EQuIKa: Epistemic Querying in OWL 2 Ontologies

Anees Mehdi, Sebastian Rudolph, and Jens Wissmann

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

Abstract. Extending ontology querying facilities with epistemic features provides practically useful additional functionalities for ontology management tasks. In this paper, we motivate the benefits of such a formalism for expressing integrity constraints on ontologies.
We present a practical system called EQuIKa capable of epistemic inferencing on OWL 2 DL ontologies. It implements our recently developed reduction to standard reasoning along with several novel performance optimizations. First experiments demonstrate practical feasibility of our system. For convenience of use, we developed querying and constraint checking front-ends for Protégé and the NeOn Toolkit.

1 Introduction

OWL 2 DL is the most expressive yet decidable dialect of the Web Ontology Language (OWL) [10]. It is based on the description logic (DL) $\mathcal{SROIQ}$ [5]. Being a decidable first-order logic fragment, reasoning in $\mathcal{SROIQ}$ is inherently monotonic; addition of information to the ontology never invalidates entailed conclusions. Nevertheless, certain variants of non-monotonicity are sometimes desired in various semantic applications. As one example, [4] discusses the inadequacy of DLs for the task of matching semantic services and presents a solution based on a non-monotonic extension of DLs.

A mild form of non-monotonicity which can still be handled rather conveniently while being sufficient for many purposes arises when only the querying language is endowed with non-monotonic constructors. In the early 1980s, Hector J. Levesque was the first to present the idea of enriching the query language with the epistemic operator $\mathbf{K}$ [6]. Also Raymond Reiter made a similar argument [11] and discussed why a language enriched with an epistemic operator is desirable for integrity constraint (IC) checking. Also in the DL community, extensions by epistemic operators have been considered [1–3, 9]. In epistemic extensions of DLs, the epistemic operator, also called $\mathbf{K}$-operator and paraphrased as "known to be", is allowed to occur in front of concepts and roles to represent the set of individuals (or of pairs of individuals) known by an ontology to possess the property described by the concept or role, respetively. In other words, a query language with **K**-operators allows for ontology introspection.

In this paper, we present a system called *EQuIKa*[1] which allows for epistemic querying and IC checking. It basically implements and significantly improves the algorithm devised in our prior work [8, 7]. For the application in practice, we have also implemented Protégé and NeOn Toolkit plugins that allow a convenient deployment of epistemic querying and IC checking during the ontology design phase. Our paper is organized as follows: In the next section, we motivate the need for an epistemic query language and demonstrate via examples the flexibility we get by epistemic querying approach to IC checking in ontologies. Notions that we use in the sequel are introduced in Section 3, where we also discuss our reduction method for epistemic query answering. Toward a major performance improvement, we developed several optimization rewriting rules in order to ensure the feasibility of EQuIKa in practice, as described in Section 4. Implementation issues and evaluation results of EQuIKa are presented in Section 5 and Section 6. We compare our approach to the one used in Pellet ICV[2] in Section 7. Finally, we conclude our work in Section 8 and identify some lines of future work.

2 Ontological Integrity Constraints

We now discuss diverse specific types of epistemic consequences in order to both make the reader familiar with the expressivity of epistemically extended OWL and argue for its practical relevance.

The **K**-operator allows for querying for known class or role instances. E.g., by performing an instance retrieval for the concept expression

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

on the Wine ontology, we ask for "known white wines that are not known to be produced in French regions". This concept represents all the wines that are explicitly excluded from being French wines but additionally also those for which there is just no evidence of being French wines (neither directly nor indirectly via deduction). For an ontology containing

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

[1] *E*pistemic *Qu*erying *I*nterface *Ka*rlsruhe
[2] http://clarkparsia.com/pellet/icv/

and no additional location information, the query would yield *MountadamRiesling* as a result, since it is known to be a white wine (stated explicitly in the ontology) and not known to have a French region as location. On the other hand, querying for the instances of the concept

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

yields an empty result since – due to the open world assumption – it cannot be excluded that *MountadamRiesling* does not also have a French location.

We will now focus on a specific purpose for which epistemic reasoning seems particularly useful, viz. a specific type of integrity constraints on ontologies. On a logical level, it is helpful to subdivide integrity constraints into two groups, depending whether they address ill-specification (the presence of wrong or contradicting information) or under-specification (the absence of required information). While the first type can be handled well within the classical framework of standard reasoning in OWL (e.g. by detecting inconsistencies), the second type can be modeled satisfactorily only by requiring an extension of the standard OWL language.

We will illustrate our argument using the example ontology $\mathcal{O}$ displayed in Table 1. In words, $\mathcal{O}$ specifies that persons are partitioned in males and females, it enumerates all EU members, specifies EU citizens as citizens of EU member states and provides the information that (at each time point) exactly one member of the EU has the presidency.

