Closed World Reasoning in the Semantic Web through Epistemic Operators

Stephan Grimm and Boris Motik

FZI Research Center for Information Technologies at the University of Karlsruhe
Karlsruhe, Germany
{grimm,motik}@fzi.de

Abstract. The open world assumption makes OWL principally suitable to handle incomplete knowledge in Semantic Web scenarios, however, some scenarios desire closed world reasoning. Autoepistemic description logics allow to realise closed world reasoning in open world settings through epistemic operators. An extension of OWL by epistemic operators therefore allows for non-monotonic features known from closed world systems, such as default rules, integrity constraints or epistemic querying. These features can be beneficially applied in Semantic Web scenarios, where OWL lacks expressiveness.

1 Introduction

An important goal in the design of the Ontology Web Language (OWL) [8] was to produce a language with a well-defined formal semantics. This goal was achieved by basing the semantics on *description logics* (DL) [2]. The DLs underlying OWL are actually fragments of first-order logic, so they employ the so-called *open world assumption* (OWA) [6]. Under OWA, failure to derive a fact does not imply the opposite. For example, assume we only know that Peter is a person. From this information we can neither conclude that Peter is a vegetarian, nor that he is not one. Hence, we admit the fact that our knowledge of the world is incomplete. The open world assumption is closely related to the *monotonic* nature of first-order logic: adding new information never falsifies a previous conclusion. Namely, if we subsequently learn that Peter is indeed a vegetarian, this does not change any positive or negative conclusions.

The open world assumption seems to correctly model much of day-to-day reasoning. However, the framework of first-order logic may be unsuitable for certain situations which require complete knowledge about the world. Consider a table of train departure times. If the table does not explicitly state that a train leaves at 12:47, then we usually conjecture that there is no such train. In other words, for train time-tables we typically use the *closed world assumption* (CWA), assuming that our knowledge about that part of the world is complete. Under CWA, we conclude that there is no train at 12:47 unless we can prove the contrary. Such inference is *non-monotonic*, meaning that additional knowledge can invalidate previous conclusions. For example, learning that there is a train at 12:47 invalidates our earlier conjecture.

Many knowledge modelling constructs are related to CWA and cannot be expressed in first-order logic. *Default rules* allow for modelling exceptions. For example, we may make a common conjecture that people eat meat, unless we know them to be vegetarians. This rule relieves us from the burden of explicitly asking each person whether he is a vegetarian or not.

Constraints also depend on closed world reasoning. For example, we could easily embarrass ourselves by inviting a vegetarian to dinner and then preparing Beef Stoganoff just because we forgot to ask the guest if he is a vegetarian. To prevent such situations, we might introduce a constraint stating that for each guest his views on eating meat should be known.

Choosing between OWA or CWA is often an all-or-nothing game, thus posing problems for applications which need to deal with both kinds of information at once. For example, in an application dealing with travelling vegetarians, assuming that a person eats meat just because we do not know that he is not a vegetarian may be wrong; however, assuming that one might get to the International Vegetarianism Convention by a train which is not listed in the time table, seems wrong as well. In other words, we believe that many applications require OWA and CWA in parallel, allowing for *local closed world* (LCW) *reasoning* [4]. Such reasoning is based on the OWA augmented by the possibility to explicitly *close off* parts of the world.

A common objection to extending DLs with non-monotonic constructs is that completeness of knowledge can be stated in a purely first-order setting. For example, using nominals one can restrict an interpretation of a concept to exactly the specified set of individuals. However, this solves the problem only partially, since there is no equivalent nominal construct for roles. Moreover, such a solution does not provide *introspection* — reasoning about the state of the knowledge base. Introspection is not definable in first-order logic, but is necessary for formalising defaults or constraints. Similarly, a common objection to introducing defaults and constraints is that they should be realised outside the logic, for example, by checking for missing information in a preprocessing step. We strongly disagree with such a view. Namely, it is unclear how to define the semantics of such a step. If the semantics were defined in an ad-hoc manner, we would soon experience the same problems observed in the early frame representation systems, which eventually lead to formal reconstruction of their semantics.

To summarise, we believe that OWL should be extended with non-monotonic constructs. In this paper we sketch a possible solution based on *autoepistemic description logics* (ADL) [3]. Of all candidate formalisms, we find this formalism to be particularly suitable since it properly extends OWL. We show how ADLs can be used to provide local closed world reasoning, default rules and constraints in the Semantic Web setting. Whereas such applications of ADLs were already discussed in [3], with our presentation we aim at additionally explaining some technicalities underlying ADLs. Furthermore, our goal is to demonstrate the benefits of non-monotonic extensions of OWL to the Semantic Web community. Finally, we point out to the remaining questions which need to be answered to realise a non-monotonic extension of OWL by epistemic operators.

