A Defeasible Ontology Language

S. Heymans and D. Vermeir

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

Abstract. We extend the description logic $\mathcal{SHOQ}(\mathbf{D})$ with a preference order on the axioms. With this strict partial order certain axioms can be overruled, if defeated with more preferred ones. Furthermore, we impose a preferred model semantics, thus effectively introducing nonmonotonicity into $\mathcal{SHOQ}(\mathbf{D})$. Since a description logic can be viewed as an ontology language, or a proper translation of one, we obtain a defeasible ontology language. Finally, we argue that such a defeasible language may be usefully applied for learning and integrating ontologies.

1 Introduction

The "Semantic Web" [BLHL01] seeks to improve on the current World Wide Web, making knowledge not only viewable and interpretable by humans, but also by software agents. Ontologies play a crucial role in the realization of this next generation web, by providing a "shared understanding" [UG96] of certain domains.

In order to describe ontologies, one needs ontology languages, such as DAML+OIL or OIL [BGH01,FHvH+00,FvHH+01]. For example the OIL language is built on three roots [HFB+00]:

- frame-based systems provide the basic modeling primitives: frames (classes) with attributes;
- by mapping the language to a suitable description logic (DL), one obtains a precise semantics and associated inference procedures; and
- the concrete syntax is based on web languages such as XML and RDF [LS99,DvHB+00].

A description logic is 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 checking whether an instance is of a certain type, whether classes are subclasses of other classes, ... [BS00,HST99].

In particular the DL $\mathcal{SHIQ}$ corresponds to the ontology language OIL [Hor00]. As explained in [HS01] this mapping is incomplete with respect to *concrete datatypes* and *named individuals*, two features that are present in current ontology languages. A DL that overcomes these two deficiencies is $\mathcal{SHOQ}(\mathbf{D})$[HS01], which includes support for datatypes ($\mathbf{D}$) and named individuals ($\mathcal{O}$, see also [Sch94] and [HST00] for reasoning with individuals).

In this paper we further extend the DL $\mathcal{SHOQ}(\mathbf{D})$ with a preference order, as in [HV02]. This order indicates whether a certain axiom is more preferred than another and thus may defeat the meaning of that axiom. For example, we could be tempted to assume that, in general, movie stars are bright people. If we came to the discovery that movie stars residing in Hollywood are actually not that clever, we would not be able to retain this information consistently. However by *defeating* the rule saying that movie stars are clever with the rule saying they are not if they are Hollywood stars, we can still retain a consistent knowledge base.

In addition to adding a preference order on axioms, implementing the notion of defeat, we restrict the semantics of [HV02], by introducing an order on the models of such a description logic knowledge base, taking into account the order on the axioms. Nonmonotonicity is then introduced by preferring models that defeat as few axioms as possible, and if defeat cannot be avoided, we select those models that defeat less preferred axioms.

Nonmonotonic reasoning in description logics is not new: e.g. [BH92] and [BH93] introduce defaults in description logic. Our approach is different, however, as it is based on an explicit ordering of defeasible axioms, as in ordered logic programming, see e.g. [GV98,GLV91,LV90]. Besides being often more intuitive, we also do not restrict ourselves to just the object names in an Abox, thus staying closer to the "open world assumption" spirit of description logics. [QR93] also works with preferred models, and thus nonmonotonicity, but axioms are just split up in defeasible and not defeasible axioms, while our approach allows not only to express defeasible knowledge but also some gradation in defeasibility, i.e. some axioms are more preferred or less defeasible than others.

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. We argue that a description logic with a preference order may prove useful when integrating ontologies, since conflicting rules may be defeated.

The remainder of this paper is organized as follows: Section 2 extends $\mathcal{SHOQ}(\mathbf{D})$ to ordered $\mathcal{SHOQ}(\mathbf{D})$ (denoted $\mathcal{OSHOQ}(\mathbf{D})$) by providing a strict partial order on the axioms, indicating a preference for certain axioms over others. Section 3 provides a nonmonotonic semantics for $\mathcal{OSHOQ}(\mathbf{D})$, effectively modeling this preference relation. Applications such as an algorithm that learns the preference order from examples and a discussion of ontology integration with $\mathcal{OSHOQ}(\mathbf{D})$ can be found in Section 4. Finally, Section 5 contains conclusions and directions for further research.

2 Extending $\mathcal{SHOQ}(\mathrm{D})$ with a Preference Order

An example

In order to obtain some intuition, consider the following example from the field of law, adapted and modified from [HVL93]. According to the law, if you steal something, you normally will be punished, i.e.

$$TP : \mathit{Thief} \sqsubseteq \mathit{Punished}$$

where we use Thief to denote the concept of people that have stolen something and $\mathit{Punished}$ for people that have received a sentence.

Instantiating this small conceptual schema with a particular individual *Bill* that was caught stealing, denoted as

$$BT : \{\mathit{Bill}\} \sqsubseteq \mathit{Thief} \ ,$$

we can deduce, from our schema and basic reasoning, that Bill must be punished ($\{\mathit{Bill}\} \sqsubseteq \mathit{Punished}$).

Now assume that, according to the law, minors (i.e. people younger than 18) do not get punished for committing crimes:

$$MP : \mathit{Minors} \sqsubseteq \neg \mathit{Punished}$$

Additionally assuming that Bill is a minor

$$BM : \{\mathit{Bill}\} \sqsubseteq \mathit{Minors}$$

leads to a problem. On the one hand we can deduce, using BM and MP, that Bill should not be punished (because he is a minor) while on the other hand, according to BT and TP, he should be punished.