Table 1. Example ontology $\mathcal{O}$

$$\begin{array}{rcl}
\mathit{Person} & \sqsubseteq & \mathit{Male} \sqcup \mathit{Female} \\
\mathit{Male} \sqcap \mathit{Female} & \sqsubseteq & \bot \\
\mathit{EUMember} & \equiv & \{\mathit{austria}, \mathit{belgium}, \ldots, \mathit{uk}\} \\
\mathit{EUMember} & \equiv & \exists \mathit{memberOf}.\{\mathit{eu}\} \\
\mathit{EUCitizen} & \equiv & \mathit{Person} \sqcap \exists \mathit{citizenOf}.\mathit{EUMember} \\
\mathit{EUCitizen} & \equiv & \exists \mathit{citizenOf}.\{\mathit{eu}\} \\
\{\mathit{eu}\} & \sqsubseteq & =1.\mathit{hasPresidencyOf}^{-}.\top \\
\mathit{EUPresidency} & \equiv & \exists \mathit{hasPresidencyOf}.\{\mathit{eu}\} \\
\mathit{EUPresidency} & \sqsubseteq & \mathit{EUMember}
\end{array}$$

$$\begin{array}{ll}
\mathit{EUCitizen}(\mathit{denny}) & \neg\mathit{Female}(\mathit{denny}) \\
\mathit{citizenOf}(\mathit{nadeschda}, \mathit{germany}) & \mathit{Female}(\mathit{nadeschda})
\end{array}$$

Integrity constraints related to under-specification often can be seen as a demand for explicitness of information. For instance, disjunctive explicitness for the concepts $C_1, \ldots, C_n$ of concepts requires that whenever a named individual is known to be in one of these classes, it must also be known in which. This can be expressed by the axiom

$$\mathsf{K}(C_1 \sqcup \ldots \sqcup C_n) \sqsubseteq \mathsf{K}C_1 \sqcup \ldots \sqcup \mathsf{K}C_n.$$

For instance, a reasonable integrity constraint for a knowledge base may demand that if an entity is known to have a gender, it should be known which. This can be expressed by the axiom $\mathsf{K}(\mathit{Male} \sqcup \mathit{Female}) \sqsubseteq (\mathsf{K}\mathit{Male} \sqcup \mathsf{K}\mathit{Female})$. That this integrity constraint is satisfied by $\mathcal{O}$ can be checked as follows: the only individuals in $\mathcal{O}$ known to have a gender are *nadeschda* and *denny* (by virtue of being *EUCitizen*s, and thus *Person*s). For *nadeschda*, the gender is explicitly given, whereas for *denny* it can be derived from the given information (which is also OK). Hence the IC is satisfied.

As another case, sometimes, it might be in place to require that for every named individual known to be iin an R-relationship to an individual satisfying some concept C, there must be a named "witness" for this. However, depending on what kind of information precisely is required to be explicit, there are several options for an according integrity constraint exhibiting subtle semantic differences:

$$\mathsf{K}(\exists R.C) \sqsubseteq \exists \mathsf{K}R.\mathsf{K}C \tag{1}$$
$$\mathsf{K}(\exists R.C) \sqsubseteq \exists \mathsf{K}R.C \tag{2}$$
$$\mathsf{K}(\exists R.C) \sqsubseteq \exists R.\mathsf{K}C \tag{3}$$

For illustration, consider the following constraints on $\mathcal{O}$:

$$\mathsf{K}(\exists \mathit{citizenOf}.\mathit{EUMember}) \sqsubseteq \exists \mathsf{K}\mathit{citizenOf}.\mathsf{K}\mathit{EUMember} \tag{IC1}$$
$$\mathsf{K}(\exists \mathit{citizenOf}.\mathit{EUMember}) \sqsubseteq \exists \mathit{citizenOf}.\mathsf{K}\mathit{EUMember} \tag{IC2}$$
$$\mathsf{K}(\exists \mathit{citizenOf}.\mathit{EUMember}) \sqsubseteq \exists \mathsf{K}\mathit{citizenOf}.\mathit{EUMember} \tag{IC3}$$
$$\mathsf{K}(\exists \mathit{memberOf}^-.\mathit{EUPresidency}) \sqsubseteq \exists \mathsf{K}\mathit{memberOf}^-.\mathsf{K}\mathit{EUPresidency} \tag{IC4}$$
$$\mathsf{K}(\exists \mathit{memberOf}^-.\mathit{EUPresidency}) \sqsubseteq \exists \mathit{memberOf}^-.\mathsf{K}\mathit{EUPresidency} \tag{IC5}$$
$$\mathsf{K}(\exists \mathit{memberOf}^-.\mathit{EUPresidency}) \sqsubseteq \exists \mathsf{K}\mathit{memberOf}^-.\mathit{EUPresidency} \tag{IC6}$$

For instance, we find that the integrity constraint (IC1) is violated due to *denny* as it is not known which EU state he is a citizen of. On the other hand, the constraint (IC2) is satisfied, since we know that all *EUMember*s are known by name, hence also the one *denny* is a citizen of must be. Conversely (IC3) is violated, since no guaranteed specific citizenship relationship can be derived from the knowledge base.

Finally, consider the integrity constraint (IC4). Clearly, this is violated, since in our example, no membership of a concrete state in the class *EUPresidency* can be inferred. For the same reason, also the relaxed constraint (IC5) is violated. However, the constraint (IC6) is satisfied since O guarantees that one of the named individuals whose membership in the eu is inferrable must have its presidency.

Arguably, these examples demonstrate that by means of putting or not putting $\mathbf{K}$ in front of concept expression constituents, one can fine-tune integrity constraints toward the actual explicitness requirements. This is particularly useful in scenarios where some information is supposed to remain unknown e.g., due to privacy issues. Moreover, note that current state-of-the-art approaches to ICs in ontologies based on querying cannot tackle these scenarios adequately.

3 Preliminaries

In this section, we present an introduction to the description logic $\mathcal{SROIQ}$ [5] and its extension with the epistemic operator $\mathbf{K}$.

3.1 Description Logics $\mathcal{SROIQ}$

We start by presenting the syntax and semantics of $\mathcal{SROIQ}$. It is an extension of $\mathcal{ALC}$ [12] by inverse roles($\mathcal{I}$), role hierarchies($\mathcal{H}$), nominals($\mathcal{O}$) and qualifying number restrictions($\mathcal{Q}$). Besides it also allows for several role constructs and axioms.