2 Epistemic Operators for OWL

Autoepistemic logic is a formalism concerned with the notions of 'knowledege' and 'assumption' and allows for introspection of knowledge bases, i.e. to ask what a knowledge base *knows* or *assumes.* (See e.g. [1].) In this section we present an autoepistemic extension to DL introduced in [3]. Although OWL-DL corresponds to the expressive DL $\mathcal{SHOIN}(D)$, we adopt the simpler DL $\mathcal{ALC}$ for this extension, for which the underlying theory is covered by [3]. One of the open research problems remains how this theory can be extended to also cover additional constructs in $\mathcal{SHOIN}(D)$ and reasoning with OWL ontologies.

Autoepistemic Description Logics

In [3] the basic DL $\mathcal{ALC}$ has been extended by two operators, **K** and **A**, reflecting the notions of 'knowledge' and 'assumption'. The following rules define the syntax of the resulting language $\mathcal{ALCK}_{\mathcal{NF}}$, where C, D denote concepts, A denotes a primitive concept, r denotes a role and p denotes a primitive role.

$$\begin{array}{rcl} C, D & \longrightarrow & A \mid \top \mid \bot \mid C \sqcap D \mid C \sqcup D \mid \neg C \mid \forall\, r.C \mid \exists\, r.C \mid \mathbf{K}C \mid \mathbf{A}C \\ r & \longrightarrow & p \mid \mathbf{K}p \mid \mathbf{A}p \end{array}$$

An *epistemic interpretation* is a triple $(\mathcal{I}, \mathcal{W}_{\mathbf{K}}, \mathcal{W}_{\mathbf{A}})$ where $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a *first-order interpretation* with interpretation domain $\Delta^{\mathcal{I}}$ and interpretation function $\cdot^{\mathcal{I}}$, and $\mathcal{W}_{\mathbf{K}}, \mathcal{W}_{\mathbf{A}}$ are sets of first-order interpretations, seen as possible worlds for the two modalities **K** and **A** in the sense of modal logics. The following equations define how the elements of $\mathcal{ALCK}_{\mathcal{NF}}$ are epistemically interpreted.

$$\begin{array}{rcl} \top^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \Delta^{\mathcal{I}} & , & \bot^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \emptyset \\ A^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} & , & p^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = p^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \end{array}$$

$$\begin{array}{rcl} (C \sqcap D)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} &=& C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \cap D^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \\ (C \sqcup D)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} &=& C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \cup D^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \\ (\neg C)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} &=& \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \\ (\forall\, r.C)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} &=& \{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in r^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \rightarrow b \in C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}}\} \\ (\exists\, r.C)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} &=& \{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in r^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \wedge b \in C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}}\} \end{array}$$

$$\begin{array}{rcl} (\mathbf{K}C)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \bigcap_{\mathcal{J}\in\mathcal{W}_{\mathbf{K}}} C^{\mathcal{J},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} & , & (\mathbf{A}C)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \bigcap_{\mathcal{J}\in\mathcal{W}_{\mathbf{A}}} C^{\mathcal{J},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \\ (\mathbf{K}p)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \bigcap_{\mathcal{J}\in\mathcal{W}_{\mathbf{K}}} p^{\mathcal{J},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} & , & (\mathbf{A}p)^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} = \bigcap_{\mathcal{J}\in\mathcal{W}_{\mathbf{A}}} p^{\mathcal{J},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \end{array}$$

Primitive concepts are interpreted as subsets of $\Delta^{\mathcal{I}}$, and primitive roles as subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The boolean connectives and existential and universal role quantification are interpreted in terms of set operations on $\Delta^{\mathcal{I}}$, as in $\mathcal{ALC}$ [2]. Epistemic concepts $\mathbf{K}C$ and $\mathbf{A}C$ are interpreted as the sets of all individuals which belong to the concept C in all first-order interpretations in $\mathcal{W}_{\mathbf{K}}$ and $\mathcal{W}_{\mathbf{A}}$, respectively. Similarly, epistemic roles $\mathbf{K}p$ and $\mathbf{A}p$ are interpreted as the pairs of individuals that belong to the role p in all possible worlds in $\mathcal{W}_{\mathbf{K}}$ and $\mathcal{W}_{\mathbf{A}}$.