To solve this contradiction, we will make explicit our tacit assumption that TP should be read as a *default*, i.e. “Thieves should be punished unless there are overriding concerns that prohibit this”. By stating that MP **defeats** TP, we indicate that MP (“Minors should not be punished”) is such an overriding concern. Informally, this means that it is acceptable to not “apply” TP as long as the defeating rule MP is applied, e.g. Bill need not be punished (no need to apply TP for Bill) if he is a minor (MP is applied for Bill).

Note that we might obtain a similar effect by refining TP to

$$TP' : \mathit{Thief} \sqcap \neg \mathit{Minors} \sqsubseteq \mathit{Punished}$$

(“Thieves that are not minors should be punished.”) which is consistent with MP, BP and BT. However, this approach does not scale well since each addition of an “exception” will make the default rule more complex and less intuitive. Humans tend not to think about exceptions when considering the truth of a general rule such as TP; only the confrontation with an actual exception such as MP will prohibit the application of the general rule. Not having to modify default rules also allows for modular specifications, where one can concentrate on adding pieces of knowledge independently, and relating them via possible defeat relationships later.

We briefly formalize the above using the description logic $\mathcal{OSHOQ}(\mathbf{D})$ [HV02], an extension of the DL $\mathcal{SHOQ}(\mathbf{D})$ [HS01].

$\mathcal{OSHOQ}(\mathbf{D})$

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 [BH91]).

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

construct name	syntax	semantics
atomic concept $\mathbf{C}$	A	$A^\mathcal{I} \subseteq \Delta^\mathcal{I}$
abstract role $\mathbf{R}_A$	R	$R^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$
concrete role $\mathbf{R}_\mathbf{D}$	T	$T^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta_\mathbf{D}$
nominals $\mathbf{I}$	$\{o\}$	$\{o\}^\mathcal{I} \subseteq \Delta^\mathcal{I}, \#\{o\}^\mathcal{I} = 1$
datatypes $\mathbf{D}$	d	$d^\mathbf{D} \subseteq \Delta_\mathbf{D}$
	$\neg d$	$(\neg d)^\mathbf{D} = \Delta_\mathbf{D} \setminus d^\mathbf{D}$
conjunction	$C \sqcap D$	$(C \sqcap D)^\mathcal{I} = C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$(C \sqcup D)^\mathcal{I} = C^\mathcal{I} \cup D^\mathcal{I}$
negation	$\neg C$	$(\neg C)^\mathcal{I} = \Delta^\mathcal{I} \setminus C^\mathcal{I}$
exists restriction	$\exists R.C$	$(\exists R.C)^\mathcal{I} = \{x \mid \exists y : (x, y) \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\}$
value restriction	$\forall R.C$	$(\forall R.C)^\mathcal{I} = \{x \mid \forall y : (x, y) \in R^\mathcal{I} \Rightarrow y \in C^\mathcal{I}\}$
atleast restriction	$\geq nR.C$	$(\geq nR.C)^\mathcal{I} = \{x \mid \#\{y \mid (x, y) \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\} \geq n\}$
atmost restriction	$\leq nR.C$	$(\leq nR.C)^\mathcal{I} = \{x \mid \#\{y \mid (x, y) \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\} \leq n\}$
datatype exists	$\exists T.d$	$(\exists T.d)^\mathcal{I} = \{x \mid \exists y : (x, y) \in T^\mathcal{I}$ and $y \in d^\mathbf{D}\}$
datatype value	$\forall T.d$	$(\forall T.d)^\mathcal{I} = \{x \mid \forall y : (x, y) \in T^\mathcal{I} \Rightarrow y \in d^\mathbf{D}\}$

A *Tbox* $\mathcal{T}$ is a finite set of *terminological axioms* $C \sqsubseteq D$ with C and D $\mathcal{SHOQ}(\mathbf{D})$-concept expressions.

Traditionally, a description logic (DL) consists of a Tbox and an Abox, where the Abox is used for assertional statements like $C(a)$ (or $R(a, b)$) which intuitively means that the individual a is an instance of C (a is related to b by means of the role R). However, in $\mathcal{SHOQ}(\mathbf{D})$ we have named individuals together with the $\{\}$-constructor ($\mathcal{O}$ in [Sch94]) and we can simulate the Abox assertions with Tbox axioms:

$$C(a) \Leftrightarrow \{a\} \sqsubseteq C$$
$$R(a, b) \Leftrightarrow \{a\} \sqsubseteq \exists R.\{b\}$$

For simplicity, we will consider the role box to be empty in the remainder of this paper, and consider only terminological axioms. It is straightforward to extend the results to knowledge bases with nonempty role boxes.

We define the defeat relation, by means of a strict[1] partial order on the Tbox axioms. Intuitively, $a_1 < a_2$, represents a preference for a_1 over a_2, i.e. a_1 defeats a_2.

Definition 1. *An $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base is a tuple*[2] $\langle \mathcal{T}, < \rangle$ *where $\mathcal{T}$ is a Tbox, and $<$ is a strict partial order between axioms of $\mathcal{T}$. For a pair $a_1 < a_2$, a_2 is said to be* ***defeasible*** *while a_1 is a (possible)* ***defeater*** *of a_2. We use $\mathcal{T}_s$ to denote the set of* ***strict*** *axioms in $\mathcal{T}$, i.e. those axioms that have no defeaters (are minimal w.r.t. $<$).*

The semantics of $\mathcal{OSHOQ}(\mathbf{D})$ 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.

The notion of defeat is formalized in the following definition.