Definition 1. For the signature of $\mathcal{SROIQ}$ we have countably infinite disjoint sets N_C, N_R and N_I of *concept names*, *role names* and *individual names* respectively. Further the set N_R is partitioned into two sets namely, $\mathbf{R_s}$ and $\mathbf{R_n}$ of *simple* and *non-simple* roles respectively. The set $\mathbf{R}$ *of* $\mathcal{SROIQ}$ *roles* is

$$\mathbf{R} := U \mid N_R \mid N_R^-$$

where U is a special role called the *universal role*. Further, we define a function Inv on roles such that $\mathsf{Inv}(R) = R^-$ if R is a role name, $\mathsf{Inv}(R) = S$ if $R = S^-$ and $\mathsf{Inv}(U) := U$.

The set of $\mathcal{SROIQ}$ concepts (or simply concepts) is the smallest set satisfying the following properties:

- every concept name $A \in N_C$ is a concept;
- $\top$(top) and $\bot$ (bottom) are concepts;
- if C, D are concepts, R is a role, S is a simple role, $a_1, \ldots, a_n$ are individual names and n a non-negative integer then following are concepts:

$$
\begin{array}{rl}
\neg C & \text{(negation)} \\
\exists S.\mathsf{Self} & \text{(self)} \\
C \sqcap D & \text{(conjunction)} \\
C \sqcup D & \text{(disjunction)} \\
\forall R.C & \text{(universal quantification)} \\
\exists R.C & \text{(existential quantification)} \\
\leq nS.C & \text{(at least number restriction)} \\
\geq nS.C & \text{(at most number restriction)} \\
\{a_1, \ldots, a_n\} & \text{(nominals / one-of)}
\end{array}
$$

An *RBox axiom* is an expression of one the following forms:

1. $R_1 \circ \cdots \circ R_n \sqsubseteq R$ where $R_1, \ldots, R_n, R \in \mathbf{R}$ and if $n = 1$ and $R_1 \in \mathbf{R_s}$ then $R \notin \mathbf{R_n}$,
2. $\mathsf{Ref}(R)$ (reflexivity), $\mathsf{Tra}(R)$ (transitivity), $\mathsf{Irr}(R)$ (irreflexivity), $\mathsf{Dis}(R, R')$ (role disjointness), $\mathsf{Sym}(R)$ (Symmetry), $\mathsf{Asy}(R)$ (Asymmetry) with $R, R \in \mathbf{R}$.

RBox axioms of the first form i.e., $R_1 \circ \cdots \circ R_n \sqsubseteq R$ are called *role inclusion axioms* (RIAs). An RIA is *complex* if $n > 1$. Whereas the RBox axioms of the second form e.g., $\mathsf{Ref}(R)$, are called *role characteristics*. A $\mathcal{SROIQ}$ RBox $\mathcal{R}$ is a finite set of RBox axioms such that the following conditions are satisfied:[3]

- there is a strict (irreflexive) partial order $\prec$ on $\mathbf{R}$ such that
  - for $R \in \{S, \mathsf{Inv}(S)\}$, we have that $S \prec R$ iff $\mathsf{Inv}(S) \prec R$ and
  - every RIA is of the form $R \circ R \sqsubseteq R$, $\mathsf{Inv(R)} \sqsubseteq R$, $R_1 \circ \ldots R_n \sqsubseteq R$, $R \circ R_1 \circ \cdots \circ R_n \sqsubseteq R$ or $R_1 \circ \cdots \circ R_n \circ R \sqsubseteq R$ where $R, R_1 \ldots, R_n \in \mathbf{R}$ and $R_i \prec R$ for $1 \leq i \leq n$.
- any role characteristic of the form $\mathsf{Irr}(S)$, $\mathsf{Dis}(S, S')$ or $\mathsf{Asy}(S)$ is such that $S, S' \in \mathbf{R_s}$ i.e., we allow only for simple role in these role characteristics.

A *general concept inclusion axiom* (GCI) is an expression of the form $C \sqsubseteq D$, where C and D are $\mathcal{SROIQ}$ concepts. A *TBox* is a finite set of GCIs.

An *ABox axiom* is of the form $C(a)$, $R(a, b)$, $a \doteq b$ or $a \not\doteq b$ for the individual names a and b, a role R and a concept C. A *ABox* is a finite set of ABox axioms.

A $\mathcal{SROIQ}$ *knowledge base* is a tuple $(\mathcal{T}, \mathcal{R}, \mathcal{A})$ where $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$ are $\mathcal{SROIQ}$ TBox, RBox and ABox respectively. ◇

To define the semantics of $\mathcal{SROIQ}$, we introduce the notion of interpretations.

[3] These conditions are enforced to avoid cycles in the RBoxes, which, if not taken care, would lead to undecidability. We usually call an RBox to be *regular* because of the first condition.

Definition 2. A $\mathcal{SROIQ}$ interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is composed of a non-empty set $\Delta^{\mathcal{I}}$, called the *domain of* $\mathcal{I}$ and a *mapping function* $\cdot^{\mathcal{I}}$ such that:

- $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ for every concept name A;
- $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ for every role name $R \in N_R$;
- $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ for every individual name a.