An epistemic interpretation satisfies an inclusion axiom $C \sqsubseteq D$ if $C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}} \subseteq D^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}}$, and it satisfies an assertion axiom $C(a)$ or $r(a,b)$ if $a^{\mathcal{I}} \in C^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}}$ or $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in r^{\mathcal{I},\mathcal{W}_{\mathbf{K}},\mathcal{W}_{\mathbf{A}}}$, respectively. An *epistemic model* for an $\mathcal{ALCK}_{\mathcal{NF}}$ knowledge base *KB* is a non-empty set $\mathcal{M}$ of first-order interpretations such that, for each $\mathcal{I} \in \mathcal{M}$, the epistemic interpretation $(\mathcal{I}, \mathcal{M}, \mathcal{M})$ satisfies all axioms in *KB* and there is no set $\mathcal{M}'$ of first order interpretations such that $\mathcal{M} \subset \mathcal{M}'$ and the epistemic interpretation $(\mathcal{I}, \mathcal{M}', \mathcal{M})$ also satisfies all axioms in *KB*. An $\mathcal{ALCK}_{\mathcal{NF}}$ knowledge base *KB* is satisfiable if it has an epistemic model. It entails an axiom α, denoted by $KB \models \alpha$, if α is satisfied in all its epistemic models.

Intuition behind Epistemic Operators

The semantics of both epistemic operators, **K** and **A**, is defined as an intersection of concept/role extensions over sets of first-order interpretations $\mathcal{W}_{\mathbf{K}}, \mathcal{W}_{\mathbf{A}}$, seen as possible worlds. Therefore they both ensure statements to hold in all possible worlds in these sets. The difference between **K** and **A** lies in the restrictions about which worlds belong to $\mathcal{W}_{\mathbf{K}}$ and $\mathcal{W}_{\mathbf{A}}$, respectively.

To see this difference, consider the knowledge bases $KB = \{\exists\, r.C(a)\}$, $KB_{\mathbf{K}} = \{\exists\, r.\mathbf{K}C(a)\}$ and $KB_{\mathbf{A}} = \{\exists\, r.\mathbf{A}C(a)\}$. The set of all first-order models of *KB*, denoted by $\mathcal{M}(KB)$, can be verified to be the unique epistemic model for *KB*. However, $\mathcal{M}(KB)$ is not an epistemic model for $KB_{\mathbf{K}}$, since it contains first-order interpretations in which the r-successors of a do not constantly belong to C over all $\mathcal{J} \in \mathcal{M}(KB)$. The use of the **K**-operator in $KB_{\mathbf{K}}$ requires the existence of an r-successor for a which belongs to C in all possible worlds, i.e. which is *known* to be in the extension of C. The set $\mathcal{M}_x \subset \mathcal{M}(KB)$, defined by $\{\mathcal{I} : \mathcal{I} \models r(a,x) \wedge C(x)\}$ for some $x \in \Delta^{\mathcal{I}}$, fulfils this condition. It is an epistemic model for $KB_{\mathbf{K}}$, since the epistemic interpretation $(\mathcal{I}, \mathcal{M}_x, \mathcal{M}_x)$ satisfies the axiom in $KB_{\mathbf{K}}$ whereas $(\mathcal{I}, \mathcal{M}_x \cup \{\mathcal{I}'\}, \mathcal{M}_x)$ does not, for any $\mathcal{I}' \in \mathcal{M}(KB) \backslash \mathcal{M}_x$[1]. In this sense **K** can be paraphrased as "known".

Conversely, neither any $\mathcal{M}_x$ nor any other set of first order interpretations is an epistemic model for $KB_{\mathbf{A}}$, which is unsatisfiable. To see this, consider any set $\mathcal{M}$ of first-order interpretations for which $(\mathcal{I}, \mathcal{M}, \mathcal{M})$ satisfies $KB_{\mathbf{A}}$. To verify $\mathcal{M}$ as being maximal, $(\mathcal{I}, \mathcal{M}', \mathcal{M})$ must not satisfy $KB_{\mathbf{A}}$ for any set $\mathcal{M}' \supset \mathcal{M}$. However, the choice of $\mathcal{M}'$ does not affect the modality **A**. The set $\mathcal{M}$ could only be an epistemic model if it would already be maximal, such that there is no set $\mathcal{M}'$. In this sense, the use of the **A**-operator in $KB_{\mathbf{A}}$ refers to individuals that are *assumed* to be in the extension of C already, and **A** can therefore be paraphrased as "assumed". If this assumption is not justified by other facts then the knowledge base becomes unsatisfiable.

