*** *of Σ is any interpretation $\mathcal{I}$ of the $\mathcal{SHOQ}(\mathbf{D})$-knowledge base $\langle \mathcal{T}, \mathcal{R}\rangle$. A terminological axiom $A \sqsubseteq B \in \mathcal{T}$ is* ***applicable*** w.r.t. *$x \in \Delta^{\mathcal{I}}$ iff $x \in A^{\mathcal{I}}$. A role axiom $R_1 \sqsubseteq R_2 \in \mathcal{R}$ is* ***applicable*** w.r.t. *$(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ (or $(x, y) \in \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$, if $R_1, R_2 \in \mathbf{R}_{\mathbf{D}}$) iff $(x, y) \in R_1^{\mathcal{I}}$.*

$A \sqsubseteq B \in \mathcal{T}$ is ***applied*** w.r.t. *$x \in \Delta^{\mathcal{I}}$ iff $A \sqsubseteq B$ is applicable w.r.t. x and $x \in B^{\mathcal{I}}$. $R_1 \sqsubseteq R_2 \in \mathcal{R}$ is* ***applied*** *w.r.t. $(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ (or $(x, y) \in \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$, if $R_1, R_2 \in \mathbf{R}_{\mathbf{D}}$) iff $R_1 \sqsubseteq R_2$ is applicable w.r.t. (x, y) and $(x, y) \in R_2^{\mathcal{I}}$.*

$A \sqsubseteq B \in \mathcal{T}$ is ***defeated*** w.r.t. *$x \in \Delta^{\mathcal{I}}$ and the Obox $\mathcal{O}$ iff $\exists(C \sqsubseteq D < A \sqsubseteq B) \in \mathcal{O}$ such that $C \sqsubseteq D$ is applied w.r.t. x. $R_1 \sqsubseteq R_2 \in \mathcal{R}$ is* ***defeated*** w.r.t. *$(x, y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ (or $(x, y) \in \Delta^{\mathcal{I}} \times \Delta_{\mathbf{D}}$, if $R_1, R_2 \in \mathbf{R}_{\mathbf{D}}$) and the Obox $\mathcal{O}$ iff $\exists(R_3 \sqsubseteq R_4 < R_1 \sqsubseteq R_2) \in \mathcal{O}$ such that $R_3 \sqsubseteq R_4$ is applied* w.r.t. *(x, y).*

We adjust the definition of satisfaction to take into account the preference order in the Obox.

Definition 3 *Let $\Sigma = \langle \mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base. An interpretation $\mathcal{I}$ of Σ* ***satisfies*** *an axiom ψ from $\mathcal{T}$ (resp. $\mathcal{R}$) if for each x (resp. (x, y)) for which ψ is applicable, ψ is either applied or defeated.*

A ***model*** *of Σ is any interpretation that satisfies all of the axioms in $\mathcal{T} \cup \mathcal{R}$.*

The definitions of concept satisfiability and subsumption are then straightforward and basically the same as for $\mathcal{SHOQ}(\mathbf{D})$. We call Σ *consistent* iff there exists a model $\mathcal{I}$ of Σ.

Note that while allowing the direct expression of preference between axioms is intuitive, the same effect can be achieved in first order logic. The $\mathcal{SHOQ}(\mathbf{D})$ tableau reasoning algorithm [11] can be extended to handle the preference relation, leading to the following theorem.

Theorem 1 *Satisfiability checking, consistency and subsumption checking for $\mathcal{OSHOQ}(\mathbf{D})$ are decidable problems.*

3. Applications

3.1. Default Reasoning

Representing preferences in an Obox provides a succinct and modular mechanism to support the representation of defaults and exceptions: a default can be simply formulated as a defeasible axiom while exceptions can be represented independently by adding appropriate items to the Obox. E.g.

we can start with a general description of birds and their properties: $\langle\{B \sqsubseteq F\}, \emptyset, \emptyset\rangle$ where $B \sqsubseteq F$ asserts that "birds tend to fly". Later on, we can add specific kinds of birds, e.g. sparrows (S) that inherit the default flying property: in $\langle\{S \sqsubseteq B, B \sqsubseteq F\}, \emptyset, \emptyset\rangle$, S is subsumed by F. Exceptions are accommodated by simply adding the relevant information as a new axiom that is preferred over the default, which itself need not be touched: in $\langle\{S \sqsubseteq B, P \sqsubseteq B, P \sqsubseteq \neg F, B \sqsubseteq F\}, \emptyset, \{P \sqsubseteq \neg F < B \sqsubseteq F\}\rangle$, penguins ($P$) are non-flying birds.

