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) [9] 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 Stroganoff just because we did not require the guest to specify whether 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. However, 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 'knowledge' 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, $\mathbf{K}$ and $\mathbf{A}$, reflecting the notions of 'knowledge' and 'assumption'. The following rules define the syntax of the resulting language $\mathcal{ALCK_{NF}}$, 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 $\mathbf{K}$ and $\mathbf{A}$ in the sense of modal logics. The following equations define how the elements of $\mathcal{ALCK_{NF}}$ 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 $\mathcal{ALCK_{NF}}$ 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.

As a special case, a non-epistemic knowledge base KB always has a unique epistemic model $\mathcal{M}(KB)$, which is just the set of all its first-order models [3].

Intuition behind Epistemic Operators

The semantics of both epistemic operators, $\mathbf{K}$ and $\mathbf{A}$, 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 constantly hold in all possible worlds in these sets. The difference between $\mathbf{K}$ and $\mathbf{A}$ 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 $\mathbf{K}$-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 $\mathcal{I}' \in \mathcal{M}(KB) \setminus \mathcal{M}_x$[1]. In this sense $\mathbf{K}$ 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 $\mathbf{A}$. 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 $\mathbf{A}$-operator in $KB_{\mathbf{A}}$ refers to individuals that are *assumed* to be in the extension of C already, and $\mathbf{A}$ can therefore be paraphrased as "assumed". If this assumption is not justified by other facts then the knowledge base becomes unsatisfiable. The $\mathbf{A}$-operator is directly related to the operator not for negation as failure: $\mathbf{A}C$ maps to $\neg\, \mathsf{not}\, C$, which means that any individual assumed to be in C belongs to the complement of those individuals for which there is no evidence to be in C.

[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 $\sigma \subset \Delta^{\mathcal{I}}$

Epistemic Sentences

An *epistemic concept* is a concept that contains epistemic operators. We distinguish cases in which epistemic concepts occur inside a knowledge base from those in which they occur outside only, as e.g. in queries.

Querying a knowledge base KB in general 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. In this sense, an *epistemic query* is an epistemic concept[2] C that is posed as a query to a non-epistemic knowledge base KB. To validate an entailment $\mathit{KB} \models C(\iota)$, the assertion $C(\iota)$ has to be satisfied by epistemic interpretations $(\mathcal{I}, \mathcal{M}, \mathcal{M})$ for every epistemic model $\mathcal{M}$ of KB. However, since here KB is non-epistemic, it is sufficient to consider the set $\mathcal{M}(\mathit{KB})$ of its first-order models.

An *epistemic axiom*, either inclusion or assertion, is an axiom that contains an epistemic concept. If epistemic axioms occur in a knowledge base KB, they determine the epistemic models of KB. As some special cases of epistemic axioms we look at default rules and integrity constraints in Section 3.

3 Local Closed World Reasoning in a Semantic Web Scenario

The OWA has been criticised in various Semantic Web related settings based on DL, such as natural language interfaces [6] or Semantic Web Service policies [7], description [8] and discovery [5]. In this section we show by an example how different forms of LCW reasoning, realised through epistemic operators, can be applied in order to benefit from making common-sense conjectures in an open world Semantic Web setting. We adopt the popular pizza scenario from [10].

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

$$\begin{aligned}
O_{Pizza} \supseteq \{\ & \exists\, \mathit{topping}.\top \sqsubseteq \mathit{Pizza},\ \mathit{Chili} \sqsubseteq \neg \mathit{Mozarella} \sqcap \neg \mathit{Tomato}\ , \\
& \mathit{Vesufo} \sqsubseteq \mathit{SpicyDish} \sqcap \forall\, \mathit{topping}.\neg \mathit{Chili}\ , \\
& \mathit{Margarita} \equiv \exists\, \mathit{topping}.\mathit{Tomato} \sqcap \exists\, \mathit{topping}.\mathit{Mozarella} \sqcap \\
& \qquad\qquad \forall\, \mathit{topping}.(\mathit{Tomato} \sqcup \mathit{Mozarella}) \qquad \} \\
O_{Giovanni} \supseteq \{\ & \exists\, \mathit{topping}.\mathit{Chili}(\mathit{normalChili}), \\
& \exists\, \mathit{topping}.\mathit{Chili} \sqcap \neg \mathit{SpicyDish}(\mathit{mildChili})\ \ \} \\
O_{Alberto} \supseteq \{\ & \mathit{Margarita}(\mathit{margarita}), \mathit{Vesufo}(\mathit{vesufo})\ \ \}
\end{aligned}$$