3 Realising Local Closed World Reasoning

In this section we show how LCW reasoning can be realised with the epistemic operators **K** and **A**. We distinguish cases in which epistemic operators are used inside the knowledge base from those in which they are used outside only.

If a non-epistemic knowledge base *KB* is queried by use of epistemic concepts, we only have occurrences of epistemic operators outside *KB*. In this case they

[1] Since we did not use $\exists\, \mathbf{K}r.\mathbf{K}C(a)$ in $KB_{\mathbf{K}}$, there are even epistemic models $\mathcal{M}_\sigma = \{\mathcal{I} : \mathcal{I} \models \bigvee_{x\in\sigma} r(a,x) \wedge \bigwedge_{x\in\sigma} C(x)\}$ for any subset $\sigma \subset \Delta^{\mathcal{I}}$

don't affect the epistemic models of KB, since a non-epistemic knowledge base always has a unique epistemic model $\mathcal{M}(KB)$ [3], which is just the set of all its first-order models. In the more general case, epistemic operators can also occur in the axioms of a knowledge base. Then the epistemic models of such a knowledge base are determined by its epistemic axioms.

Epistemic Queries

To query a knowledge base KB means to ask for those individuals that have certain properties specified by a concept. Therefore a query is often defined as a concept C and querying reduces to checking the entailment of concept assertions $C(\iota)$ for all known individuals ι in KB. We will consider the single case of checking the entailment $KB \models C(a)$ for a certain individual a.

In this sense, an *epistemic query* is an epistemic concept C that is posed as a query to a non-epistemic knowledge base KB. To validate the entailment $KB \models C(a)$, the assertion $C(a)$ has to be satisfied by epistemic interpretations $(\mathcal{I}, \mathcal{M}, \mathcal{M})$ for any epistemic model $\mathcal{M}$ of KB. However, since KB is non-epistemic, it is sufficient to consider the set $\mathcal{M}(KB)$ of its first-order models.

To see how the **K**-operator [2] affects the querying abilities for knowledge bases, we investigate the model-theoretic situation for the following example of an epistemic query, inspired by the popular pizza scenario from [9].

$$KB = \{\ topping(pizza, tomato),\ topping(pizza, chili),\ \neg SpicyTopping(tomato)\ \}$$

$$Q_1: \quad KB \not\models \{\ \forall\ topping.\neg SpicyTopping(pizza)\ \}$$
$$Q_2: \quad KB \models \{\ \forall\ \mathbf{K}topping.\neg\mathbf{K}SpicyTopping(pizza)\ \}$$

The pizza in KB has assigned two toppings, of which only *tomato* is declared as non-spicy whereas for *chili* there is no such information. The queries Q_1 and Q_2 ask whether *pizza* has non-spicy toppings in two different ways. Query Q_1 is asking whether all toppings of *pizza* are non-spicy, using 'ordinary' DL constructs. However, this is not entailed by KB for two reasons: a) it is not sure whether *pizza* has any toppings other than the ones asserted in KB, and b) for the *chili* topping it is not clear whether it is spicy or not.

Query Q_2 is asking whether all known toppings of *pizza* are not known to be spicy, using the **K**-operator. Since for neither *tomato* nor *chili* there is evidence to conclude that they are spicy, we would expect the answer to be positive. To verify the entailment for Q_2 we have to check whether the epistemic interpretation $(\mathcal{I}, \mathcal{M}(KB), \mathcal{M}(KB))$ satisfies the concept assertion in Q_2 for all first-order models $\mathcal{I} \in \mathcal{M}(KB)$. The epistemic role $\mathbf{K}topping$ is interpreted as the intersection of pairs of individuals in *topping* over all first-order models of $\mathcal{M}(KB)$. This eliminates those pairs which sometimes belong to the role extension and sometimes not, leaving only $topping(pizza, tomato)$ and $topping(pizza, chili)$. Analogously, for the epistemic concept $\mathbf{K}SpicyTopping$ those individuals are eliminated that sometimes belong to the concept *SpicyTopping* and sometimes

[2] Observe that Outside the knowledge base the two operators **K** and **A** show the same behaviour [3]. Therefore we use only **K** in epistemic queries.

not. Therefore the negated expression $\neg\mathbf{K}SpicyTopping$ refers to those toppings for which it is not clear that they are spicy from the given knowledge, which includes *tomato* and *chili*. Hence, Q_2 is answered positively.

