Epistemic Querying of OWL Knowledge Bases

Anees ul Mehdi, Sebastian Rudolph, and Stephan Grimm

Institute AIFB, Karlsruhe Institute of Technology, DE
{sebastian.rudolph,anees.mehdi}@kit.edu
Forschungszentrum Informatik Karlsruhe, DE
grimm@fzi.de

Abstract. Epistemic querying extends standard ontology inferencing by allowing for deductive introspection. We propose a technique for epistemic querying of OWL 2 ontologies not featuring nominals and universal roles by a reduction to a series of standard OWL 2 reasoning steps thereby enabling the deployment of off-the-shelf OWL 2 reasoning tools for this task. We prove formal correctness of our method, justify the omission of nominals and universal role, and provide an implementation as well as evaluation results.

1 Introduction

Ontologies play a crucial role in the Semantic Web and the Web Ontology Language (OWL, [7]) is the currently single most important formalism for web-based semantic applications. OWL 2 DL – the most comprehensive version of OWL that still allows for automated reasoning – is based on the description logic (DL) $\mathcal{SROIQ}$ [5]. Querying ontologies by means of checking entailment of axioms or instance retrieval is a crucial and prominent reasoning task in semantic applications. Despite being an expressive formalism, these standard querying capabilities with OWL ontologies lack the ability for introspection (i.e., asking what the knowledge base "knows" *within* the query language). Autoepistemic DLs cope with this problem and have been investigated in the context of OWL and Semantic Web. In particular, they allow for introspection of the knowledge base in the query language by means of epistemic operators, such as the **K**-operator (paraphrased as "known to be") that can be applied to concepts and roles.

The **K**-operator allows for epistemic querying. E.g., in order to formulate queries like "known white wine that is not known to be produced in a French region" we could do an instance retrieval w.r.t. the DL concept

$$\mathbf{K}\, \mathit{WhiteWine} \sqcap \neg\exists \mathbf{K} \mathit{locatedIn}.\{\mathit{FrenchRegion}\}.$$

This can e.g. be used to query for wines that aren't explicitly excluded from being French wines but for which there is also no evidence of being French wines either (neither directly nor indirectly via deduction). For the knowledge base containing

$$\{\mathit{WhiteWine}(\mathit{MountadamRiesling}), \mathit{locatedIn}(\mathit{MountadamRiesling}, \mathit{AustralianRegion})\}$$

the query would yield *MountadamRiesling* as a result, since it is known to be a white wine not known to be produced in a France, while a similar query without epistemic operators would yield an empty result. Hence, in the spirit of nonmonotonicity, more instances can be retrieved (and thus conclusions can been drawn) than with conventional queries in this way. Another typical use case is integrity constraint checking: testing whether the axiom

$$\mathbf{K}\,\mathit{Wine} \sqsubseteq \exists \mathbf{K}\mathit{hasSugar}.\{\mathit{Dry}\} \sqcup \exists \mathbf{K}\mathit{hasSugar}.\{\mathit{OffDry}\} \sqcup \exists \mathbf{K}\mathit{hasSugar}.\{\mathit{Sweet}\}$$

is entailed allows to check whether for every named individual that is known to be a wine it is also known (i.e. it can be logically derived from the ontology) what degree of sugar it has.[1]

However, epistemic operators (or other means for nonmonotonicity) have not found their way into the OWL specification and current reasoners do not support this feature; former research has been focused on extending tableaux algorithms for less expressive formalisms than OWL and have not paced up with the development of OWL reasoners towards optimized tableaux for expressive languages; in particular, some expressive features like nominals require special care when combined with the idea of introspection by epistemic operators.

In this paper, we take a different approach to make epistemic querying possible with OWL ontologies; namely, we reuse existing OWL reasoners in a black box fashion while providing a mechanism for reducing the problem of epistemic querying to standard DL instance retrieval; our approach reduces occurrences of the **K**-operator to introspective look-ups of instances of a concept by calls to a standard DL reasoner, while we keep the number of such calls minimal; we have implemented this approach in form of a reasoner that accepts epistemic queries and operates on non-epistemic OWL ontologies

Our contributions are the following:

[1] Note that this cannot be taken for granted even if $\mathit{Wine} \sqsubseteq \exists \mathit{hasSugar}.\{\mathit{Dry}\} \sqcup \exists \mathit{hasSugar}.\{\mathit{OffDry}\} \sqcup \exists \mathit{hasSugar}.\{\mathit{Sweet}\}$ is stated in (or can be derived from) the ontology.

- We introduce a transformation of epistemic queries to semantically identical non-epistemic queries by making introspective calls to a standard DL reasoner and by propagating the respective answer sets as nominals to the resulting query.
- We prove the correctness of this transformation in the light of some difficulties that occur with the common domain and rigid term assumptions that underly autoepistemic DLs.
- We present an efficient algorithm for implementing the above transformation with a minimal number of calls to a standard DL reasoner for the introspective look-ups of instances.
- Based on this algorithm, we provide a reasoner capable of answering epistemic queries by means of reduction to standard DL reasoning in the framework of the OWL-API extended by constructs for epistemic concepts and roles to be used in epistemic queries. First experiments show that our approach to epistemic querying is practically feasible.