Definition 2. *Let $\Sigma = \langle \mathcal{T}, < \rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base and $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ an* ***interpretation*** *of Σ. A terminological axiom $A \sqsubseteq B \in \mathcal{T}$ is*

- ***applicable*** *w.r.t. $x \in \Delta^{\mathcal{I}}$ and $\mathcal{I}$*[3] *iff $x \in A^{\mathcal{I}}$.*
- ***applied*** *w.r.t. $x \in \Delta^{\mathcal{I}}$ iff it is applicable w.r.t. x and $x \in B^{\mathcal{I}}$.*
- ***classically satisfied***[4] *w.r.t. $x \in \Delta^{\mathcal{I}}$ iff it is applied w.r.t. x whenever it is applicable w.r.t. x.*
- ***defeated*** *w.r.t. $x \in \Delta^{\mathcal{I}}$ iff $\exists C \sqsubseteq D < A \sqsubseteq B$ such that $C \sqsubseteq D$ is applied w.r.t. x. In this case, we say that $C \sqsubseteq D$* ***defeats*** *$A \sqsubseteq B$ w.r.t. x.*

$\mathcal{I}$ ***satisfies*** *an axiom $A \sqsubseteq B$ from $\mathcal{T}$ if for each x for which $A \sqsubseteq B$ is applicable, $A \sqsubseteq B$ is either applied or defeated, $\mathcal{I}$ is a* ***model*** *of Σ if it satisfies all the axioms in Σ.*

Essentially, the above definition allows a less preferred (larger according to the order $<$) axiom $C \sqsubseteq D$ to not be classically satisfied w.r.t. an individual x, provided that it is defeated by a more preferred applied axiom $A \sqsubseteq B < C \sqsubseteq D$ w.r.t. the same domain element x.

The earlier example, without defeat, can then be formulated as the $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base

$$\Sigma = \langle \{ \mathit{Thief} \sqsubseteq \mathit{Punished}, \mathit{Minors} \sqsubseteq \neg \mathit{Punished}, \{\mathit{Bill}\} \sqsubseteq \mathit{Thief}, \{\mathit{Bill}\} \sqsubseteq \mathit{Minors} \}, < \rangle$$

Note that the order $<$ is empty, since we did not yet impose a preference order on the axioms. That there was a problem with this knowledge base, can now formally be stated as "the knowledge base is inconsistent", i.e. there does not exist a model for it. Indeed, assume that, on the contrary, there exists a model $\mathcal{I}$ of Σ with $\{\mathit{Bill}\}^{\mathcal{I}} = \{\mathit{Bill}^{\mathcal{I}}\}$.

[1] A strict partial order $<$ on a set X is a binary relation on X that is antisymmetric, anti-reflexive and transitive. Note that this relation is also well-founded, i.e. no infinite chain $\ldots < x_n < \ldots < x_1$ exists, since the Tbox is finite.

[2] 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.

[3] In the following we will only mention $\mathcal{I}$ if it is not clear from the context.

[4] We omit the "classically" qualification if it is clear from the context.

Then it can easily be deduced that $Bill^{\mathcal{I}}$ would have to be both in $Punished^{\mathcal{I}}$ and $(\neg Punished)^{\mathcal{I}}$, which is impossible. Our solution was to defeat $Thief \sqsubseteq Punished$ with $Minors \sqsubseteq \neg Punished$, i.e. allowing a thief not to be punished if he is a minor. Formally, we add to the order $<$

$$Minors \sqsubseteq \neg Punished < Thief \sqsubseteq Punished$$

expressing precisely this intuition, yielding models such as $\mathcal{I}$, with $Thief^{\mathcal{I}} = \{Bill^{\mathcal{I}}\}$, $Minors^{\mathcal{I}} = \{Bill^{\mathcal{I}}\}$ and $Punished^{\mathcal{I}} = \emptyset$, e.g. Bill is a stealing minor who is not being punished, thus effectively bypassing the rule TP with the rule MP.

Continuing the example, we add a recent development (in Belgium) where it has been proposed that committing a serious crime leads to punishment, even if the criminal is a minor. Hence the rule

$$CP : Criminal \sqsubseteq Punished$$

where the concept $Criminal$ stands for "people that have committed a serious crime". Clearly, we would again have a contradiction if Bill is a minor and a known criminal, MP saying Bill should not be punished and CP saying that Bill should in fact be punished. However, the intention is that the CP law has precedence over MP, and thus we add CP **defeats** MP to the knowledge base (note that we do not assert that Bill is a criminal).

$$\Sigma = \langle \{ Thief \sqsubseteq Punished, Minors \sqsubseteq \neg Punished, Criminal \sqsubseteq Punished, \\ \{Bill\} \sqsubseteq Thief, \{Bill\} \sqsubseteq Minors\}, < \rangle$$

with $<$ generated by

$$Criminal \sqsubseteq Punished < Minors \sqsubseteq \neg Punished \\ \text{and } Minors \sqsubseteq \neg Punished < Thief \sqsubseteq Punished$$

Table 2. Two example models

	$Minor$	$Thief$	$Criminal$	$Punished$
$\mathcal{I}_1$	$\{bill\}$	$\{bill\}$	$\{bill\}$	$\{bill\}$
$\mathcal{I}_2$	$\{bill\}$	$\{bill\}$	$\emptyset$	$\emptyset$

Definition 2 now yields two kinds of models (see Table 2, page 6): those such as $\mathcal{I}_1$ where Bill is assumed to be a criminal (and thus should be punished) and those, like $\mathcal{I}_2$, where Bill is not a criminal and thus should not be punished ($\mathcal{I}_1$ and $\mathcal{I}_2$ share the interpretation *bill* for Bill).

Intuitively, the second type of model, exemplified by $\mathcal{I}_2$, is to be preferred since there is no reason to assume that Bill is a criminal and thus MP should not be defeated.