The ontology O_{Pizza} contains knowledge about the relation between pizzas and their toppings. It defines particular kinds of pizzas, such as Vesufo,

[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.

which is spicy but has only non-chili toppings, or $Margarita$, which has exactly tomato and mozarella as toppings. Giovanni offers two pizzas with chili topping, *normalChili* and *mildChili*, the latter of which is non-spicy. Alberto offers the pizzas $margarita$ and $vesufo$, using the predefined pizza classes from O_{Pizza}.

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. In general, a Semantic Web agent that interprets such annotations and ontologies should not expect to come across situations in which everything is completely specified. It rather gathers pieces of knowledge from different sources and has to draw a conclusion out of these in situations of incomplete information.

In our concrete scenario, consider a Semantic Web agent that is interested in non-spicy pizzas only. Using the OWL-DL concept $\neg$*SpicyDish* to query the annotations of Giovanni and Alberto, this agent would only get the pizza *mildChili* as a result due to the OWA – for the other pizzas there is no evidence to be non-spicy. 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 also not be achieved by just posing a closed world query, asking for all spicy pizzas and then 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. Thus, there is no straightforward way to incorporate the conjectures about the spiciness into the querying by means of OWL-DL concepts.

The epistemic operators introduced in Section 2 can be used as a means to express such conjectures in the knowledge representation formalism, reducing 'don't know' answers that OWL-DL reasoning potentially produces in situations of incomplete knowledge. In the following, we present three different techniques for realising local closed world reasoning by epistemic operators, namely epistemic querying, default rules and integrity constraints, which we apply in our scenario to yield the intuitively desired result when asking for non-spicy pizzas.

3.1 Epistemic Querying

In epistemic queries, as introduced in Section 2, the **K**-operator is used to refer to locally closed off parts of the domain model. They 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". Querying the annotations of Giovanni and Alberto, this would yield the intuitively desired result as follows.

$$Q \equiv \neg \mathit{SpicyDish} \sqcup (\neg \mathbf{K}\mathit{SpicyDish} \sqcap \mathbf{K}\forall\, \mathit{topping}.\neg \mathit{Chili})$$

$$O_{Pizza} \cup O_{Giovanni} \cup O_{Alberto} \models \{Q(\mathit{mildChili}), Q(margarita)\}$$

The epistemic query yields the pizzas *mildChili*, since it is declared as non-spicy, and $margarita$, since it is known to not have any chili topping.

To verify this entailment, we have to check whether the epistemic interpretation $(\mathcal{I}, \mathcal{M}(KB), \mathcal{M}(KB))$ satisfies the above concept assertions for all first-order models $\mathcal{I} \in \mathcal{M}(KB)$, with $KB := O_{Pizza} \cup O_{Giovanni} \cup O_{Alberto}$. The assertion for *mildChili* is satisfied, since this individual is explicitly asserted to the first disjunct $\neg SpicyDish$ of the query in $O_{Giovanni}$. In the second disjunct, the epistemic concept $\mathbf{K}SpicyDish$ is interpreted as the intersection of individuals in *SpicyDish* over all first-order models of $\mathcal{M}(KB)$. This eliminates those individuals which do not always belong to the concept extension for these models, in this case *normalChili*, *mildChili* and *margarita*. The negated expression $\neg\mathbf{K}SpicyDish$ therefore refers to exactly these individuals, which are not known to be spicy dishes. Analogously, for the epistemic concept $\mathbf{K}\forall topping.\neg Chili$ those individuals are eliminated which are not excluded from having chili toppings, leaving only *vesufo* and *margarita*. Hence, the assertion for *margarita* is also satisfied.

In general, epistemic queries can 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 when querying shared ontologies.

3.2 Default Reasoning

A *default rule*, according to [11], 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].

Default rules provide a means to incorporate conjectures into the domain knowledge. In our example scenario, 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.