The rest of this paper is structured as follows: Section 2 puts our approach into context with related work. Section 3 introduces the description logic $\mathcal{SROIQ}$ and its extension with the epistemic operator $\mathbf{K}$. In Section 4, we provide the formal justification for our method of reducing $\mathcal{SROIQK}$ axiom entailment from $\mathcal{SRIQ}$ knowledge bases. In Section 5, we describe principle problems arising from allowing the use nominals or universal role in the knowledge base. In Section 6, we discuss the implementation issues and some evaluation results. We conclude in Section 7. For details and proofs we refer to the accompanying technical report[2].

2 Related Work

In the early 80s Hector J. Levesque argued for the need for a richer query language in knowledge formalisms [6]. He describes that the approach to knowledge representation should be functional rather than structural and defends the idea of extending a querying language by the attribute *knows* denoted by **K** (a modality in Modal Logic terminology). In [8], Raymond Reiter makes a similar argument of in-adequacy of the standard first-order language for querying. Nevertheless, he discusses this issue in the context of databases. Similar lines of argumentation can be seen in the DL-community as well [3, 4, 2, 1] where several extensions of DLs have been presented as well as algorithms for deciding the reasoning services

[2] http://www.aifb.kit.edu/images/2/23/EpistemicQueryingTR.pdf

Table 1. Syntax and semantics of role and concept constructors in $\mathcal{SROIQ}$.Thereby a denotes an individual name, R an arbitrary role name and S a simple role name. C and D denote concept expressions.

Name	Syntax	Semantics
inverse role	R^-	$\{\langle x,y\rangle \in \Delta^\mathcal{I} \times \Delta^\mathcal{I} \mid \langle y,x\rangle \in R^\mathcal{I}\}$
universal role	U	$\Delta^\mathcal{I} \times \Delta^\mathcal{I}$
top	$\top$	$\Delta^\mathcal{I}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^\mathcal{I} \setminus C^\mathcal{I}$
conjunction	$C \sqcap D$	$C^\mathcal{I} \cap D^\mathcal{I}$
disjunction	$C \sqcup D$	$C^\mathcal{I} \cup D^\mathcal{I}$
nominals	$\{a\}$	$\{a^\mathcal{I}\}$
univ. restriction	$\forall R.C$	$\{x \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in R^\mathcal{I}$ implies $y \in C^\mathcal{I}\}$
exist. restriction	$\exists R.C$	$\{x \in \Delta^\mathcal{I} \mid$ for some $y \in \Delta^\mathcal{I}$, $\langle x,y\rangle \in R^\mathcal{I}$ and $y \in C^\mathcal{I}\}$
Self concept	$\exists S.\mathsf{Self}$	$\{x \in \Delta^\mathcal{I} \mid \langle x,x\rangle \in S^\mathcal{I}\}$
qualified number	$\leqslant n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \leq n\}$
restriction	$\geqslant n\, S.C$	$\{x \in \Delta^\mathcal{I} \mid \#\{y \in \Delta^\mathcal{I} \mid \langle x,y\rangle \in S^\mathcal{I}$ and $y \in C^\mathcal{I}\} \geq n\}$

in such extensions. The extension of the DL $\mathcal{ALC}$ [9] by the epistemic operator **K** called $\mathcal{ALCK}$, is presented in [3]. A tableau algorithm has been designed for deciding the satisfiability problem. Answering queries in $\mathcal{ALCK}$ put to $\mathcal{ALC}$ knowledge bases is also discussed. In this work we mainly focus on DLs extended with the epistemic operator **K** following notions presented in [3]. However, we consider more expressive DLs rather than just $\mathcal{ALC}$.

3 Preliminaries

We present an introduction to the description logic $\mathcal{SROIQ}$ and its extension with the epistemic operator **K**.

Let N_I, N_C, and N_R be finite, disjoint sets called *individual names*, *concept names* and *role names* respectively, with $N_R = \mathbf{R_s} \uplus \mathbf{R_n}$ called *simple* and *non-simple* roles, respectively. These atomic entities can be used to form complex ones in the usual way (see Table 1).

A $\mathcal{SROIQ}$*-knowledge base* is a tuple $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ where $\mathcal{T}$ is a $\mathcal{SROIQ}$-TBox, $\mathcal{R}$ is a regular $\mathcal{SROIQ}$-role hierarchy[3] and $\mathcal{A}$ is a $\mathcal{SROIQ}$-ABox containing axioms as presented in Table 2.

The semantics of $\mathcal{SROIQ}$ is defined via interpretations $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$ composed of a non-empty set $\Delta^\mathcal{I}$ called the *domain of* $\mathcal{I}$ and a function

[3] We assume the usual regularity assumption for $\mathcal{SROIQ}$, but omit it for space reasons.

Table 2. Syntax and semantics of $\mathcal{SROIQ}$ axioms

Axiom α	$\mathcal{I} \models \alpha$, if	
$R_1 \circ \cdots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \cdots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$	RBox $\mathcal{R}$
$\mathsf{Dis}(S, T)$	$S^{\mathcal{I}} \cap T^{\mathcal{I}} = \emptyset$	
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	TBox $\mathcal{T}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$	ABox $\mathcal{A}$
$R(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$	
$a \doteq b$	$a^{\mathcal{I}} = a^{\mathcal{I}}$	
$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$	

$\cdot^{\mathcal{I}}$ mapping individuals to elements of $\Delta^{\mathcal{I}}$, concepts to subsets of $\Delta^{\mathcal{I}}$ and roles to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. This mapping is extended to complex roles and concepts as in Table 1 and finally used to evaluate axioms (see Table 2). We say $\mathcal{I}$ satisfies a knowledge base $\Sigma = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ (or $\mathcal{I}$ is a model of Σ, written: $\mathcal{I} \models \Sigma$) if it satisfies all axioms of $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{A}$. We say that a knowledge base Σ *entails* an axiom α (written $\Sigma \models \alpha$) if all models of Σ are models of α.