More precisely, we can base our preference on the fact that $\mathcal{I}_2$ (classically) satisfies more preferred rules than $\mathcal{I}_1$ does: indeed, $\mathcal{I}_2$ satisfies (w.r.t. *bill*) $\{CP, MP, BT, BM\}$ while $\mathcal{I}_1$ satisfies $\{CP, TP, BT, BM\}$. Thus, while, unlike $\mathcal{I}_1$, $\mathcal{I}_2$ does not satisfy TP, it does satisfy the more preferred MP, which is not satisfied by $\mathcal{I}_1$.

Since, for $\mathcal{I}_1$, we need to defeat a more preferred rule than we do for $\mathcal{I}_2$ ($MP < TP$), it is natural to prefer $\mathcal{I}_2$ as a model that better respects the preference order.

We formalize this intuition by defining the notion of "support" for a model as the set containing the domain elements and the axioms they satisfy, i.e. the set of "instantiated axioms" that are classically satisfied by the model.

Definition 3. *The **support** for a model $\mathcal{I}$ of $\Sigma = \langle \mathcal{T}, < \rangle$ is the set*

$$\mathcal{S}^{\mathcal{I}} = \{(x, A \sqsubseteq B) \mid x \in \Delta^{\mathcal{I}}, A \sqsubseteq B \in \mathcal{T} \text{ is (classically) satisfied w.r.t. } x \text{ and } \mathcal{I}\}$$

Whether one model is preferred over another, is then a matter of checking the supports for the models.

Definition 4. *A model $\mathcal{I}$ of a knowledge base Σ is **preferred** over a model $\mathcal{J}$ of Σ, denoted $\mathcal{I} \preceq \mathcal{J}$, if*

$$\forall (x, A \sqsubseteq B) \in \mathcal{S}^{\mathcal{J}} \setminus \mathcal{S}^{\mathcal{I}} \cdot \exists (x, C \sqsubseteq D) \in \mathcal{S}^{\mathcal{I}} \setminus \mathcal{S}^{\mathcal{J}} \cdot C \sqsubseteq D < A \sqsubseteq B$$

We use $\mathcal{I} \prec \mathcal{J}$ just when $\mathcal{I} \preceq \mathcal{J}$ and $\mathcal{S}^{\mathcal{I}} \neq \mathcal{S}^{\mathcal{J}}$ (note that $\mathcal{I} \preceq \mathcal{J}$ whenever $\mathcal{S}^{\mathcal{J}} \subseteq \mathcal{S}^{\mathcal{I}}$).

Intuitively, this means that $\mathcal{I}$ is preferred over $\mathcal{J}$ if all elements that support $\mathcal{J}$ and not $\mathcal{I}$ are countered by more preferred element that supports $\mathcal{I}$ and not $\mathcal{J}$. For the models $\mathcal{I}_1$ and $\mathcal{I}_2$ of the Bill-example (we use the names for the axioms), we have that

$$\begin{aligned}\mathcal{S}^{\mathcal{I}_1} &= \{(bill, CP), (bill, BT), (bill, BM), (bill, TP)\}\\ \mathcal{S}^{\mathcal{I}_2} &= \{(bill, CP), (bill, BT), (bill, BM), (bill, MP)\}\end{aligned}$$

and thus, $(bill, TP) \in \mathcal{S}^{\mathcal{I}_1} \setminus \mathcal{S}^{\mathcal{I}_2}$, is countered by (note that $MP < TP$) $(bill, MP) \in \mathcal{S}^{\mathcal{I}_2} \setminus \mathcal{S}^{\mathcal{I}_1}$, yielding that $\mathcal{I}_2 \prec \mathcal{I}_1$, which fits our intuition.

The preference order $\prec$ is itself a strict partial order.

Theorem 1. *Let $\Sigma = \langle \mathcal{T}, < \rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$-knowledge base. $\prec$ defines a strict partial order on the models of Σ.*

The definition of preferred models is straightforward.

Definition 5. *A model $\mathcal{I}$ of $\Sigma = \langle \mathcal{T}, < \rangle$ is a **preferred model** of Σ if there is no model $\mathcal{I}'$ of Σ, such that $\mathcal{I}' \prec \mathcal{I}$, i.e. $\mathcal{I}$ is minimal w.r.t. $\prec$.*

In the sequel, we make the following extra assumption [BMNPS02], basically to make sure that e.g. $\{Bill\}^{\mathcal{I}_1}$ does not mean something else than $\{Bill\}^{\mathcal{I}_2}$ does.

- The *Common Domain Assumption* assumes that every interpretation is defined over the same abstract domain Δ, i.e. $\Delta^{\mathcal{I}} = \Delta$ for all interpretations $\mathcal{I}$.
- The *Rigid Term Assumption* assumes a fixed function $\gamma : \mathbf{I} \to \Delta$, such that, for any individual name a and interpretation $\mathcal{I}$, $\{a\}^{\mathcal{I}} = \{\gamma(a)\}$. Thus named individuals have a fixed interpretation.

Intuitively, the above conditions restrict our attention to a single UoD (Universe of Discourse) corresponding to the knowledge base.

3 Nonmonotonic Reasoning with $\mathcal{OSHOQ}(\mathrm{D})$

While description logics can be given a user-friendly interface for designing and maintaining ontologies (see e.g. [BHGS01]), their main use lies in their reasoning capabilities. Using a description logic representation, one may for example answer the questions below.

- Given an individual o, what is its type, what classes does it belong to?
- Given a new class with certain properties, what is its place in the ontology's taxonomy? What are its sub- and super classes?
- Is a class subsumed by another class?
- Is a class satisfiable, i.e. can there exist instances of this class?
- Is the ontology consistent, i.e. does it have models?

E.g. subsumption and consistency can then be stated in $\mathcal{OSHOQ}(\mathbf{D})$ (and in other DLs, see e.g. [BS00,HST99]) as in the following definition.