Here, LCW reasoning is realised by closing off the role *topping* and the concept *SpicyTopping*, reducing the 'don't known' answers in the reasoning. The conjecture made is "if a pizza is not definitely known to have a spicy topping it is assumed to be safe".

Epistemic Axioms

An *epistemic axiom*, either inclusion or assertion, is an axiom that contains an epistemic concept. If we allow epistemic axioms in a knowledge base *KB*, things become trickier because *KB* potentially has several epistemic models which have to be determined for reasoning. As some special cases of epistemic axioms we look at default rules and integrity constraints.

Default Rules

A *default rule* according to [10] has the form $\alpha : \beta \,/\, \gamma$ and is read as "if α is true and it is consistent to assume that β is true then conclude that γ is true". In [3] it has been shown that such a default rule can be formalised as the epistemic axiom $\mathbf{K}\alpha \sqcap \neg\mathbf{A}\neg\beta \sqsubseteq \mathbf{K}\gamma$ [3]. We explain the default behaviour of this formalisation by the following example of a knowledge base augmented by a default rule.

$$\begin{aligned} KB &= \{\ Pizza(margarita),\ Pizza \sqcap \neg FlatDish(calzone)\ \} \\ D &= \{\ \mathbf{K}Pizza \sqcap \neg\mathbf{A}\neg FlatDish \sqsubseteq \mathbf{K}FlatDish\ \} \\ KB \cup D &\models \{\ FlatDish(margarita),\ \neg FlatDish(calzone)\ \} \end{aligned}$$

Of the pizzas in the knowledge base *KB*, only for *calzone* it is known whether it is a flat dish – for *margarita* there is no such information. The default rule in D says that pizzas are typically flat dishes, unless specified otherwise. Included in *KB*, we would intuitively like this default rule to be applied on the pizza *margarita*, concluding that it is a flat dish, but not on the pizza *calzone*, since it is already asserted to be no flat dish. We will verify these conclusions by determining the epistemic models of *KB*.

To obtain candidates for epistemic models of $KB \cup D$, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two partitions for all first-order models $\mathcal{M}(KB)$ of *KB*, such that $\mathcal{M}_1 = \{\mathcal{I} \in \mathcal{M}(KB) : \mathcal{I} \models FlatDish(margarita)\}$ and $\mathcal{M}_2 = \{\mathcal{I} \in \mathcal{M}(KB) : \mathcal{I} \models \neg FlatDish(margarita)\}$. Interpretations $\mathcal{I} \notin \mathcal{M}(KB)$ can be ruled out, since they do not satisfy *KB*, and other candidate sets $\mathcal{M}_{12}$, containing interpretations from both $\mathcal{M}_1$ and $\mathcal{M}_2$, do not satisfy the inclusion axiom in D because *margarita* is in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} Pizza^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$ and not in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} \neg FlatDish^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$ but not in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} FlatDish^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$, making the inclusion false. We verify that only $\mathcal{M}_1$ is an epistemic model of $KB \cup D$ using Table 1, which shows the extensions of the epistemic concepts involved in the inclusion from D for different epistemic interpretations. The epistemic interpretation $(\mathcal{I}, \mathcal{M}_1, \mathcal{M}_1)$ satisfies $KB \cup D$, since the inclusion in D is true for both individuals: *margarita* is in

[3] We exclude prerequisite-free defaults (no presence of α) and cases where α=$\top$, see [3]

	$\mathcal{I}, \mathcal{M}_1, \mathcal{M}_1$	$\mathcal{I}, \mathcal{M}'_1, \mathcal{M}_1$	$\mathcal{I}, \mathcal{M}_2, \mathcal{M}_2$	$\mathcal{I}, \mathcal{M}'_2, \mathcal{M}_2$
$\mathbf{K}Pizza$	$\{cal, mar\}$	$\{cal, mar\}$	$\{cal, mar\}$	$\{cal, mar\}$
$\mathbf{A}\neg FlatDish$	$\{cal\}$	$\{cal\}$	$\{cal, mar\}$	$\{cal, mar\}$
$\mathbf{K}FlatDish$	$\{mar\}$	$\{\}$	$\{\}$	$\{\}$