Next, we present the extension of the DL $\mathcal{SROIQ}$ by the epistemic operator K. Let $\mathcal{SROIQK}$ denote the extension of $\mathcal{SROIQ}$ by K, where we allow K to appear in front of concept or role expressions. We call a $\mathcal{SROIQK}$-role an *epistemic role* if $\mathbf{K}$ occurs in it. An epistemic role is *simple* if it is of the form $\mathbf{K}S$ where S is a simple $\mathcal{SROIQ}$-role.

The semantics of $\mathcal{SROIQK}$ is given as *possible world semantics* in terms of *epistemic interpretations*. Thereby the following two central assumptions are made:

1. *Common Domain Assumption*: all interpretations are defined over a fixed infinite domain Δ.
2. *Rigid Term Assumption*: For all interpretations, the mapping from individuals to domains elements is fixed: it is just the identity function.

Definition 1. An *epistemic interpretation* for $\mathcal{SROIQK}$ is a pair $(\mathcal{I}, \mathcal{W})$ where $\mathcal{I}$ is a $\mathcal{SROIQ}$-interpretation and $\mathcal{W}$ is a set of $\mathcal{SROIQ}$-interpretations, where $\mathcal{I}$ and all of $\mathcal{W}$ have the same infinite domain Δ with $N_I \subset \Delta$. The interpretation function $\cdot^{\mathcal{I},\mathcal{W}}$ is then defined as follows:

$$
\begin{aligned}
a^{\mathcal{I},\mathcal{W}} &= a \quad \text{for } a \in N_I \\
X^{\mathcal{I},\mathcal{W}} &= X^{\mathcal{I}} \quad \text{for } A \in N_C \cup N_R \cup \{\top, \bot\} \\
(\mathsf{K}C)^{\mathcal{I},\mathcal{W}} &= \textstyle\bigcap_{\mathcal{J}\in\mathcal{W}}(C^{\mathcal{J},\mathcal{W}}) \qquad (\mathsf{K}R)^{\mathcal{I},\mathcal{W}} = \textstyle\bigcap_{\mathcal{J}\in\mathcal{W}}(R^{\mathcal{J},\mathcal{W}}) \\
(C \sqcap D)^{\mathcal{I},\mathcal{W}} &= C^{\mathcal{I},\mathcal{W}} \cap D^{\mathcal{I},\mathcal{W}} \qquad (C \sqcup D)^{\mathcal{I},\mathcal{W}} = C^{\mathcal{I},\mathcal{W}} \cup D^{\mathcal{I},\mathcal{W}} \\
(\neg C)^{\mathcal{I},\mathcal{W}} &= \Delta \setminus C^{\mathcal{I},\mathcal{W}} \\
(\exists R.\mathsf{Self})^{\mathcal{I},\mathcal{W}} &= \{p \in \Delta \mid (p,p) \in R^{\mathcal{I},\mathcal{W}}\} \\
(\exists R.C)^{\mathcal{I},\mathcal{W}} &= \{p_1 \in \Delta \mid \exists p_2.(p_1,p_2) \in R^{\mathcal{I},\mathcal{W}} \wedge p_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\forall R.C)^{\mathcal{I},\mathcal{W}} &= \{p_1 \in \Delta \mid \forall p_2.(p_1,p_2) \in R^{\mathcal{I},\mathcal{W}} \rightarrow p_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\leqslant nR.C)^{\mathcal{I},\mathcal{W}} &= \{d \mid \#\{e \in C^{\mathcal{I},\mathcal{W}} \mid (d,e) \in R^{\mathcal{I},\mathcal{W}}\} \leq n\} \\
(\geqslant nR.C)^{\mathcal{I},\mathcal{W}} &= \{d \mid \#\{e \in C^{\mathcal{I},\mathcal{W}} \mid (d,e) \in R^{\mathcal{I},\mathcal{W}}\} \geq n\}
\end{aligned}
$$

where C and D are $\mathcal{SROIQK}$-concepts and R is a $\mathcal{SROIQK}$-role. $\diamondsuit$

From the above one can see that $\mathsf{K}C$ is interpreted as the set of objects that are in the interpretation of C under every interpretation in $\mathcal{W}$. Note that the rigid term assumption implies the unique name assumption (UNA) i.e., for any epistemic interpretation $\mathcal{I} \in \mathcal{W}$ and for any two distinct individual names a and b we have that $a^{\mathcal{I}} \neq b^{\mathcal{I}}$.

The notions of GCI, assertion, role hierarchy, ABox, TBox and knowledge base, and their interpretations as defined for $\mathcal{SROIQ}$ can be extended to $\mathcal{SROIQK}$ in the obvious way.

An *epistemic model* for a $\mathcal{SROIQK}$-knowledge base $\Sigma = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ is a *maximal* non-empty set $\mathcal{W}$ of $\mathcal{SROIQ}$-interpretations such that $(\mathcal{I}, \mathcal{W})$ satisfies $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$ for each $\mathcal{I} \in \mathcal{W}$. A $\mathcal{SROIQK}$-knowledge base Σ is said to be *satisfiable* if it has an epistemic model. The knowledge base Σ *(epistemically) entails* an axiom α (written $\Sigma \Vdash\!\!= \alpha$), if for every epistemic model $\mathcal{W}$ of Σ, we have that for every $\mathcal{I} \in \mathcal{W}$, the epistemic interpretation $(\mathcal{I}, \mathcal{W})$ satisfies α. By definition every $\mathcal{SROIQ}$-knowledge base is an $\mathcal{SROIQK}$-knowledge base. Note that a given $\mathcal{SROIQ}$-knowledge base Σ has up to isomorphism only one unique epistemic model which is the set of all models of Σ having infinite domain and satisfying the unique name assumption. We denote this model by $\mathcal{M}(\Sigma)$.