Definition 6. *A $\mathcal{SHOQ}(\mathbf{D})$-concept expression C is **satisfiable** w.r.t. an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base Σ if there exists a model $\mathcal{I}$ of Σ such that $C^{\mathcal{I}} \neq \emptyset$. C is **subsumed by** a concept expression D w.r.t. Σ (notation: $\Sigma \models C \sqsubseteq D$) if $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each model $\mathcal{I}$ of Σ. Furthermore, we call Σ **consistent** iff there exists a model $\mathcal{I}$ of Σ.*

Focusing on subsumption, we alter the definition to take into account preferred models instead of models.

Definition 7. *Let Σ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base, C and D concept expressions. C is **defeasibly subsumed by** D, denoted, $\Sigma \mid\!\approx C \sqsubseteq D$, iff $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ for each preferred model $\mathcal{I}$ of Σ.*

Defeasible subsumption ($\mid\!\approx$) is "strictly weaker" than classical subsumption ($\models$), in the sense that if $\Sigma \models C \sqsubseteq D$ then $\Sigma \mid\!\approx C \sqsubseteq D$, but not the other way around.

Moreover, $\models$ is monotonic, while $\mid\!\approx$ is not.

Theorem 2 ($\models$ is monotonic). *Let $\langle \mathcal{T}, < \rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base and let A and B be concept expressions. If $\langle \mathcal{T}, < \rangle \models A \sqsubseteq B$ then $\langle \mathcal{T} \cup \mathcal{T}', < \rangle \models A \sqsubseteq B$ where $\mathcal{T}'$ is a finite set of terminological axioms*[5].

Thus, extending a knowledge base preserves earlier subsumption conclusions. This does not hold for defeasible subsumption, $\mid\!\approx$, as illustrated by the knowledge base

$$\Sigma = \langle \{P \sqsubseteq B, P \sqsubseteq \neg F, \{\mathit{Tweety}\} \sqsubseteq B, B \sqsubseteq F\}, < \rangle$$

with $<$ generated by $P \sqsubseteq \neg F < B \sqsubseteq F$. According to Σ, birds (B) tend to fly (F), penguins (P) don't fly, penguins are birds and Tweety is a bird. The rule that birds tend to fly may be defeated by the more specialized rule saying that penguins don't fly.

[5] Note that $\langle \mathcal{T} \cup \mathcal{T}', < \rangle$ may actually be inconsistent, i.e. it may not have any models. However, it is easy to verify that, for an inconsistent knowledge base, all subsumptions hold, and thus the theorem remains valid.

Note that, $\Sigma \not\models \{Tweety\} \sqsubseteq F$ since we can easily construct a model $\mathcal{I}$ where $\{Tweety\}^{\mathcal{I}} = P^{\mathcal{I}} = B^{\mathcal{I}} = \{t\}$ while $F^{\mathcal{I}} = \emptyset$, which does not satisfy $\{Tweety\} \sqsubseteq F$.

Still, in the absence of other information, in particular if we have no evidence that $Tweety$ might be a penguin, it is common sense to tentatively conclude that $Tweety$ flies, i.e. $\{Tweety\} \sqsubseteq F$. It is precisely this intuition that is captured by defeasible subsumption. Indeed, there exists another model $\mathcal{J}$ with $\{Tweety\}^{\mathcal{J}} = F^{\mathcal{J}} = B^{\mathcal{J}} = \{t\}$ and $P^{\mathcal{J}} = \emptyset$. Clearly, $\mathcal{J}$, unlike $\mathcal{I}$, classically satisfies all axioms of Σ and therefore $\mathcal{J} \prec \mathcal{I}$ and $\mathcal{I}$ is not a preferred model (but $\mathcal{J}$ is). In fact, it is easy to verify that all preferred models satisfy $\{Tweety\} \sqsubseteq F$ and therefore $\Sigma \mid\!\approx \{Tweety\} \sqsubseteq F$.

If, however, we extend Σ to Σ' by adding a fresh axiom $\{Tweety\} \sqsubseteq P$, i.e. $Tweety$ is a penguin, $\mathcal{J}$ ceases to be a model and $\mathcal{I}$ becomes a preferred model of Σ'. Thus, while $\Sigma \mid\!\approx \{Tweety\} \sqsubseteq F$, $\Sigma' \not\mid\!\approx \{Tweety\} \sqsubseteq F$, which shows that $\mid\!\approx$ is nonmonotonic.

The following characterization shows that the definition of defeasible subsumption follows our intuition: C is defeasibly subsumed by D just when, for any new individual a, if all we know about a is that it belongs to C, then it also (defeasibly) belongs to D.

Theorem 3. *Let $\Sigma = \langle \mathcal{T}, < \rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base, C and D concept expressions and a a new individual, not appearing in Σ, then*

$$\langle \mathcal{T}, < \rangle \mid\!\approx C \sqsubseteq D \iff \langle \mathcal{T} \cup \{\{a\} \sqsubseteq C\}, < \rangle \mid\!\approx \{a\} \sqsubseteq D$$

It can be checked that the consequence relation $\mid\!\approx$ satisfies some properties that are highly desirable for any nonmonotonic relation [KLM90,QR93]. More specifically, $\mid\!\approx$ can be classified as a so-called *cumulative consequence relation*, i.e. it satisfies all instances of the *Reflexivity* axiom and is closed under the inference rules of *Left Logical Equivalence*, *Right Weakening*, *Cut* and *Cautious Monotonicity* [KLM90]. A logical system where the consequence relation satisfies those properties is called a **C** (for cumulative) system.