Table 1. Extensions of epistemic concepts in different epistemic interpretations

$\bigcap_{\mathcal{J}\in\mathcal{M}_1} Pizza^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$, not in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} \neg FlatDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ and in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} FlatDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$, whereas *calzone* is not in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} \neg FlatDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$. To check whether $\mathcal{M}_1$ is indeed an epistemic model for $KB \cup D$ we need to verify its maximality. Let $\mathcal{M}'_1 := \mathcal{M}_1 \cup \mathcal{I}'$ for some $\mathcal{I}' \in \mathcal{M}_2$. The epistemic interpretation $(\mathcal{I}, \mathcal{M}'_1, \mathcal{M}_1)$ does not satisfy $KB \cup D$, since *margarita* is still not in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} \neg FlatDish^{\mathcal{J},\mathcal{M}'_1,\mathcal{M}_1}$, as before, but not in $\bigcap_{\mathcal{J}\in\mathcal{M}'_1} FlatDish^{\mathcal{J},\mathcal{M}'_1,\mathcal{M}_1}$, contradicting the inclusion.

If we check whether $\mathcal{M}_2$ is also an epistemic model of $KB \cup D$, we observe that $(\mathcal{I}, \mathcal{M}_2, \mathcal{M}_2)$ does also satisfy the axioms in $KB \cup D$. However, $\mathcal{M}_2$ does not fulfil the maximality condition: if we consider the set $\mathcal{M}'_2 := \mathcal{M}_2 \cup \{\mathcal{I}'\}$, for some $\mathcal{I}' \in \mathcal{M}_1$, then $(\mathcal{I}, \mathcal{M}'_2, \mathcal{M}_2)$ does not contradict the inclusion because *margarita* is in $\neg FlatDish^{\mathcal{I},\mathcal{M}'_2,\mathcal{M}_2}$ for all $\mathcal{J}\in\mathcal{M}_2$.

Having determined $\mathcal{M}_1$ as the only epistemic model of $KB \cup D$, it can be seen that the inferences given above are correct, since in $\mathcal{M}_1$ *margarita* is always a flat dish whereas *calzone* never is.

From this example it can be seen that default rules can be used to reduce don't know answers in reasoning by eliminating those epistemic models which contain uncertainty about a particular property.

Integrity Constraints

The concept of integrity constraint is known from the field of databases. An *integrity constraint* is used to check the state of a knowledge base without deriving new facts – something that cannot be done in OWL. In [3] it has been shown that ADLs are well suited to formalise integrity constraints due to their introspective nature. We describe this formalisation by the following example.

$$KB = \{ \ \mathbf{K}Pizza \sqsubseteq \mathbf{A}\exists\, topping.\top, \ Pizza(pizzabread) \ \}$$

The integrity constraint in KB says that any individual that is known to be a pizza is also assumed to have some topping. We verify that KB has no epistemic model due to the fact that *pizzabread* has no asserted topping.

Again, we consider a set of first-order interpretations $\mathcal{M}_1 = \{\mathcal{I} : \mathcal{I} \models Pizza(pizzabread) \wedge \exists\ topping.\top(pizzabread)\}$. The epistemic interpretation $(\mathcal{I}, \mathcal{M}_1, \mathcal{M}_1)$ satisfies KB, since *pizzabread* is in $Pizza^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ and also in $\exists\ topping.\top^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ for all $\mathcal{J} \in \mathcal{M}_1$. However, for any set $\mathcal{M}'_1 \supset \mathcal{M}_1$ the epistemic interpretation $(\mathcal{I}, \mathcal{M}'_1, \mathcal{M}_1)$ also satisfies KB, since *pizzabread* is still in $\exists\ topping.\top^{\mathcal{J},\mathcal{M}'_1,\mathcal{M}_1}$ for all $\mathcal{J} \in \mathcal{M}_1$. From the generality of the sets of first-order models we have chosen, it can be concluded that KB is unsatisfiable.

From this example, it can be seen that epistemic operators in DL can be used to realise integrity constraints as epistemic axioms inside the knowledge base. By identifying known individuals and posing assumptions on them, those epistemic models are ruled out in which the assumptions are violated.

4 Applying Local Closed World Reasoning

The OWA has been criticised in various Semantic Web related settings, such as natural language interfaces [6] or Semantic Web Service policies [11], description [7] and discovery [5]. In this section we show by an example how LCW reasoning can be applied in order to benefit from making conjectures in an open world Semantic Web setting.