4 Deciding Entailment of Epistemic Axioms

In this section we provide a way for deciding epistemic entailment based on techniques for non-epistemic standard reasoning. More precisely, we consider the problem whether a $\mathcal{SROIQK}$ axiom α is entailed by a $\mathcal{SRIQ}$ knowledge base Σ, where $\mathcal{SRIQ}$ is defined as $\mathcal{SROIQ}$ excluding nominals and the universal role. That is, we distinguish the *querying language* from the *modeling language*. One primary use of the **K** operator that we focus on in this paper is for knowledge base introspection in the query, which justifies to exclude it from the modeling language in exchange for reducibility to standard reasoning. The reasons for disallowing the use of nominals and the universal role will be discussed in Section 5.

The basic, rather straightforward idea to decide entailment of an axiom containing **K** operators is to disassemble the axiom, query for the named individuals contained in extensions for every subexpression preceded by **K**, and use the results to rewrite the axiom into one that is free of **K**s. While we will show that this idea is theoretically and practically feasible, some problems need to be overcome that arise from the definition of epistemic models, in particular the rigid term assumption and the common domain assumption.

As a consequence of the rigid name assumption, every $\mathcal{I} \in \mathcal{M}(\Sigma)$ satisfies the condition that individual names are interpreted by different individuals (this condition per se is commonly referred to as the *unique* name assumption). In order to enforce this behavior (which is not ensured by the non-epistemic standard DL semantics) we have to explicitly axiomatize this condition.

Definition 2. Given a $\mathcal{SRIQ}$ knowledge base Σ, we denote by Σ_{UNA} the knowledge base $\Sigma \cup \{a \not\approx b \mid a, b \in N_I, a \neq b\}$. $\Diamond$

Fact 3. *The set of models of Σ_{UNA} is exactly the set of those models of Σ that satisfy the unique name assumption.*

As another additional constraint on epistemic interpretations, the domain is required to be infinite (imposed by the common domain assumption). However, standard DL reasoning as performed by OWL inference engines adheres to a semantics that allows for both finite and infinite models. Therefore, in order to show that we can use standard inferencing tools as a basis of epistemic reasoning, we have to prove that finite models can be safely dismissed from the consideration, without changing the results. We obtain this result by arguing that for any finite interpretation we find an infinite one which "behaves the same" in terms of satisfaction of axioms and hence will make up for the loss of the former. The following definition and lemma provide a concrete construction for this.

Definition 4. For any $\mathcal{SRIQ}$ interpretation $\mathcal{I}$, the *lifting* of $\mathcal{I}$ to ω is the interpretation $\mathcal{I}_\omega$ defined as follows:

- $\Delta^{\mathcal{I}_\omega} := \Delta^{\mathcal{I}} \times \mathbb{N}$,

- $a^{\mathcal{I}_\omega} := \langle a^{\mathcal{I}}, 0\rangle$ for every $a \in N_I$,
- $A^{\mathcal{I}_\omega} := \{\langle x, i\rangle \mid x \in A^{\mathcal{I}}$ and $i \in \mathbb{N}\}$ for each concept name $A \in N_C$,
- $r^{\mathcal{I}_\omega} := \{(\langle x, i\rangle, \langle x', i\rangle) \mid (x, x') \in r^{\mathcal{I}}$ and $i \in \mathbb{N}\}$ for every role name $r \in N_R$. $\Diamond$

Lemma 5. *Let Σ be a $\mathcal{SRIQ}$ knowledge base. For any interpretation $\mathcal{I}$ we have that*

$$\mathcal{I} \models \Sigma \textit{ if and only if } \mathcal{I}_\omega \models \Sigma.$$

The actual justification for our technique of rewriting axioms containing **K**s into **K**-free ones exploiting intermediate reasoner calls comes from the fact that (except for some remarkable special cases) the semantic extension of expressions proceeded by **K** can only contain named individuals. We prove this by exploiting certain symmetries on the model set $\mathcal{M}(\Sigma)$. Intuitively, one can freely swap or permute anonymous individuals (i.e., domain elements which do not correspond to any individual name) in a model of some knowledge base without losing modelhood, as detailed in the following definition and lemma.

Definition 6. Given an interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$, a set Δ with $N_I \subseteq \Delta$, and a bijection $\varphi : \Delta^{\mathcal{I}} \to \Delta$ with $\varphi(a^{\mathcal{I}}) = a$ for all $a \in N_I$, the *renaming* of $\mathcal{I}$ according to φ, denoted by $\varphi(\mathcal{I})$, is defined as the interpretation $(\Delta, \cdot^{\varphi(\mathcal{I})})$ with:

- $a^{\varphi(\mathcal{I})} = \varphi(a^{\mathcal{I}}) = a$ for every individual name a
- $A^{\varphi(\mathcal{I})} = \{\varphi(z) \mid z \in A^{\mathcal{I}}\}$ for every concept name A
- $P^{\varphi(\mathcal{I})} = \{(\varphi(z), \varphi(w)) \mid (z, w) \in P^{\mathcal{I}}\}$ for every role name P $\Diamond$