In view of the roles of DLs as providing the basic reasoning mechanism for ontologies, an important requirement is that the reasoning procedures are decidable. We can extend the $\mathcal{SHOQ}(\mathbf{D})$ tableau algorithm, deciding satisfiability and subsumption [HS01], to incorporate the preference order and the notion of preferred models.

Theorem 4. *Let Σ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base, C and D concept expressions. $\Sigma \mid\!\approx C \sqsubseteq D$ is a decidable problem.*

4 Applications

Ontology Learning

However convenient the preference order may be, when designing an ontology for a certain application domain, the ontology engineer may not be aware of the preferences amongst axioms. He will, however, almost certainly be confronted with conflicts or inconsistencies during the design process, e.g. as a result of the DLs reasoning procedures. Inconsistencies may have various causes, like the designer wrongly assuming an axiom to be universally valid.

As an example, suppose the aim is to design an ontology regarding the nature of sports. Having seen a live coverage of a bowling game, the designer believes that in general “sports (S) is an exciting pastime (P)”. However, since the next show he saw was a cricket game, he added the belief that “English sports (E) are boring (B)”. This leads to the knowledge base

$$\Sigma = \{S \sqsubseteq P, E \sqsubseteq B, E \sqsubseteq S, P \sqsubseteq \neg B, \{cricket\} \sqsubseteq E\}$$

additionally saying that “English sport are sports”, “an exciting pastime is not boring” and “cricket is an English sport”.

The DLs reasoning procedures will tell the designer that this is an inconsistent knowledge base. However, they will not tell him that defeating one axiom with another is a possible solution for the inconsistency. We propose an order-learning algorithm for extending the order of an inconsistent knowledge base $\langle \mathcal{T}, < \rangle$ to a consistent version $\langle \mathcal{T}, <' \rangle$ where $< \subseteq <'$, i.e. if we denote $<$ and $<'$ as two subsets $\mathcal{O}$ and $\mathcal{O}'$ of $\mathcal{T} \times \mathcal{T}$, $\mathcal{O} \subseteq \mathcal{O}'$.

This algorithm [HV02], based on the candidate elimination algorithm of [Mit97], will provide the ontology designer with a choice of extensions to the current order on his inconsistent knowledge base. The decision of which order to take, remains the responsibility of the designer and should correspond with the underlying UoD, however, every order he picks is guaranteed to solve the inconsistency.

The algorithm in Table 3 is initialized with the order $<$ of the original knowledge base (viewed as a subset $\mathcal{O}$ of $\mathcal{T} \times \mathcal{T}$). By adding real-world examples (a training set E) which represent the knowledge that must be satisfied by the the resulting ontology, we minimally extend $\mathcal{O}$ as to make $\langle \mathcal{T} \cup E, \mathcal{O}' \rangle$ consistent. Additionally, we can restrict the range of possible orders by forcing certain axioms to be “strict” (as in Def. 1), and not allowing the order to defeat strict axioms. In our sports ontology we can clearly assume $E \sqsubseteq S, P \sqsubseteq \neg B$ and $\{cricket\} \sqsubseteq E$, to be strict (English sports are always sports, an exciting pastime is never a boring pastime, and cricket is an English sport).

Table 3. “candidate elimination” algorithm

1. Start with $S = \{\mathcal{O}\}$ where $\mathcal{O}$ is the set representing the original relation, and $\Sigma = \langle \mathcal{T}_0 = \mathcal{T}, < \rangle$ 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 $\langle \mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}, \mathcal{O} \rangle$ 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 Σ, to be a generator of a strict partial order such that no strict axiom is defeated) of $\mathcal{O}$ such that
      i. $\langle \mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}, \mathcal{O}' \rangle$ is consistent, and
      ii. $\mathcal{O}'$ is minimal, i.e. $\forall \mathcal{O}'', \mathcal{O} \subset \mathcal{O}'' \subset \mathcal{O}' \cdot \langle \mathcal{T}_{i-1} \cup \{\{a_i\} \sqsubseteq K_i\}, \mathcal{O}'' \rangle$ 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 algorithm fails, or all examples in E have been considered and $S \neq \emptyset$. In the latter case, the algorithm succeeds and the learned orders are in S.

Formally we have the result,

Theorem 5. *Let $\Sigma = \langle \mathcal{T}, < \rangle$ be an $\mathcal{OSHOQ}(\mathbf{D})$ knowledge base and let $E = \{\{a_1\} \sqsubseteq K_1, \ldots, \{a_n\} \sqsubseteq K_n\}$, be a set of examples. If the algorithm from Table 3 succeeds with non-empty solution set S, then $<' \in S$ iff $\langle \mathcal{T}, <' \rangle$ is a minimal order-extension ($< \subseteq <'$) of Σ such that $\langle \mathcal{T} \cup E, <' \rangle$ is consistent.*

Note that consistency means "has a model". However since $\prec$, is transitive (Theorem 1) and well-founded, we have that for each model $\mathcal{I}$ of a knowledge base Σ, $\mathcal{I}$ is preferred, or there exists a preferred model $\mathcal{J}$ of Σ, $\mathcal{J} \prec \mathcal{I}$ (formally, $\prec$ is stoppered [QR93]). Thus if Σ is consistent, it has a model $\mathcal{I}$, and either $\mathcal{I}$ is preferred or there exists a preferred model $\mathcal{J} \prec \mathcal{I}$ of Σ, so Σ is also consistent in the "has a preferred model" way.