In our scenario, the pizza delivery services of Giovanni and Alberto allow to order pizzas via the web. They use the vocabulary from a general pizza ontology O_{Pizza} to describe the pizzas they offer through semantic annotations $O_{Giovanni}$ and $O_{Alberto}$ in OWL as follows.

$$\begin{aligned} O_{Pizza} \supseteq \{ & \exists\, topping.\top \sqsubseteq Pizza,\ Chili \sqsubseteq \neg Mozarella \sqcap \neg Tomato\ , \\ & Vesufo \sqsubseteq SpicyDish \sqcap \forall\, topping.\neg Chili\ , \\ & Margarita \equiv \exists\, topping.Tomato \sqcap \exists\, topping.Mozarella \sqcap \forall\, topping.(Tomato \sqcup Mozarella)\ \} \end{aligned}$$

$$O_{Giovanni} \supseteq \{\ \exists\, topping.Chili(normalChili),\ \exists\, topping.Chili \sqcap \neg SpicyDish(mildChili)\ \}$$

$$O_{Alberto} \supseteq \{\ Margarita(margarita), Vesufo(vesufo)\ \}$$

The concept *SpicyDish* in O_{Pizza} is intended to indicate whether a pizza is spicy or not. However, Giovanni and Alberto do not consequently use this concept to classify all their pizzas – only some are explicitly said to be spicy or non-spicy.

Now consider a Semantic Web agent that is interested in non-spicy pizzas only. Using the OWL concept $\neg SpicyDish$ to query the annotations of Giovanni and Alberto, this agent would only get the pizza *mildChili* as a result. Intuitively, we would like the pizza *margarita* to also be in the result of the query, since as humans we make conjectures such as "the toppings *tomato* and *mozarella* typically don't make a pizza spicy". This more intuitive result can not be achieved by just posing a closed world query, asking for all spicy pizzas and inverting the result by taking all the others. In this case we would, besides the pizzas *margarita* and *mildChili*, also get the pizza *normalChili*, which we would intuitively conjecture to be typically spicy due to its chili topping.

In the following we show how the techniques for realising LCW reasoning from Section 3 can be applied to include such conjectures in the querying process.

Applying Epistemic Queries

Epistemic queries provide a means to encode conjectures, like the ones made above, directly into the query. In our example, the agent could pose an epistemic query, asking for "pizzas that are either non-spicy or not known to be spicy but known to have only non-chili toppings". This would yield the intuitively desired result from the annotations of Giovanni and Alberto as follows.

$$O_{Pizza} \cup O_{Giovanni} \models \neg SpicyDish \sqcup \neg \mathbf{K} SpicyDish \sqcap \mathbf{K}\forall\, topping.\neg Chili(mildChili)$$

$$O_{Pizza} \cup O_{Alberto} \models \neg SpicyDish \sqcup \neg \mathbf{K} SpicyDish \sqcap \mathbf{K}\forall\, topping.\neg Chili(margarita)$$

The epistemic query yields the pizzas *mildChili*, since it is declared as non-spicy, and *margarita*, since it matches our conjecture.

In general, epistemic queries should be used to make conjectures on the side of a Semantic Web agent, in settings where the original ontologies involved shall be leaved untouched. In such a setting each agent can then make its own conjectures.

Applying Default Rules

Default rules provide a means to incorporate conjectures into the domain knowledge. In our example, the designers of the domain ontology O_{Pizza} could decide to make the conjecture "pizzas with chili toppings are typically spicy, whereas pizzas without chili toppings are typically non-spicy" part of the domain knowledge for pizzas by means of the following default rules.

$$\begin{aligned} D_{Pizza} = \{\ & \mathbf{K}\exists\ topping.Chili \sqcap \neg\mathbf{A}\neg SpicyDish \sqsubseteq SpicyDish\ , \\ & \mathbf{K}\forall\ topping.\neg Chili \sqcap \neg\mathbf{A} SpicyDish \sqsubseteq \neg SpicyDish\ \ \} \end{aligned}$$

This allows our agent to draw some additional conclusions as follows.

$$\begin{aligned} & O_{Pizza} \cup D_{Pizza} \cup O_{Giovanni} \cup O_{Alberto}\ \models \\ & \qquad \{\ SpicyDish(normalChili),\ \neg SpicyDish(margarita)\ \} \end{aligned}$$

In particular, this leaves no pizza for which it cannot be concluded whether it is spicy or not. Therefore the agent can now safely use the non-epistemic query concept $\neg SpicyDish$ to ask for all the non-spicy pizzas.

In general, default rules should be used for including commonly agreed conjectures in the domain knowledge. This relieves the Semantic Web agent from the burden of making conjectures itself.