Lemma 7. *Let Σ be a $\mathcal{SRIQ}$ knowledge base and let $\mathcal{I}$ be a model of Σ with infinite domain. Then, every renaming $\varphi(\mathcal{I})$ of $\mathcal{I}$ satisfies $\varphi(\mathcal{I}) \in \mathcal{M}(\Sigma)$.*

Proof. By definition, the renaming satisfies the common domain and rigid term assumption. Modelhood w.r.t. Σ immediately follows from the isomorphism lemma of first-order interpretations [10] since $\mathcal{I}$ and $\varphi(\mathcal{I})$ are isomorphic and φ is an isomorphism from $\mathcal{I}$ to $\varphi(\mathcal{I})$. $\square$

This insight can be used to "move" every anonymous individual into the position of another individual which serves as a counterexample for membership in some given concept D, unless the concept is equivalent to $\top$. This allows to prove that $\mathbf{K}D$ contains merely named individuals, given that it is not universal.

Lemma 8. *Let Σ be a $\mathcal{SHIQ}$ knowledge base. For any epistemic concept $C = \mathbf{K}D$ with $\Sigma_{\text{UNA}} \not\models D \equiv \top$ and $x \in \Delta$, we have that $x \in C^{\mathcal{I},\mathcal{M}(\Sigma)}$ iff x is named such that there is an individual $a \in N_I$ with $x = a^{\mathcal{I},\mathcal{M}(\Sigma)}$ and $\Sigma_{\text{UNA}} \models D(a)$.*

A similar property can be proved for the roles as well. Before, we have to take care of the exceptional case of the universal role.

Claim 9. *Let Σ be a knowledge base. For the universal role U we have:*

$$\mathbf{K}U^{\mathcal{I},\mathcal{M}(\Sigma)} = U^{\mathcal{I},\mathcal{M}(\Sigma)}$$

The claim follows trivially as $U^{\mathcal{J}} = \Delta \times \Delta$ for any $\mathcal{J} \in \mathcal{M}(\Sigma)$. This means that $\bigcap_{\mathcal{J} \in \mathcal{M}(\Sigma)} U^{\mathcal{J}} = \Delta \times \Delta$. Thus, as in the case of concepts, whenever an epistemic concept contains a role of the form $\mathbf{K}U$, it will be simply replaced by U. That, for $\mathcal{SRIQ}$ knowledge bases, no other role than U is universal (in all models) is straightforward and can be shown using the construction from Definition 4.

We can now also show that the extension of every role preceded by $\mathbf{K}$ (except for the universal one), consists only of pairs of named individuals.

Lemma 10. *Let Σ be a $\mathcal{SRIQ}$ knowledge base. For any epistemic role $R = \mathbf{K}P$ with $P \neq U$, and $x, y \in \Delta$ we have that $(x, y) \in R^{\mathcal{I},\mathcal{M}(\Sigma)}$ iff there are individuals $a, b \in N_I$ such that $a^{\mathcal{I},\mathcal{M}(\Sigma)} = x$, $b^{\mathcal{I},\mathcal{M}(\Sigma)} = y$ and $\Sigma_{\text{UNA}} \models P(a, b)$.*

Having established the above correspondences, we are able to define a translation procedure that maps (complex) epistemic concept expressions to non-epistemic ones which are equivalent in all models of Σ.

Definition 11. Given a $\mathcal{SRIQ}$ knowledge base Σ, we define the function Φ_Σ mapping $\mathcal{SROIQK}$ concept expressions to $\mathcal{SROIQ}$ concept expressions as follows (where we let $\{\} = \emptyset = \bot$):

$$
\Phi_\Sigma : \begin{cases}
C \mapsto C \quad \text{if } C \text{ is an atomic or one-of concept, } \exists S.\mathsf{Self}, \top \text{ or } \bot; \\
\mathbf{K}D \mapsto \begin{cases} \top & \text{if } \Sigma_{\text{UNA}} \models \Phi_\Sigma(D) \equiv \top \\ \{a \in N_I \mid \Sigma_{\text{UNA}} \models \Phi_\Sigma(D)(a)\} & \text{otherwise} \end{cases} \\
\exists \mathbf{K}S.\mathsf{Self} \mapsto \{a \in N_I \mid \Sigma_{\text{UNA}} \models S(a,a)\} \\
C_1 \sqcap C_2 \mapsto \Phi_\Sigma(C_1) \sqcap \Phi_\Sigma(C_2) \\
C_1 \sqcup C_2 \mapsto \Phi_\Sigma(C_1) \sqcup \Phi_\Sigma(C_2) \\
\neg C \mapsto \neg \Phi_\Sigma(C) \\
\exists R.D \mapsto \exists R.\Phi_\Sigma(D) \quad \text{for non-epistemic role } R \\
\exists \mathbf{K}P.D \mapsto \{a \in N_I \mid \exists b \in N_I.\Sigma_{\text{UNA}} \models P(a,b) \wedge \Sigma_{\text{UNA}} \models \Phi_\Sigma(D)(b)\} \\
\forall R.D \mapsto \forall R.\Phi_\Sigma(D) \quad \text{for non-epistemic role } R; \\
\forall \mathbf{K}P.D \mapsto \neg \Phi_\Sigma(\exists \mathbf{K}P.\neg D) \\
\geqslant nS.D \mapsto \geqslant nS.\Phi_\Sigma(D) \quad \text{for non-epistemic role } S; \\
\geqslant n\mathbf{K}S.D \mapsto \{a \in N_I \mid \#\{b \in N_I.\Sigma_{\text{UNA}} \models \Phi_\Sigma(D)(b) \wedge \Sigma_{\text{UNA}} \models S(a,b)\} \geq n\} \\
\leqslant nS.D \mapsto \leqslant nS.\Phi_\Sigma(D) \quad \text{for non-epistemic role } S; \\
\leqslant n\mathbf{K}S.D \mapsto \neg \Phi_\Sigma(\geqslant (n+1)\mathbf{K}S.D) \\
\Xi \mathbf{K}U.D \mapsto \Xi U.\Phi_\Sigma(D) \quad \text{for } \Xi \in \{\forall, \exists, \geqslant n, \leqslant n\}
\end{cases}
$$