3.2. Synthesis of Ontologies

Often, new ontologies are constructed starting from (a combination of) existing ontologies, adding refinements that correspond to specialized knowledge. Both integration of ontologies and ontology refinement may lead to inconsistencies. E.g. consider the following fragment of an ontology about different religions (a variation of the well-known "Nixon diamond"): $\Sigma_1 = \langle\{Q \sqsubseteq P\}, \emptyset, \emptyset\rangle$ which asserts that "quakers (Q) tend to be pacifists (P)". Another "political" ontology $\Sigma_2 = \langle\{\{Nixon\} \sqsubseteq R, R \sqsubseteq \neg P\}, \emptyset, \emptyset\rangle$ states that "republicans (R) tend not to be pacifists" and that $Nixon$ is a republican. When joining the two ontologies and adding the specialized knowledge that Nixon is a quaker, one obtains $\langle\{\{Nixon\} \sqsubseteq R, \{Nixon\} \sqsubseteq Q, R \sqsubseteq \neg P, Q \sqsubseteq P\}, \emptyset, \emptyset\rangle$ which is inconsistent. The designer can then incorporate his specialized knowledge by simply adding $\{R \sqsubseteq \neg P < Q \sqsubseteq P\}$ to the Obox, yielding a consistent $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base.

In general, the appropriate Obox can be learned from examples and inconsistent knowledge bases may be made consistent by simply adding appropriate items to the Obox, as is implied by the following theorem which asserts that extending the Obox is monotonic w.r.t. the semantics.

Definition 4 *Let $\Sigma = \langle\mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base. An* ***order-extension*** *of Σ is any knowledge base $\langle\mathcal{T}, \mathcal{R}, \mathcal{O}'\rangle$ where $\mathcal{O}' \supseteq O$.*

Theorem 2 *Let Σ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base. All models of Σ are models of any order-extension of Σ.*

We propose an Obox-learning algorithm for extending the Obox of an inconsistent knowledge base $\langle\mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ to a consistent version $\langle\mathcal{T}, \mathcal{R}, \mathcal{O}'\rangle$ where $\mathcal{O} \subseteq \mathcal{O}'$. The algorithm is based on the "candidate elimination algorithm" from [13]. It operates on a *hypothesis space H* that contains all possible Oboxes, i.e., identifying $A \sqsubseteq B < C \sqsubseteq D$ with $(A \sqsubseteq B, C \sqsubseteq D)$ and $R_3 \sqsubseteq R_4 < R_1 \sqsubseteq R_2$ with $(R_3 \sqsubseteq R_4, R_1 \sqsubseteq R_2)$, subsets of $U = ((\mathcal{T} \cup \mathcal{T}^d) \times \mathcal{T}^d) \cup ((\mathcal{R} \cup \mathcal{R}^d) \times \mathcal{R}^d)$, partially ordered by the *generalization partial order* $\subseteq$. Here $\mathcal{T}^d \subseteq \mathcal{T}$ and $\mathcal{R}^d \subseteq \mathcal{R}$ are those sets of axioms for which one is willing to accept defeaters. The *training set* contains axioms $E = \{\{a_1\} \sqsubseteq K_1, \ldots, \{a_n\} \sqsubseteq K_n\}$ where each $\{a_i\} \sqsubseteq K_i$, a_i an individual and K_i a concept expression, represents an example piece of knowledge that must be satisfied by the resulting ontology, i.e. $\langle\mathcal{T} \cup E, \mathcal{R}, \mathcal{O}'\rangle$ must be consistent.