Further the universal role U is interpreted as a total relation on $\Delta^{\mathcal{I}}$ i.e., $U^{\mathcal{I}} = \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$. The bottom concept $\bot$ and top concept $\top$ are interpreted by $\emptyset$ and $\Delta^{\mathcal{I}}$ respectively. Now the mapping $.^{\mathcal{I}}$ is extended to roles and concepts as follows:

$$\begin{aligned}
(R^{-})^{\mathcal{I}} &= \{(x,y) \mid (y,x) \in R^{\mathcal{I}}\} \\
(\neg C)^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}} \\
(\exists S.\mathsf{Self})^{\mathcal{I}} &= \{x \mid (x,x) \in S^{\mathcal{I}}\} \\
(C \sqcap D)^{\mathcal{I}} &= C^{\mathcal{I}} \cap D^{\mathcal{I}} \\
(C \sqcup D)^{\mathcal{I}} &= C^{\mathcal{I}} \cup D^{\mathcal{I}} \\
(\forall R.C)^{\mathcal{I}} &= \{p_1 \in \Delta^{\mathcal{I}} \mid \forall p_2.(p_1,p_2) \in R^{\mathcal{I}} \rightarrow p_2 \in C^{\mathcal{I}}\} \\
(\exists R.C)^{\mathcal{I}} &= \{p_1 \in \Delta^{\mathcal{I}} \mid \exists p_2.(p_1,p_2) \in R^{\mathcal{I}} \wedge p_2 \in C^{\mathcal{I}}\} \\
(\leq nS.C)^{\mathcal{I}} &= \{p_1 \in \Delta^{\mathcal{I}} \mid \#\{p_2 \mid (p_1,p_2) \in S^{\mathcal{I}} \wedge p_2 \in C^{\mathcal{I}}\} \leq n\} \\
(\geq nS.C)^{\mathcal{I}} &= \{p_1 \in \Delta^{\mathcal{I}} \mid \#\{p_2 \mid (p_1,p_2) \in S^{\mathcal{I}} \wedge p_2 \in C^{\mathcal{I}}\} \geq n\} \\
\{a_1,\ldots,a_n\}^{\mathcal{I}} &= \{a_1^{\mathcal{I}},\ldots,a_n^{\mathcal{I}}\} \qquad \diamond
\end{aligned}$$

where C, D are concepts, R, S are roles, n is a non-negative integer and $\#M$ represents the cardinality of the set M.

Given an axiom α (TBox, RBox or ABox axiom), we say the an interpretation $\mathcal{I}$ satisfies α, written $\mathcal{I} \models \alpha$, if it satisfies the condition given in Table 2. Similarly $\mathcal{I}$ satisfies a TBox $\mathcal{T}$, written $\mathcal{I} \models \mathcal{T}$, if it satisfies all the axioms in $\mathcal{T}$. The satisfaction of an RBox and an ABox by an interpretation is defined in the same way. We say $\mathcal{I}$ satisfies a knowledge base $\Sigma = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ if it satisfies $\mathcal{T}$, $\mathcal{R}$ and $\mathcal{A}$. We write $\mathcal{I} \models \Sigma$. We call $\mathcal{I}$ a model of Σ. A knowledge base is said to be *consistent* if it has a model.

We now present the extension of the DL $\mathcal{SROIQ}$ by the epistemic operator $\mathbf{K}$. We call this extension $\mathcal{SROIQK}$.

3.2 K-extensions of $\mathcal{SROIQ}$

The embedding of the epistemic operator $\mathbf{K}$ into the description logic $\mathcal{ALC}$ was first proposed in [1]. The logic obtained is called $\mathcal{ALCK}$. A similar approach

Table 2. Semantics of $\mathcal{SROIQ}$ axioms

Axiom α	$\mathcal{I} \models \alpha$, if
$R_1 \circ \cdots \circ R_n \sqsubseteq R$	$R_1^{\mathcal{I}} \circ \cdots \circ R_n^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
Tra(R)	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
Ref(R)	$(x, x) \in R^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
Irr(S)	$(x, x) \notin S^{\mathcal{I}}$ for all $x \in \Delta^{\mathcal{I}}$
Dis(S,T)	$(x, y) \in S^{\mathcal{I}}$ implies $(x, y) \notin T^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
Sym(S)	$(x, y) \in S^{\mathcal{I}}$ implies $(y, x) \in S^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
Asy(S)	$(x, y) \in S^{\mathcal{I}}$ implies $(y, x) \notin S^{\mathcal{I}}$ for all $x, y \in \Delta^{\mathcal{I}}$
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
$R(a, b)$	$(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$
$a \doteq b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$
$a \neq b$	$a^{\mathcal{I}} \neq b^{\mathcal{I}}$

has been taken in [2], which we follow in this work, although we extend the DL $\mathcal{SRIQ}$ rather than $\mathcal{ALC}$ i.e., we consider $\mathcal{SROIQ}$ as the basis DL and call its **K**-extension $\mathcal{SROIQK}$. In $\mathcal{SROIQK}$ we allow **K** in front of the concepts and roles. In the following we provide the formal syntax and semantics of such language where N_C, N_R, N_I, **R** are as in Definition 1.

Definition 3. A $\mathcal{SROIQK}$ role is defined as follows:

- every $R \in \mathbf{R}$ is a $\mathcal{SROIQK}$ role;
- if R is a $\mathcal{SROIQK}$ role then so are $\mathbf{K}R$ and R^-.