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

By this they would achieve that pizzas, for which there is no evidence of whether they are spicy or not in O_{Pizza}, are concluded to be spicy or non-spicy in $O_{Pizza} \cup D_{Pizza}$ by default, depending on whether they are known to have chili toppings or non-chili toppings only. In a joint ontology $O_{Pizza} \cup D_{Pizza} \cup O_{Giovanni} \cup O_{Alberto}$ we would intuitively like these default rules to be applied on the pizzas *normalChili* and *margarita*, concluding that *normalChili* is spicy and that *margarita* is not. Contrarily, we would not like the default rules to be applied on the pizzas *mildChili* and *vesufo*, since we already know about their spiciness. In order to verify the appropriate application of the default rules, we will determine the epistemic models of the ontologies involved. For sake of simplicity, we will only consider the knowledge base $KB := O_{Pizza} \cup O_{Alberto}$ together with the default rule D_b .

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

	$\mathcal{I},\mathcal{M}_1,\mathcal{M}_1$	$\mathcal{I},\mathcal{M}'_1,\mathcal{M}_1$	$\mathcal{I},\mathcal{M}_2,\mathcal{M}_2$	$\mathcal{I},\mathcal{M}'_2,\mathcal{M}_2$
$\mathbf{K}\forall\, topping.\neg Chili$	$\{ves, mar\}$	$\{ves, mar\}$	$\{ves, mar\}$	$\{ves, mar\}$
$\mathbf{A} SpicyDish$	$\{ves\}$	$\{ves\}$	$\{ves, mar\}$	$\{ves, mar\}$
$\mathbf{K}\neg SpicyDish$	$\{mar\}$	$\{\}$	$\{\}$	$\{\}$

Table 1. Extensions of epistemic concepts in different epistemic interpretations

To obtain candidates for epistemic models of $KB \cup \{D_b\}$, let $\mathcal{M}_1$ and $\mathcal{M}_2$ be two partitions for all first-order models $\mathcal{M}(KB)$ of KB, such that $\mathcal{M}_1 = \{\mathcal{I} \in \mathcal{M}(KB) : \mathcal{I} \models \neg SpicyDish(margarita)\}$ and $\mathcal{M}_2 = \{\mathcal{I} \in \mathcal{M}(KB) : \mathcal{I} \models SpicyDish(margarita)\}$. 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_b because $margarita$ is in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} \forall\, topping.\neg Chili^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$ and not in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} SpicyDish^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$ but not in $\bigcap_{\mathcal{J}\in\mathcal{M}_{12}} \neg SpicyDish^{\mathcal{J},\mathcal{M}_{12},\mathcal{M}_{12}}$, making the inclusion false. We verify that only $\mathcal{M}_1$ is an epistemic model of $KB \cup \{D_b\}$ using Table 1, which shows the extensions of the epistemic concepts involved in the inclusion from D_b for different epistemic interpretations. The epistemic interpretation $(\mathcal{I},\mathcal{M}_1,\mathcal{M}_1)$ satisfies $KB \cup \{D_b\}$, since the inclusion in D_b is true for both individuals: $margarita$ is in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} \forall\, topping.\neg Chili^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$, not in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} SpicyDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ and in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} \neg SpicyDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$, whereas $vesufo$ is in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} SpicyDish^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$. To check whether $\mathcal{M}_1$ is indeed an epistemic model for $KB \cup \{D_b\}$ 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 $KB \cup \{D_b\}$, since $margarita$ is still not in $\bigcap_{\mathcal{J}\in\mathcal{M}_1} SpicyDish^{\mathcal{J},\mathcal{M}'_1,\mathcal{M}_1}$, as before, but also not in $\bigcap_{\mathcal{J}\in\mathcal{M}'_1} \neg SpicyDish^{\mathcal{J},\mathcal{M}'_1,\mathcal{M}_1}$, contradicting the inclusion.

If we check whether $\mathcal{M}_2$ is also an epistemic model of $KB \cup \{D_b\}$, we observe that $(\mathcal{I},\mathcal{M}_2,\mathcal{M}_2)$ does also satisfy the axioms in $KB \cup \{D_b\}$. 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 $SpicyDish^{\mathcal{J},\mathcal{M}'_2,\mathcal{M}_2}$ for all $\mathcal{J}\in\mathcal{M}_2$.