Table 2. "candidate elimination" algorithm

```
1. Start with $S = \{\mathcal{O}_0\}$, where $\Sigma = \langle\mathcal{T}_0 = \mathcal{T}, \mathcal{R}, \mathcal{O}_0\rangle$ is the
   original KB, and examples $E = \{\{a_1\} \sqsubseteq K_1, \ldots, \{a_n\} \sqsubseteq K_n\}$, $i = 1$.
2. Consider an example $\{a_i\} \sqsubseteq K_i$ from $E$.
3. For each $\mathcal{O} \in S$ such that $\langle\mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}, \mathcal{R}, \mathcal{O}\rangle$ is
   not consistent
     (a) Remove $\mathcal{O}$ from $S$.
     (b) Add to $S$ all generalizations $\mathcal{O}' \supset \mathcal{O}$ ($\mathcal{O}'$ formed with
         axioms from $\Sigma$) of $\mathcal{O}$ such that
           i. $\langle\mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}, \mathcal{R}, \mathcal{O}'\rangle$ is consistent, and
          ii. $\mathcal{O}'$ is minimal, i.e. $\forall \mathcal{O}'', \mathcal{O} \subset \mathcal{O}'' \subset \mathcal{O}' \cdot \langle\mathcal{T}_{i-1} \cup$
              $\{\{a_i\} \sqsubseteq K_i\}, \mathcal{R}, \mathcal{O}''\rangle$ is not consistent.
4. $\mathcal{T}_i = \mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}$; $i \leftarrow i + 1$
5. Continue from 2. until either $S = \emptyset$, in which case the algo-
   rithm fails, or all examples in $E$ have been considered and
   $S \neq \emptyset$. In the latter case, the algorithm succeeds and the
   learned Oboxes are in $S$.
```

In [13] the algorithm is initialized with a lower bound S (of specialized items) and an upper bound G (of generalized items). These collections form the limits in between which all the solutions must lie. Here, $S = \{\mathcal{O}\}$ contains the original Obox, and $G = \{U\}$. In the original algorithm (see [13]) there are positive and negative examples. Since we only deal with positive examples, we obtain the algorithm in Table 2.

Theorem 3 *Let $\Sigma = \langle\mathcal{T}, \mathcal{R}, \mathcal{O}\rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$ KB and let $E = \{\{a_1\} \sqsubseteq K_1, \ldots, \{a_n\} \sqsubseteq K_n\}$, be a set of examples. If the algorithm from Table 2 succeeds with non-empty solution set S, then $\mathcal{O}' \in S$ iff $\langle\mathcal{T}, \mathcal{R}, \mathcal{O}'\rangle$ is a minimal order-extension of Σ such that $\langle\mathcal{T} \cup E, \mathcal{R}, \mathcal{O}'\rangle$ is consistent.*

We illustrate the algorithm with a small extension of the previous example, for which the original combined ontology is shown in Figure 2. In this ontology republicans tend to be football fans (F), while pacifists tend to be anti-military (A). Furthermore, football fans tend not to be anti-military.

Rewriting this as an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base, it becomes $\langle\{R \sqsubseteq \neg P \sqcap F, Q \sqsubseteq P, P \sqsubseteq A, F \sqsubseteq \neg A\}, \emptyset, \emptyset\rangle$. We assume we have the following set of examples $E = \{\{Nixon\} \sqsubseteq R, \{Nixon\} \sqsubseteq Q, \{Nixon\} \sqsubseteq \neg A, \{Nixon\} \sqsubseteq \neg P\}$, which corresponds to the knowledge we have about Nixon, i.e. a military-minded non-

```
ontology-definitions
  class-def pacifists
  class-def football fans
  class-def republicans
    subclass-of ((NOT pacifists) AND football fans)
  class-def quakers
    subclass-of pacifists
  class-def pacifists
    subclass-of anti-military
  class-def football fans
    subclass-of NOT anti-military
```

Figure 2. Nixon's anti-militarism

pacifist republican quaker. We pick-up the learning algorithm after have seen these four examples. Our result set S of learned Oboxes then becomes $S = \{\{R \sqsubseteq \neg P \sqcap F < Q \sqsubseteq P\}, \{F \sqsubseteq \neg A < Q \sqsubseteq P\}\}$.