We call a $\mathcal{SROIQK}$ role an *epistemic role* if **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. Now $\mathcal{SROIQK}$ concepts are defined as follows:

- every $\mathcal{SROIQ}$ concept is an $\mathcal{SROIQ}$ concept;
- if C and D are $\mathcal{SROIQK}$ concepts, and S and R are $\mathcal{SROIQK}$ roles with S being simple, then the following are $\mathcal{SROIQK}$ concepts:

$$\mathbf{K}C \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid \leq nS.C \mid \geq nS.C \qquad \diamond$$

The semantics for $\mathcal{SROIQK}$ is based on the possible world semantics. As the traditional semantics leads to unintuitive effects when employed for expressive languages like $\mathcal{SROIQK}$, we adopt our revised semantics introduced in [7]. Unlike the traditional semantics, our semantics enforces neither the common domain assumption(CDA) nor the rigid term assumption(RTA). Hence, the domain we consider in a possible world can be of arbitrary size, (non-empty essentially) composed of arbitrary elements and different individual names can stand for different elements in each possible world i.e., we interpret individual names non-rigidly. As in traditional semantics, by the extension of an epistemic concept $\mathsf{K}C$ we mean all the elements which belong to the extension of C in every possible world, this justifies intersecting of the interpretation of C under each interpretation. Nevertheless, in our semantics as we don't enforce any of the assumptions i.e., CDA or RTA, interpreting $\mathsf{K}C$ in this manner leads to unsatisfiability. For example, in a world, we can interpret C by the set of individuals in way that none of these individual occurs in the extension of C in any other world. As intersecting all these extensions yields to an empty set, therefore, to unsatisfiability of the concept $\mathsf{K}C$. To overcome this problem, we propose the notion of *designators*. The idea here is to extend a standard interpretation $\mathcal{I}$ by a mapping from the set $N_I \cup \mathbb{N}$ to the domain $\Delta^{\mathcal{I}}$ of $\mathcal{I}$. To this end, we define the notion of an extended interpretation.

Definition 4. An extended $\mathcal{SROIQ}$ interpretation $\mathcal{I}$ is a tuple $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}}, \varphi_{\mathcal{I}})$ such that

1. $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is a standard $\mathcal{SROIQ}$ interpretation,
2. $\varphi_{\mathcal{I}}$ is a surjective function $\varphi_{\mathcal{I}} : N_I \cup \mathbb{N} \rightarrow \Delta^{\mathcal{I}}$, such that for all $a \in N_I$ we have that $\varphi_{\mathcal{I}}(a) = a^{\mathcal{I}}$.

We extend the definition of $\varphi_{\mathcal{I}}$ to subsets of $N_I \cup \mathbb{N}$. For a set S, $\varphi_{\mathcal{I}}(S) := \{\varphi_{\mathcal{I}}(t) \mid t \in S\}$. Similarly we extend $\varphi_{\mathcal{I}}$ to order pairs and set of order pairs on $N_I \cup \mathbb{N}$ as follows:

- $\varphi_{\mathcal{I}}((s,t)) := (\varphi_{\mathcal{I}}(s), \varphi_{\mathcal{I}}(t))$ for some ordered-pair $(s,t) \in (N_I \cup \mathbb{N})^2$.
- $\varphi_{\mathcal{I}}(T) := \{\varphi_{\mathcal{I}}((s,t)) \mid (s,t) \in T\}$ for some set $T \subseteq (N_I \cup \mathbb{N})^2$. ◇

We also define the inverse $\varphi_{\mathcal{I}}^{-1}$ of the mapping $\varphi_{\mathcal{I}}$ for an extended interpretation $\mathcal{I}$ as follows:

- $\varphi_{\mathcal{I}}^{-1}(x) := \{t \in N_I \cup \mathbb{N} \mid \varphi_{\mathcal{I}}(t) = x\}$ for every $x \in \Delta^{\mathcal{I}}$.
- $\varphi_{\mathcal{I}}^{-1}(E) := \{\varphi_{\mathcal{I}}^{-1}(x) \mid x \in E\}$ for $E \subseteq \Delta^{\mathcal{I}}$.
- $\varphi_{\mathcal{I}}^{-1}((x,y)) := \varphi_{\mathcal{I}}^{-1}(x) \times \varphi_{\mathcal{I}}^{-1}(y) = \{(x',y') \mid x' \in \varphi_{\mathcal{I}}^{-1}(x) \text{ and } y' \in \varphi_{\mathcal{I}}^{-1}(y)\}$ for any ordered-pair $(x,y) \in \Delta^{\mathcal{I}} \times \Delta^{\varphi_{\mathcal{I}}}$.
- $\varphi_{\mathcal{I}}^{-1}(H) := \bigcup_{(x,y)\in H} \varphi_{\mathcal{I}}^{-1}((x,y))$ for any $H \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Note that for any extended interpretation $\mathcal{I}$, the definition of $\varphi_{\mathcal{I}}$ guarantees that each individual name a is the designator of the interpretation of a under $\mathcal{I}$. For the rest of the elements of $\Delta^{\mathcal{I}}$, we use elements of $\mathbb{N}$ as their designators. Based on the notion of extended interpretation we now provide a new semantics for $\mathcal{SROIQK}$.