◊

We are now ready to establish the correctness of this translation in terms of (epistemic) entailment. In the following lemma, we show that the extension of a $\mathcal{SROIQK}$-concept and the extension of the $\mathcal{SROIQ}$-concept, obtained using the translation function Φ_Σ, agree under each model of the knowledge base.

Lemma 12. *Let Σ be a $\mathcal{SRIQ}$-knowledge base, x be an element of Δ, and C be a $\mathcal{SROIQK}$ concept. Then for any interpretation $\mathcal{I} \in \mathcal{M}(\Sigma)$, we have that $C^{\mathcal{I},\mathcal{M}(\Sigma)} = (\Phi_\Sigma(C))^{\mathcal{I},\mathcal{M}(\Sigma)}$.*

Moreover Lemma 12 allows to establish the result that the translation function Φ_Σ can be used to reduces the problem of entailment of $\mathcal{SROIQK}$ axioms by $\mathcal{SRIQ}$ knowledge bases to the problem of entailment of $\mathcal{SROIQ}$ axioms, formally put into the following theorem.

Theorem 13. *For a $\mathcal{SRIQ}$ knowledge base Σ, $\mathcal{SROIQK}$-concepts C and D and an individual a the following hold:*

1. *$\Sigma \models\!\!\models C(a)$ exactly if $\Sigma_{\text{UNA}} \models \Phi_\Sigma(C)(a)$.*
2. *$\Sigma \models\!\!\models C \sqsubseteq D$ exactly if $\Sigma_{\text{UNA}} \models \Phi_\Sigma(C) \sqsubseteq \Phi_\Sigma(D)$.*

Proof. For the first case, we see that $\Sigma \models\!\!\models C(a)$ is equivalent to $a^{\mathcal{I},\mathcal{M}(\Sigma)} \in C^{\mathcal{I},\mathcal{M}(\Sigma)}$ which by Lemma 12 is the case exactly if $a^{\mathcal{I},\mathcal{M}(\Sigma)} \in \Phi_\Sigma(C)^{\mathcal{I},\mathcal{M}(\Sigma)}$ for all $\mathcal{I} \in \mathcal{M}(\Sigma)$. Since $\Phi_\Sigma(C)$ does not contain any **K**s, this is equivalent to $a^\mathcal{I} \in \Phi_\Sigma(C)^\mathcal{I}$ and hence to $\mathcal{I} \models \Phi_\Sigma(C)(a)$ for all $\mathcal{I} \in \mathcal{M}(\Sigma)$. Now we can invoke Fact 3 and Lemma 5 to see that this is the case if and only if $\Sigma_{\text{UNA}} \models \Phi_{\Sigma}(C)(a)$. The second case is proven in exactly the same fashion. □

Hence standard DL-reasoners can be used in order to answer epistemic queries. It can be seen from the definition of Φ_{Σ} that deciding epistemic entailment along those lines may require deciding many classical entailment problems and hence involve many calls to the reasoner. Nevertheless, the number of reasoner calls is bounded by the number of **K**s occurring in the query.

5 Semantical Problems Caused by Nominals and the Universal Role

One of the basic assumptions that is made regarding the epistemic interpretations is the *common domain assumption* as mentioned in Section 3. It basically has two parts: all the interpretations considered in an epistemic interpretation share the same fixed domain and the domain is infinite. However, there is no prima facie reason, why the domain that is described by a knowledge base should not be finite, yet finite models are excluded from the consideration entirely. We have shown that this is still tolerable for description logics up to $\mathcal{SRIQ}$ due to the fact that every finite model of a knowledge base gives rise to an infinite one that behaves the same (i.e. the two models cannot be distinguished by means of the underlying logic), as shown in Lemma 5. However, this situation changes once nominals or the universal role are allowed. In fact, the axioms $\top \sqsubseteq \{a, b, c\}$ or $\top \sqsubseteq \leqslant 3U.\top$ have only models with at most three elements. Consequently, according to the prevailing epistemic semantics, these axioms are epistemically unsatisfiable. In general, the coincidence of $\Vdash$ and $\models$ under the UNA which holds for nonepistemic KBs and axioms up to $\mathcal{SRIQ}$ does not hold any more, once nominals or the universal role come into play.

We believe that this phenomenon is not intended but rather a side effect of a semantics crafted for and probed against less expressive description logics, as it contradicts the intuition behind the **K** operator. A refinement of the semantics in order to ensure an intuitive behavior also in the presence of very expressive modeling features is subject of ongoing research.