If one wishes to constrain this set of possible Oboxes, one may consider more examples. Note however that additionally adding an example $\{Nixon\} \sqsubseteq \neg F$ would yield an empty S and we would be unable to make our knowledge base consistent, simply because the rule $R \sqsubseteq \neg P \sqcap F$ would have no possible defeaters that effectively defeat the rule.

4. Conclusion and Directions for Further Research

We believe that $\mathcal{OSHOQ}(\mathbf{D})$ may be further exploited as a tool for knowledge base integration and refinement. On a theoretical level, the algorithm of Table 2 may come up with too many solutions that, while theoretically optimal, are not the most natural for fixing the problem at hand. A more "focused" algorithm may incorporate the order generation into a tableau procedure for checking consistency itself, adding required order items when an inconsistency is detected. In addition, the algorithm may be further refined by taking into account meta-properties of concepts and roles, such as the ones described in [8], which may suggest or forbid certain axioms to be defeasible. Experiments with a planned implementation of $\mathcal{OSHOQ}(\mathbf{D})$ will verify the practical feasibility of the approach with non-trivial examples.

The ontology integration problem is much more complex and layered than our simple Nixon-example suggests. Nevertheless, we believe that also in more complicated integration problems, a key issue regarding the Semantic Web, defeasible ontologies will prove useful.

Acknowledgments

Thanks to R. Meersman for stimulating discussion regarding the use of ontologies.

References

[1] F. Baader and P. Hanschke. A scheme for integrating concrete domains into concept languages. Technical Report RR-91-10, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 1991.

[2] F. Baader and B. Hollunder. Embedding defaults into terminological knowledge representation formalisms. In B. Nebel, C. Rich, and W. Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 306–317, San Mateo, California, 1992. Morgan Kaufmann.

[3] F. Baader and B. Hollunder. How to prefer more specific defaults in terminological default logic. In R. Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 669–674, San Mateo, California, 1993. Morgan Kaufmann.

[4] S. Bechhofer, C. Goble, and I. Horrocks. DAML+OIL is not enough. In *Proceedings of the First Semantic Web Working Symposium (SWWS'01)*, pages 151–159. CEUR, 2001.

[5] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, pages 34–43, May 2001.

[6] D. Gabbay, E. Laenens, and D. Vermeir. Credulous vs. sceptical semantics for ordered logic programs. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, pages 208–217. Morgan Kaufmann, 1991.

[7] P. Geerts and D. Vermeir. Defeasible logics. In D. M. Gabbay and P. Smets, editors, *Handbook of defeasible reasoning and uncertainty management systems*, volume 2, pages 175–210. Kluwer Academic Press, 1998.

[8] N. Guarino and C. Welty. Evaluating Ontological Decisions with Ontoclean. *CACM*, 45, 2002.

[9] I. Horrocks. A denotational semantics for Standard OIL and Instance OIL. `http://www.ontoknowledge.org/oil/downl/semantics.pdf`, 2000.

[10] I. Horrocks, D. Fensel, J. Boekstra, S. Decker, M. Erdmann, C. Goble, F. V. Harmelen, M. Klein, S. Staab, R. Studer, and E. Motta. The ontology inference layer OIL. `http://www.cs.vu.nl/~dieter/oil/Tr/oil.pdf`, 2000.

[11] I. Horrocks and U. Sattler. Ontology reasoning in the $\mathcal{SHOQ}(\mathbf{D})$ description logic. In B. Nebel, editor, *Proc. of the 17th Int. Joint Conf. on Artificial Intelligence (IJCAI-01)*, pages 199–204. Morgan Kaufmann, 2001.

[12] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. Ganzinger, D. McAllester, and A. Voronkov, editors, *Proceedings of the 6th International Conference on Logic for Programming and Automated Reasoning (LPAR'99)*, number 1705, pages 161–180. Springer-Verlag, 1999.

[13] T. Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[14] A. Schaerf. Reasoning with individuals in concept languages. *Data Knowledge Engineering*, 13(2):141–176, 1994.

[15] M. Uschold and M. Grüninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.