Definition 5. (extended semantics for $\mathcal{SROIQK}$)

An *extended epistemic interpretation* for $\mathcal{SROIQK}$ is a pair $(\mathcal{I}, \mathcal{W})$, where $\mathcal{I}$ is an extended $\mathcal{SROIQ}$ interpretation and $\mathcal{W}$ is a set of extended $\mathcal{SROIQ}$

interpretations. Similar to epistemic interpretations, we define an extended interpretation function $\cdot^{\mathcal{I},\mathcal{W}}$:

$$
\begin{array}{rcl}
a^{\mathcal{I},\mathcal{W}} &=& a^{\mathcal{I}} \quad \text{for } a \in N_I \\
A^{\mathcal{I},\mathcal{W}} &=& A^{\mathcal{I}} \quad \text{for } A \in N_C \\
R^{\mathcal{I},\mathcal{W}} &=& R^{\mathcal{I}} \quad \text{for } R \in N_R \\
\top^{\mathcal{I},\mathcal{W}} &=& \Delta^{\mathcal{I}} \quad \text{(the domain of } \mathcal{I}) \\
\bot^{\mathcal{I},\mathcal{W}} &=& \emptyset \\
(C \sqcap D)^{\mathcal{I},\mathcal{W}} &=& C^{\mathcal{I},\mathcal{W}} \cap D^{\mathcal{I},\mathcal{W}} \\
(C \sqcup D)^{\mathcal{I},\mathcal{W}} &=& C^{\mathcal{I},\mathcal{W}} \cup D^{\mathcal{I},\mathcal{W}} \\
(\neg C)^{\mathcal{I},\mathcal{W}} &=& \Delta^{\mathcal{I}} \setminus C^{\mathcal{I},\mathcal{W}} \\
(\forall R.C)^{\mathcal{I},\mathcal{W}} &=& \{p_1 \in \Delta^{\mathcal{I}} \mid \forall p_2.(p_1, p_2) \in R^{\mathcal{I},\mathcal{W}} \to p_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\exists R.C)^{\mathcal{I},\mathcal{W}} &=& \{p_1 \in \Delta^{\mathcal{I}} \mid \exists p_2.(p_1, p_2) \in R^{\mathcal{I},\mathcal{W}} \wedge p_2 \in C^{\mathcal{I},\mathcal{W}}\} \\
(\leq nR.C)^{\mathcal{I},\mathcal{W}} &=& \{d \mid \#\{e \in C^{\mathcal{I},\mathcal{W}} \mid (d, e) \in R^{\mathcal{I},\mathcal{W}}\} \leq n\} \\
(\geq nR.C)^{\mathcal{I},\mathcal{W}} &=& \{d \mid \#\{e \in C^{\mathcal{I},\mathcal{W}} \mid (d, e) \in R^{\mathcal{I},\mathcal{W}}\} \geq n\} \\
(\mathbf{K}C)^{\mathcal{I},\mathcal{W}} &=& \varphi_{\mathcal{I}}\left(\bigcap_{\mathcal{J}\in\mathcal{W}} \varphi_{\mathcal{J}}^{-1}\left(C^{\mathcal{J},\mathcal{W}}\right)\right) \\
(\mathbf{K}R)^{\mathcal{I},\mathcal{W}} &=& \varphi_{\mathcal{I}}\left(\bigcap_{\mathcal{J}\in\mathcal{W}} \varphi_{\mathcal{J}}^{-1}\left(R^{\mathcal{J},\mathcal{W}}\right)\right)
\end{array}
$$

For an epistemic role $(\mathbf{K}R)^-$, we set $[(\mathbf{K}R)^-]^{\mathcal{J},\mathcal{W}} := (\mathbf{K}R^-)^{\mathcal{J},\mathcal{W}}$. Note that $\mathbf{KK}R$ and $\mathbf{K}R$ are interpreted identically under an epistemic interpretation. Hence, it suffices to consider only epistemic roles of the form $\mathbf{K}R$, with R being non-epistemic. ◇

The semantics of GCI, assertion, role hierarchy, ABox, TBox, RBox and knowledge base under an extended epistemic interpretation can be defined in a straight forward way like in Definition 2. Here, instead $\models$ as the symbol of the satisfaction relation, we use the symbol $\Vdash$. We now introduce the notion of an extended epistemic model of a knowledge base.

Definition 6. An *extended epistemic model* of a $\mathcal{SROIQK}$ knowledge base $\Psi = (\mathcal{T}, \mathcal{R}, \mathcal{A})$ is a *maximal* non-empty set $\mathcal{W}$ of extended $\mathcal{SROIQ}$ 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 Ψ is *satisfiable* (under the extended semantics) if it has an extended epistemic model. Similarly the knowledge base Ψ *entails* an axiom α, written $\Psi \Vdash \alpha$, if for every extended epistemic model $\mathcal{W}$ of Ψ, we have that for every $\mathcal{I} \in \mathcal{W}$, the extended epistemic interpretation $(\mathcal{I}, \mathcal{W})$ satisfies α. Like in case of the current semantics, a standard DL-knowledge base Σ, one without any occurence of $\mathbf{K}$, admits a unique extended epistemic model, which is the set of all models of Σ extended by all possible surjective mappings that map individuals names and elements of $\mathbb{N}$ to the elements of their domain. We denote this model by $\mathcal{M}(\Sigma)$.

For a detailed discussion on the extended semantics we recommend the interestred readers to [7], where a method of translating epistemic concept expressions into equivalent **K**-free ones is also presented. Since EQuIKa implements this method, we recall the translation function in the following. Note that the translation itself requires to check entailment of (**K**-free) axioms, hence it is not strictly syntactical and depends on the underlying knowledge base.

Definition 7. Given a $\mathcal{SROIQ}$ knowledge base Σ, we define a function $\tilde{\Phi}_\Sigma$ mapping $\mathcal{SROIQK}$ concept expressions to $\mathcal{SROIQ}$ concept expressions (where we let $\{\} = \emptyset = \bot$):

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

$\diamond$

EQuIKa implements this translation function. In the next section we discuss several optimization rules that we devised in order to ensure the run-time feasibility of EQuIKa.