Having determined $\mathcal{M}_1$ as the only epistemic model of $KB \cup \{D_b\}$, and with a similar reasoning for the pizzas in $O_{Giovanni}$ together with D_a, our agent can draw the following additional conclusions.

$$O_{Pizza} \cup D_{Pizza} \cup O_{Giovanni} \cup O_{Alberto} \models \{\ SpicyDish(normalChili),\ \neg SpicyDish(margarita)\ \}$$

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 retrieve the desired non-spicy pizzas.

In general, default rules can be used for including commonly agreed conjectures in the domain knowledge. This relieves Semantic Web agents from the burden of making such conjectures themselves.

3.3 Constraining Ontologies

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 by means of integrity constraints.

An *integrity constraint* is used to check the state of a knowledge base without deriving new facts – something that cannot be done in OWL-DL. In [3] it has been shown that ADLs are well suited to formalise integrity constraints due to their introspective nature.

In our example, the designers of O_{Pizza} could include an integrity constraint requiring that any pizza is classified as either spicy or non-spicy, invalidating knowledge bases with non-classified pizzas.

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

The integrity constraint in IC_{Pizza} says that any individual that is known to be a pizza can either be assumed to be spicy or assumed to be non-spicy. Recall that such an assumption, expressed through the **A**-operator, requires a justification by other facts in a knowledge base. Both the pizzas *normalChili* and *margarita* fail to be determined as either spicy or non-spicy, which results in both $O_{Pizza} \cup IC_{Pizza} \cup O_{Giovanni}$ and $O_{Pizza} \cup IC_{Pizza} \cup O_{Alberto}$ being unsatisfiable.

To exemplarily verify this unsatisfiability for the knowledge base $\mathit{KB} := O_{Pizza} \cup IC_{Pizza} \cup O_{Alberto}$, consider a set of first-order interpretations $\mathcal{M}_1 = \{\mathcal{I} \in \mathcal{M}(\mathit{KB}) : \mathcal{I} \models \mathit{SpicyDish}(\mathit{margarita})\}$, in which the pizza *margarita* is constantly spicy. The epistemic interpretation $(\mathcal{I}, \mathcal{M}_1, \mathcal{M}_1)$ satisfies KB, since *margarita* is in $\mathit{Pizza}^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ and also in $\mathit{SpicyDish}^{\mathcal{J},\mathcal{M}_1,\mathcal{M}_1}$ 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 *margarita* is still in $\mathit{SpicyDish}^{\mathcal{J},\mathcal{M}_1',\mathcal{M}_1}$ for all $\mathcal{J} \in \mathcal{M}_1$. A similar reasoning can be done for a set of first-order models in which *margarita* is constantly non-spicy, and hence KB is unsatisfiable.

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

4 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]. By extending the popular pizza example from [10], we have shown how the epistemic operators **K** and **A** can be used to realise different forms of local closed world reasoning in a Semantic Web scenario. In particular, we have applied epistemic querying, default rules and integrity constraints. We have demonstrated how a Semantic Web agent can use such non-monotonic features to make common-sense conjectures for reasoning in this scenario.

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 and their optimisations needs to be tested.

While modelling ontologies in OWL-DL is already a complicated task for many users, ADLs make things even more complicated for the modeller. A promising alternative to the free use of epistemic operators would therefore be to employ direct modelling constructs for default rules or integrity constraints. In any case, a more expressive knowledge representation formalism requires additional care to be taken, and new modelling methodologies and patterns need to be introduced in order to handle e.g. situations with conflicting defaults [12].

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 Comput. 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. V. Kolovski, B. Parsia, Y. Katz, and J.Hendler. Representing Web Service Policies in OWL-DL. *To appear in Proc. 4th Intern. Semantic Web Conf. (ISWC)*, 2005.
8. 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.
9. 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.
10. 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.
11. R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(81–132), 1980.
12. R. Reiter and G. Criscuolo. On interacting defaults. In *Proc. IJCAI'87*, pages 94–100, San Francisco, CA, USA, 1987. Morgan Kaufmann Publishers Inc.