Going back to our game of cricket, we wish to learn which minimal orders will solve the inconsistency. We provide the algorithm with the knowledge we have about cricket

$$\{cricket\} \sqsubseteq B$$

We pick-up the learning algorithm after have seen this example. Our result set S of learned orders then becomes

$$\begin{aligned} S = \{&\{E \sqsubseteq B < S \sqsubseteq P\}, \\ &\{E \sqsubseteq S < S \sqsubseteq P\}, \\ &\{\{cricket\} \sqsubseteq E < S \sqsubseteq P\}\} \end{aligned}$$

The designer then has three corresponding choices. He probably should not choose the third order, since this involves defeating the rule with a single fact, which may not be general enough. The choice between the first and the second is a matter of a taste. In both cases a sport that is not exciting must be an English sport. The second choice adds nothing, being an English sport is enough for not being exciting, while the first choice would amount to not only having an English sport but also the rule that English sports are boring.

Ontology Integration

We describe integration of ontologies as the problem of merging two ontologies into a single unified ontology. In addition to representing the information from both ontologies, the integrated ontology should also describe the relationships between them.

In practice, merging ontologies is a complex multi-layered and difficult to automate task. Possible problems include [BK00], for example, two terms with the same name in different ontologies, but actually representing different concepts. E.g. in one ontology, the term $President$ may mean "president of a country", while in the other it may be used to express "president of a company". Another related problem is two different terms that represent the same concept in different ontologies: $Mafia$ and Mob can both represent a large organized group of criminals. Although these and other problems are crucial to ontology integration, we assume here a simplified setting where different terms in different ontologies represent different concepts, and terms have a single common meaning.

In this setting, we attempt to integrate two consistent ontologies (expressed as DLs knowledge bases) in a new ontology. The basic procedure is then, assuming all necessary pre-processing has taken place, to take the union of the two DLs. Clearly, this new ontology may not be consistent. A typical action would be, as the SMART algorithm for merging and aligning ontologies indicates [NM99], to put the conflict on a conflict list together with actions that remedy the inconsistency.

This is were $\mathcal{OSHOQ}(\mathbf{D})$ becomes useful. While there may be several reasons for having an inconsistency, it could well be that an inconsistency arose because of the fact that a certain axiom is in general more preferred than another one, or because an axiom applies in a general ontology while it has exceptions in a more specific ontology, or, in the most extreme case, an axiom in one ontology may just be wrong, and it should be defeated by another one. All the foregoing problems can be relatively easily solved by placing a preference order on some axioms of the ontologies.

The ontology learning algorithm in Table 3 can, when encountering an inconsistency, suggest several orders that remedy the inconsistency. It is then up to the designer to decide which order to choose according to the UoD he is modeling. As in the SMART algorithm, the designer should be able to adapt some parameters as to (partially) automate this behavior. For example, when integrating two ontologies where one of them has greater authority (e.g., it comes from a trusted source), axioms of this preferred ontology should always be preferred.

To make things more explicit we take a look at a toy example. Consider two little ontologies, one representing the knowledge that quakers are pacifists and that Nixon is a quaker, while the other acknowledges the fact that republicans are not pacifists and that Nixon is a republican.

$$\mathcal{O}_1 = \langle \{ \mathit{quakers} \sqsubseteq \mathit{pacifists}, \{\mathit{Nixon}\} \sqsubseteq \mathit{quakers} \} \rangle$$

and

$$\mathcal{O}_2 = \langle \{ \mathit{republicans} \sqsubseteq \neg \mathit{pacifists}, \{\mathit{Nixon}\} \sqsubseteq \mathit{republicans} \} \rangle$$

While both ontologies are consistent in their own right, when one attempts to unite the knowledge in a new ontology

$$\begin{aligned}\mathcal{O} = \langle \{ &\mathit{quakers} \sqsubseteq \mathit{pacifists}, \{\mathit{Nixon}\} \sqsubseteq \mathit{quakers},\\ &\qquad \mathit{republicans} \sqsubseteq \neg \mathit{pacifists}, \{\mathit{Nixon}\} \sqsubseteq \mathit{republicans} \} \rangle\end{aligned}$$

one ends up with an inconsistent knowledge base. More specifically, Nixon can be shown to be both a pacifist and a non-pacifist.

However if, for some reason, ontology $\mathcal{O}_2$ is more preferred than $\mathcal{O}_1$, for example because $\mathcal{O}_2$ was released by the government and $\mathcal{O}_1$ by some unauthorized Nixon website, we have the means to incorporate this preference in $\mathcal{OSHOQ}(\mathbf{D})$ by simply defining a preference order $<$ on $\mathcal{O}$, such that $a < b$ for every axiom a in $\mathcal{O}_2$ and b

in $\mathcal{O}_1$, claiming that axioms from the authorized source are preferred. From this new $\mathcal{OSHOQ}(\mathbf{D})$ ontology

$$\mathcal{O} = \langle \{ quakers \sqsubseteq pacifists, \{Nixon\} \sqsubseteq quakers, \\ republicans \sqsubseteq \neg pacifists, \{Nixon\} \sqsubseteq republicans\}, < \rangle$$

we can deduce, with the preferred model semantics, that Nixon is not a pacifist.

5 Conclusion and Directions for Further Research

We provided a nonmonotonic extension for the $\mathcal{SHOQ}(\mathbf{D})$ description logic, by imposing a strict partial order on the axioms. In this way we were able to express that not all knowledge can be caught in rigid rules since rules may be valid in a general situation but have exceptions where more preferred rules override the general case. The preferred model semantics provides a natural way to express such situations.

We discussed an order learning algorithm that, given an inconsistent knowledge base, can suggest orders to solve the inconsistency. While $\mathcal{OSHOQ}(\mathbf{D})$ does not claim to solve the ontology integration problem as such, it can however be a helpful instrument for removing inconsistencies from the merged ontology, by suggesting (or automatically enforcing) a preference on axioms.