4 Optimization

A naive implementation of the translation function presented in Definition 7 does not need to be feasible in practice, in particular when dealing with ontologies containing relatively large number of individuals. We came up with several optimization rules. As we will see in the next section, these rules speed up the computation time of EQuIKa in retrieving instances of an epistemic concept. Basically, every rule checks the structure of a given epistemic concept and reduces either the number of $\mathbf{K}$'s occurring in the concept or the number of calls to the core reasoner during the translation of the concept such that the correctness of the answers is preserved. For the proof of the correctness of the answers via translation based upon the optimization rules, we show that for an epistemic concept C, the extensions of the translation of C coincide when the translation is done with and without the optimization rules.

In the following we discuss each of these rules and provide the proof of their correctness, where by $\hat{\Phi}_\Sigma$ we mean the translation function based on the optimization rules.

– ***Rule 1 (Nominals):*** For individual names $a_1, \ldots, a_n$ we have

$$\hat{\Phi}_\Sigma(\mathbf{K}\{a_1, \ldots, a_n\}) \mapsto \{a_1, \ldots, a_n\}$$

Proof. For the left to right direction, let $x \in \mathbf{K}\{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$. By semantics, therefore,

$$x \in \varphi_\mathcal{I}(\bigcap_{\mathcal{J} \in \mathcal{M}(\Sigma)} \varphi_\mathcal{J}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{J}))$$

In other words, $x \in \varphi_\mathcal{I}(\varphi_\mathcal{J}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{J}))$ for each $\mathcal{J} \in \mathcal{M}(\Sigma)$. In particular, $x \in \varphi_\mathcal{I}(\varphi_\mathcal{I}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{I}))$. Since $\varphi_\mathcal{I}$ is a surjective mapping, we get that $x \in \{a_1, \ldots, a_n\}^\mathcal{I} = \{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$.
For the right to left direction, let $x \in \{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$ and suppose that $x \notin \mathbf{K}\{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$. Note that by definition of $\varphi_\mathcal{J}$, we have that $a_i \in {\varphi_\mathcal{J}}^{-1}(a_i^\mathcal{J})$ for each $\mathcal{J} \in \mathcal{M}(\Sigma)$ and $1 \leq i \leq n$. It means that $\{a_1, \ldots, a_n\} \subseteq {\varphi_\mathcal{J}}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{J})$ for each $\mathcal{J} \in \mathcal{M}(\Sigma)$ i.e.,

$$\{a_1, \ldots, a_n\} \subseteq \bigcap_{\mathcal{J} \in \mathcal{M}(\Sigma)} {\varphi_\mathcal{J}}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{J})$$

Applying $\varphi_\mathcal{I}$ we get

$$\varphi_\mathcal{I}(\{a_1, \ldots, a_n\}) \subseteq \varphi_\mathcal{I}(\bigcap_{\mathcal{J} \in \mathcal{M}(\Sigma)} {\varphi_\mathcal{J}}^{-1}(\{a_1, \ldots, a_n\}^\mathcal{J}))$$

In other words,

$$\varphi_\mathcal{I}(\{a_1, \ldots, a_n\}) \subseteq \mathbf{K}\{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$$

By assumption, since $x \notin \mathbf{K}\{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$, consequently we get that $x \notin \varphi_\mathcal{I}(\{a_1, \ldots, a_n\}) = \{\varphi_\mathcal{I}(a_1), \ldots, \varphi_\mathcal{I}(a_n)\}$. By definition of $\varphi_\mathcal{I}$ we have that

$\varphi_{\mathcal{I}}(a) = a^{\mathcal{I}}$ for each $a \in N_I$. Therefore we get that $x \notin \{{a_1}^{\mathcal{I}}, \ldots, {a_n}^{\mathcal{I}}\} = \{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$, which is a contradiction. Hence, $x \in \mathbf{K}\{a_1, \ldots, a_n\}^{\mathcal{I},\mathcal{M}(\Sigma)}$ must hold.

– ***Rule 2 (Conjunction):*** Let $C_1, \ldots, C_n$ be concepts where each C_i is either an epistemic concept of the form $\mathbf{K}D$ for some concept D or a one-of concept for $1 \leq i \leq n$. By C'_i we denote the concept obtained from C_i by dropping $\mathbf{K}$ or $C'_i = C_i$ otherwise. Then,

$$\hat{\Phi}_{\Sigma}(C_1 \sqcap, \ldots, \sqcap C_n) \mapsto \mathbf{K}(C'_1 \sqcap, \ldots, \sqcap C'_n)$$

Proof. For a one-of concept $\{a_1, \ldots, a_k\}$, it follows from the proof of Rule 1 that $\{a_1, \ldots, a_k\}$ is equivalent to $\mathbf{K}\{a_1, \ldots, a_k\}$. Hence, we assume that each concept in $C_1 \sqcap \cdots \sqcap C_n$ is of the form $\mathbf{K}D$ for some concept D. Now
$x \in [\mathbf{K}D_1 \sqcap \ldots \mathbf{K}D_n]^{\mathcal{I},\mathcal{M}(\Sigma)}$
$\Leftrightarrow x \in (\mathbf{K}D_1^{\mathcal{I},\mathcal{M}(\Sigma)} \cap \cdots \cap \mathbf{K}D_n^{\mathcal{I},\mathcal{M}(\Sigma)})$
$\Leftrightarrow x \in \varphi_{\mathcal{I}}(\bigcap_{\mathcal{J}\in\mathcal{M}(\Sigma)} \varphi_{\mathcal{J}}^{-1}(D_1^{\mathcal{J}})) \cap \cdots \cap \varphi_{\mathcal{I}}(\bigcap_{\mathcal{J}\in\mathcal{M}(\Sigma)} \varphi_{\mathcal{J}}^{-1}(D_n^{\mathcal{J}}))$
$\Leftrightarrow x \in \varphi_{\mathcal{I}}\Big(\bigcap_{\mathcal{J}\in\mathcal{M}(\Sigma)} (\varphi_{\mathcal{J}}^{-1}(D_1^{\mathcal{J}}) \cap \cdots \cap \varphi_{\mathcal{J}}^{-1}(D_n^{\mathcal{J}}))\Big)$
$\Leftrightarrow x \in \varphi_{\mathcal{I}}\Big(\bigcap_{\mathcal{J}\in\mathcal{M}(\Sigma)} \varphi_{\mathcal{J}}^{-1}((D_1 \sqcap \cdots \sqcap D_n)^{\mathcal{J}})\Big)$
$\Leftrightarrow x \in [\mathbf{K}(D_1 \sqcap \cdots \sqcap D_n)]^{\mathcal{I},\mathcal{M}(\Sigma)}$ □