Algorithm 1 translate (Σ, C) – Translate epistemic query concepts to non-epistemic ones

Require: a $\mathcal{SRIQ}$ knowledge base Σ, an epistemic concept C
Ensure: the return value is the non-epistemic concept $\Phi(C)$

translate $(\Sigma, C = \mathbf{K}D)$
  $\mathcal{X}$:= retrieveInstances $(\Sigma,$ translate $(\Sigma, D))$
  return $\{\ldots, o_i, \ldots\}$, $o_i \in \mathcal{X}$
translate $(\Sigma, C = \exists \mathbf{K}R.D)$
  $\mathcal{X}_D$:= retrieveInstances $(\Sigma,$ translate $(\Sigma, D))$
  $\mathcal{X}$:= retrieveInstances $(\Sigma, \exists R.\{\ldots, o_i, \ldots\})$, $o_i \in \mathcal{X}_D$
  return $\{\ldots, o_i, \ldots\}$, $o_i \in \mathcal{X}$
translate $(\Sigma, C = \forall \mathbf{K}R.D)$
  $\mathcal{X}_{\bar{D}}$:= retrieveInstances $(\Sigma,$ translate $(\Sigma, \neg D))$
  $\mathcal{X}$:= retrieveInstances $(\Sigma, \exists R.\{\ldots, o_i, \ldots\})$, $o_i \in \mathcal{X}_{\bar{D}}$
  return $\neg\{\ldots, o_i, \ldots\}$, $o_i \in \mathcal{X}$
translate $(\Sigma, C = \ldots)$
  ...

6 A System

To check the feasibility of our method in practice, we have implemented a system that we called *EQuIKa*[4] and performed some first experiments for epistemic querying.

Implementation The *EQuIKa* system implements the transformation Φ of an epistemic concept to its non-epistemic version from Definition 11 involving calls to an underlying standard DL reasoner that offers the reasoning task of instance retrieval. To obtain an efficient implementation of Φ it is crucial to keep the number of calls to the DL reasoner minimal. With Algorithm 1 we provide such an efficient implementation, exploiting the fact that extensions of epistemic roles (that occur in role restrictions) only contain known individuals. It shows the transformation in terms of virtual recursive translation functions for the various cases of epistemic concept expressions (see TR for the complete algorithm).

From Algorithm 1, it can be seen that the number of calls to the underlying DL reasoner is at most twice the number of **K**-operators that occur in the original query. This is much better than a naive implementation of Φ according to Definition 11 with iteration over intermediate retrieved individuals.

[4] **E**pistemic **Qu**erying **I**nterfance **Ka**rlsruhe.

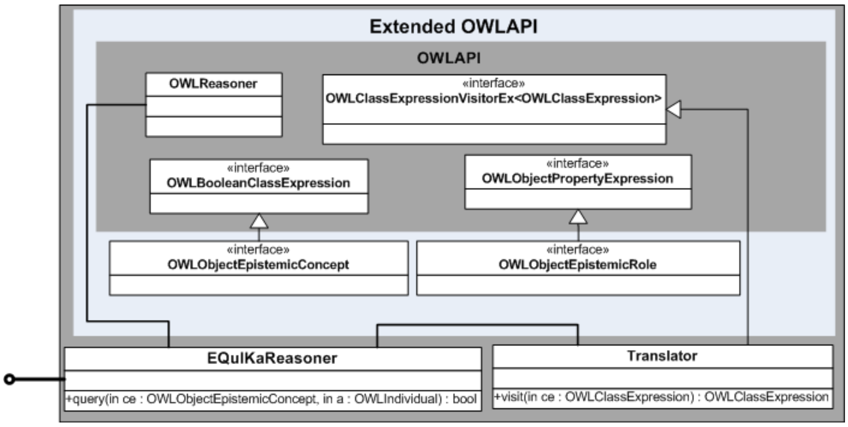

Fig. 1. The *EQuIKa*-system extending the OWL-API

The *EQuIKa* system is implemented on top of the OWL-API[5] extending its classes and interfaces with constructs for epistemic concepts and roles, as shown by the UML class diagram in Figure 1. The new types OWLObjectEpistemicConcept and OWLObjectEpistemicRole are derived from the respective standard types OWLBooleanClassExpression and OWLObjectPropertyExpression to fit the design of the OWL-API.

Using these types, the transformation Φ is implemented in the class Translator following the visitor pattern mechanism built in the OWL-API, which is indicated by the virtual translation functions with different arguments in Algorithm 1. Finally, the EQuIKaReasoner uses both a Translator together with an OWLReasoner to perform epistemic reasoning tasks.

Experiments For the purpose of testing, we consider two versions of the wine ontology[6] with 483 and 1127 instances. As a measure, we consider the instance retrieval time of a concept. This suffices as entailment check can not be harder than instance retrieval. We consider different epistemic concepts. For each such concept C, we consider a non-epistemic concept obtained from C by dropping the $\mathbf{K}$-operators from it (see Table 3). Given a concept C, $\mathsf{t}_{(C)}$ and $|C_i|$ represent the time in seconds required to compute the instances and the number of instances computed for C_i. Finally for an epistemic concept EC_i, $\#\mathsf{Call}_{EC_i}$ represents the number of calls required by EQuIKa to translate it to its non-epistemic equivalent. Ta-

[5] http://owlapi.sourceforge.net/
[6] http://www.w3.org/TR/owl-guide/wine.rdf

Table 3. Concepts used for instance retrieval experiments.