Applying Integrity Constraints

So far, in our scenario, we derived additional conclusions, based on conjectures, to deal with incomplete knowledge about the spiciness of pizzas in ontologies. An alternative would be to not allow such incomplete information about spiciness, and to force pizza delivery services to explicitly classify all their pizzas accordingly. In our example, this can be achieved by including an integrity constraint IC_{Pizza} requiring that any pizza is either assumed to be spicy or assumed to be non-spicy, invalidating knowledge bases with non-classified pizzas.

$$IC_{Pizza} = \{\ \ \mathbf{K}Pizza \sqsubseteq (\mathbf{A}SpicyDish \sqcup \mathbf{A}\neg SpicyDish)\ \ \}$$

Both the pizzas *normalChili* and *margarita* fail to be determined as either spicy or non-spicy, which is reflected by both $O_{Pizza} \cup IC_{Pizza} \cup O_{Giovanni}$ and $O_{Pizza} \cup IC_{Pizza} \cup O_{Alberto}$ being unsatisfiable.

In general, integrity constraints should be used in cases where conjectures cannot be safely made on any side and where modelers should be forced to explicate certain information. Observe, that in OWL there is no way to express such an integrity constraint allowing to detect the improper modelling in Giovanni's and Alberto's ontologies.

5 Summary and Outlook

In this paper we have presented a case for extending OWL with non-monotonic features by means of autoepistemic description logics [3]. In particular, we have shown how local closed world reasoning is realised through the use of the epistemic operators **K** and **A** in formalising default rules, integrity constraints and epistemic querying. Finally, we have demonstrated how such non-monotonic features apply to making common sense conjectures for reasoning in a Semantic Web scenario by extending the popular pizza example from [9].

Although [3] provides a good theoretical foundation, several issues need to be addressed in order to achieve a true non-monotonic extension of OWL. Firstly, in non-monotonic reasoning it is a common practice to assume unique name assumption; however, such an assumption is not employed in OWL. Related to that is the fact that in [3] the authors treat only $\mathcal{ALC}$, which does not require equality reasoning; on the contrary, OWL requires equality reasoning to implement number restrictions. Hence, we shall investigate the possibility of extending ADLs to logics which use equality. Secondly, although [3] presents a tableaux algorithm for reasoning in ADLs, it needs to be clarified whether this algorithm can easily be extended to more expressive DLs like the ones current OWL reasoners can handle. Furthermore, the practicability of such algorithms needs to be tested.

References

1. G. Antoniou. *Nonmonotonic Reasoning*. MIT Press, 1997.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, January 2003.
3. F.M. Donini, D. Nardi, and R. Rosati. Description logics of minimal knowledge and negation as failure. *ACM Transactions on Computational Logic*, 3(2):177–225, 2002.
4. O. Etzioni, K. Golden, and D. Weld. Tractable Closed World Reasoning with Updates. In *Proc. KR-2004*, 1994.
5. S. Grimm, B. Motik, and C. Preist. Variance in e-business service discovery. In *Proc. 1st Intl. Workshop SWS'2004 at ISWC 2004*, November 2004.
6. U. Hustadt. Do we need the closed-world assumption in knowledge representation. In F. Baader, M. Buchheit, M. Jeusfeld, and W. Nutt, editors, *Working Notes of the KI'94 Workshop: Reasoning about Structured Objects: Knowledge Representation Meets Databases (KRDB'94)*, volume D-94-11 of *Document*, pages 24–26. DFKI.
7. H. Lausen, J. de Bruijn, A. Polleres, and D. Fensel. Wsml - a language framework for semantic web services. In *W3C Rules Workshop*, Washington DC, USA, April 2005.
8. P. F. Patel-Schneider, P. Hayes, I. Horrocks, and F. van Harmelen. OWL Web Ontology Language; Semantics and Abstract Syntax, W3C Candidate Recommendation. http://www.w3.org/TR/owl-semantics/, November 2002.
9. A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang, and C. Wroe. Owl pizzas: Common errors & common patterns from practical experience of teaching owl-dl. In *Proc. of the Eleventh Intern. Conf. on World Wide Web*, pages 89–98, 2002.
10. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(81–132):219–221, 1980.
11. V. Kolovski and B. Parsia and Y. Katz and J.Hendler. Representing Web Service Policies in OWL-DL. *To appear in Proc. of the 4th Intern. Semantic Web Conf. (ISWC)*, 2005.