For the future, it would be interesting to see how $\mathcal{OSHOQ}(\mathbf{D})$ exactly relates to other nonmonotonic description logics, for example to the general framework in [QR93]. Also other approaches, like the ordered theory presentation of [Rya92] may provide useful insights.

References

[BGH01] 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.

[BH91] Franz Baader and Philipp Hanschke. A scheme for integrating concrete domains into concept languages. Technical Report RR-91-10, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 1991.

[BH92] Franz Baader and Bernhard Hollunder. Embedding defaults into terminological knowledge representation formalisms. In Bernhard Nebel, Charles Rich, and William 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.

[BH93] Franz Baader and Bernhard Hollunder. How to prefer more specifi c defaults in terminological default logic. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artifi cial Intelligence*, pages 669–674, San Mateo, California, 1993. Morgan Kaufmann.

[BHGS01] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. OilEd: a reasonable ontology editor for the semantic web. In *Proceedings of KI2001, Joint German/Austrian conference on Artifi cial Intelligence*, number 2174 in Lecture Notes in Computer Science, pages 396–408, Vienna, September 2001. Springer-Verlag.

[BK00] A. Borgida and R. Küsters. What's not in a name: Some Properties of a Purely Structural Approach to Integrating Large DL Knowledge Bases. In F. Baader and U. Sattler, editors, *Proceedings of the International Workshop on Description Logics (DL2000)*, number 33 in CEUR-WS, 2000.

[BLHL01] T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientifi c American*, pages 34–43, May 2001.

[BMNPS02] F. Baader, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider, editors. *Description Logic Handbook*, chapter 2, pages 47–100. Cambridge University Press, 2002.

[BS00] F. Baader and U. Sattler. Tableau algorithms for description logics. In R. Dyckhoff, editor, *Proceedings of the International Conference on Automated Reasoning with Tableaux and Related Methods (Tableaux 2000)*, volume 1847 of *Lecture Notes in Artifi cial Intelligence*, pages 1–18, St Andrews, Scotland, UK, 2000. Springer-Verlag.

[DvHB+00] S. Decker, F. van Harmelen, J. Broekstra, M. Erdmann, D. Fensel, I. Horrocks, M. Klein, and S. Melnik. The Semantic Web - on the respective roles of XML and RDF. *IEEE Internet Computing*, September-October 2000.

[FHvH+00] D. Fensel, I. Horrocks, F. van Harmelen, S. Decker, M. Erdmann, and M. Klein. OIL in a Nutshell. In R. Dieng et al., editor, *Knowledge Acquisition, Modeling, and Management, Proceedings of the European Knowledge Acquisition Conference (EKAW-2000)*, Lecture Notes in Artifi cial Intelligence. Springer-Verlag, 2000.

[FvHH+01] D. Fensel, F. van Harmelen, I. Horrocks, D. McGuinness, and P. F. Patel-Schneider. OIL: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16(2):38–45, 2001.

[GLV91] 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.

[GV98] 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.

[HFB+00] I. Horrocks, D. Fensel, J. Boekstra, S. Decker, M. Erdmann, C. Goble, F. Van 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.

[Hor00] I. Horrocks. A denotational semantics for Standard OIL and Instance OIL. `http://www.ontoknowledge.org/oil/downl/semantics.pdf`, 2000.

[HS01] 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 Artifi cial Intelligence (IJCAI-01)*, pages 199–204. Morgan Kaufmann, 2001.

[HST99] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In Harald Ganzinger, David McAllester, and Andrei 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.

[HST00] I. Horrocks, U. Sattler, and S. Tobies. Reasoning with individuals for the description logic $\mathcal{SHIQ}$. In David MacAllester, editor, *Proc. of the 17th Int. Conf. on Automated Deduction (CADE-17)*, number 1831 in Lecture Notes In Artifi cial Intelligence, pages 482–496. Springer-Verlag, 2000.

[HV02] S. Heymans and D. Vermeir. Using Preference Order in Ontologies. 2002. To appear.

[HVL93] J.C. Hage, H.B. Verheij, and A.R. Lodder. Reason Based Logic; a Logic that Deals with Rules and Reasons. In J.M. Akkermans and J.A. Breuker, editors, *Working Papers NAIC '93*, pages 293–304. 1993.

[KLM90] Sarit Kraus, Daniel J. Lehmann, and Menachem Magidor. Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, 44(1-2):167–207, 1990.

[LS99] Ora Lassila and Ralph R. Swick. Resource description framework (RDF) model and syntax specification. W3C Recommendation - `http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/`, February 1999.

[LV90] E. Laenens and D. Vermeir. A Logical Basis for Object Oriented programming. In J. Siekmann, editor, *Logics in AI*, number 478 in Lecture notes in Artificial Intelligence, pages 317–332. Springer-Verlag, September 1990.

[Mit97] Tom Mitchell. *Machine Learning*. McGraw-Hill, 1997.

[NM99] N.F. Noy and M.A. Musen. SMART: Automated Support for Ontology Merging and Alignment. In *Proceedings of the Twelfth Workshop on Knowledge Acquisition (KAW'99)*, Banf, Canada, 1999.

[QR93] J.J. Quantz and M. Ryan. Preferential Default Description Logics. KIT 110, Technische Universität Berlin, 1993.

[Rya92] Mark Ryan. Representing Defaults as Sentences with Reduced Priority. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 649–660. Morgan Kaufmann, San Mateo, California, 1992.

[Sch94] Andrea Schaerf. Reasoning with individuals in concept languages. *Data Knowledge Engineering*, 13(2):141–176, 1994.

[UG96] Mike Uschold and Michael Grüninger. Ontologies: principles, methods, and applications. *Knowledge Engineering Review*, 11(2):93–155, 1996.