C_1	$\exists hasWineDescriptor.WineDescriptor$
EC_1	$\exists\mathsf{K}hasWineDescriptor.\mathsf{K}\,WineDescriptor$
C_2	$\forall hasWineDescriptor.WineDescriptor$
EC_2	$\forall\mathsf{K}hasWineDescriptor.\mathsf{K}\,WineDescriptor$
C_3	$\exists hasWineDescriptor.WineDescriptor \sqcap \exists madeFromFruit.WineGrape$
EC_3	$\exists\mathsf{K}hasWineDescriptor.\mathsf{K}\,WineDescriptor \sqcap \exists\mathsf{K}madeFromFruit.\mathsf{K}\,WineGrape$
C_4	$WhiteWine \sqcap \neg\exists locatedIn.\{FrenchRegion\}$
EC_4	$\mathsf{K}\,WhiteWine \sqcap \neg\exists\mathsf{K}locatedIn.\{FrenchRegion\}$
C_5	$Wine \sqcap \neg\exists hasSugar.\{Dry\} \sqcap \neg\exists hasSugar.\{OffDry\} \sqcap \neg\exists hasSugar.\{Sweet\}$
EC_5	$\mathsf{K}\,Wine \sqcap \neg\exists\mathsf{K}hasSugar.\{Dry\} \sqcap \neg\exists\mathsf{K}hasSugar.\{OffDry\} \sqcap \neg\mathsf{K}\exists hasSugar.\{Sweet\}$

Table 4. Evaluation

Ontology	Concept	$\mathsf{t}_{(C_i)}$	$\lvert C_i\rvert$	Concept	$\mathsf{t}_{(EC_i)}$	$\lvert EC_i\rvert$	$\#\mathsf{Call}_{EC_i}$
Wine 1	C_1	2.186	159	EC_1	0.53	132	3
	C_2	0.004	483	EC_2	0.037	0	2
	C_3	30.68	159	EC_3	7.60	3	6
	C_4	0.189	0	EC_4	156.92	72	3
	C_5	65.09	80	EC_5	353.29	119	7
Wine 2	C_1	10.15	371	EC_1	1.01	308	3
	C_2	0.10	1127	EC_2	0.038	0	2
	C_3	228.53	371	EC_3	21.86	7	6
	C_4	0.211	0	EC_4	1145.32	168	3
	C_5	311.12	240	EC_5	2526.85	331	7

ble 4 provides our evaluation results for every ontology and every concept under consideration.

One can see from the evaluation results in Table 4 that the time required to compute the number of instances is feasible; it is roughly in the same order of magnitude as for non-epistemic concepts. Note also that the runtime comparison between epistemic concepts EC_i and their non-epistemic counterparts t_{C_i} should be taken with a grain of salt as they are semantically different in general, as also indicated by the fact that there are cases where retrieval for the epistemic concept takes less time than for the non-epistemic version. As a general observation, we noticed that instances retrieval for an epistemic concept where a **K**-operator occurs within the scope of a negation, tends to require much time.

7 Conclusion

We have provided a way to answer epistemic queries to restricted OWL 2 DL ontologies via a reduction to a series of standard reasoning steps. This enables the deployment of today's highly optimized OWL inference engines for this non-standard type of queries. Experiments have shown that the approach is computationally feasible with runtimes in the same order of magnitude as standard (non-epistemic) reasoning tasks.

We identify the following avenues for future research: first and foremost we want to extend the expressivity of the underlying knowledge base to full OWL 2 DL, including nominals and the universal role. To this end, we have to alter the semantics and relinquishing the common domain assumption, to retain an intuitive entailment behavior. Second, we will provide a language extension to OWL 2 for epistemic operators in order to provide for a coherent way of serializing epistemic axioms. Finally we will investigate to which extent the promoted blackbox approach can be extended to the case where the epistemic operator occurs inside the considered knowledge base – note however, that in this case there is no unique epistemic model anymore.

References

1. Francesco Donini, Daniele Nardi, and Riccardo Rosati. Non-first-order features in concept languages. In *In Proceedings of the Fourth Conference of the Italian Association for Artificial Intelligence (AI*IA-95), Lecture Notes In Artificial Intelligence*, pages 91–102. Springer-Verlag, 1995.
2. Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, and Werner Nutt. Adding epistemic operators to concept languages. In Bernhard Nebel, Charles Rich, and William R. Swartout, editors, *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 342–353. Morgan Kaufmann, 1992.
3. Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, Andrea Schaerf, and Werner Nutt. An epistemic operator for description logics. *Artificial Intelligence*, 100(1-2):225–274, 1998.
4. Francesco M. Donini, Daniele Nardi, and Riccardo Rosati. Autoepistemic description logics. In *In Proc. of IJCAI-97*, pages 136–141, 1997.
5. Ian Horrocks, Oliver Kutz, and Ulrike Sattler. The even more irresistible $\mathcal{SROIQ}$. In Patrick Doherty, John Mylopoulos, and Christopher A. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 57–67. AAAI Press, 2006.
6. Hector J. Levesque. Foundations of a functional approach to knowledge representation. *Artif. Intell.*, 23(2):155–212, 1984.
7. W3C OWL Working Group. *OWL 2 Web Ontology Language: Document Overview.* W3C Recommendation, 27 October 2009. Available at http://www.w3.org/TR/owl2-overview/.
8. Raymond Reiter. What should a database know? *J. Log. Program.*, 14(1-2):127–153, 1992.
9. Manfred Schmidt-Schauß and Gert Smolka. Attributive concept descriptions with complements. *Artif. Intell.*, 48(1):1–26, 1991.
10. D. van Dalen. *Logic and Structure.* Springer-Verlag, 1989.