– ***Rule 3 (Existential Quantification):***
By Definition 7, for a concept $\exists\mathbf{K}R.D$ we get that

$$\exists\mathbf{K}P.D \mapsto \begin{cases} \bigsqcup_{a\in N_I}\{a\} \sqcap \exists P.(\{b \in N_I \mid \Sigma \models P(a,b) \sqcap \tilde{\Phi}_{\Sigma}(D)\}) \\ \sqcup\, \exists P.(\{b \in N_I \mid \Sigma \models \top \sqsubseteq \exists P.\{b\}\} \sqcap \tilde{\Phi}_{\Sigma}(D)) \\ \sqcup\, \{a \in N_I \mid \Sigma \models \top \sqsubseteq \exists P^{-}\{a\}\} \sqcap \exists P.\tilde{\Phi}_{\Sigma}(D) \\ \sqcup \begin{cases} \tilde{\Phi}_{\Sigma}(D) & \text{if } \Sigma \models \top \sqsubseteq \exists P.\mathsf{Self} \\ \bot & \text{otherwise} \end{cases} \end{cases}$$

Suppose $\mathsf{Inst}(D)$ returns the set of instances of a concept D and let $\mathsf{OUT}_P = \mathsf{Inst}(\exists P.\tilde{\Phi}(D))$ and $\mathsf{IN}_P = \mathsf{Inst}(\exists P^{-}.\top)$. Now Rule 3 is as follows

$$\hat{\Phi}_{\Sigma}(\exists\mathbf{K}P.D) \mapsto \begin{cases} \bigsqcup_{a\in\mathsf{OUT}_P}\{a\} \sqcap \exists P.(\{b \in \mathsf{IN}_P \mid \Sigma \models P(a,b) \sqcap \tilde{\Phi}_{\Sigma}(D)\}) \\ \sqcup\, \exists P.(\{b \in \mathsf{IN}_P \mid \Sigma \models \top \sqsubseteq \exists P.\{b\}\} \sqcap \tilde{\Phi}_{\Sigma}(D)) \\ \sqcup\, \{a \in \mathsf{OUT}_P \mid \Sigma \models \top \sqsubseteq \exists P^{-}\{a\}\} \sqcap \exists P.\tilde{\Phi}_{\Sigma}(D) \\ \sqcup \begin{cases} \tilde{\Phi}_{\Sigma}(D) & \text{if } \Sigma \models \top \sqsubseteq \exists P.\mathsf{Self} \\ \bot & \text{otherwise} \end{cases} \end{cases}$$

Note that without the optimization, translating a concept of the form $\exists\mathbf{K}P.D$ requires at least $|N_I| \times |N_I| + 2|N_I| + 1+$ (No. of calls needed to translate D) calls to the core reasoner. With the optimization Rule 3, such a translation reduces the number of calls when $|\mathsf{Inst}(\exists P.\tilde{\Phi}(D))| < |N_I|$ or $|\mathsf{Inst}(\exists P^{-}.\top)| < N_I$.

Proof. We introduce the following abbreviations,

- $C_1 = \bigsqcup_{a \in N_I} \{a\} \sqcap \exists P.(\{b \in N_I \mid \Sigma \models P(a,b) \sqcap \tilde{\Phi}_\Sigma(D)\})$
- $C'_1 = \bigsqcup_{a \in \mathsf{OUT}_P} \{a\} \sqcap \exists P.(\{b \in \mathsf{IN}_P \mid \Sigma \models P(a,b) \sqcap \tilde{\Phi}_\Sigma(D)\})$
- $C_2 = \exists P.(\{b \in N_I \mid \Sigma \models \top \sqsubseteq \exists P.\{b\}\} \sqcap \tilde{\Phi}_\Sigma(D))$
- $C'_2 = \exists P.(\{b \in \mathsf{IN}_P \mid \Sigma \models \top \sqsubseteq \exists P.\{b\}\} \sqcap \tilde{\Phi}_\Sigma(D))$
- $C_3 = \{a \in N_I \mid \Sigma \models \top \sqsubseteq \exists P^-\{a\}\} \sqcap \exists P.\tilde{\Phi}_\Sigma(D)$
- $C'_3 = \{a \in \mathsf{OUT}_P \mid \Sigma \models \top \sqsubseteq \exists P^-\{a\}\} \sqcap \exists P.\tilde{\Phi}_\Sigma(D)$

We first show that for an extended interpretation $\mathcal{I} \in \mathcal{M}(\Sigma)$ and $x \in \Delta^\mathcal{I}$, $x \in C_1^{\mathcal{I},\mathcal{M}(\Sigma)}$ iff $x \in C_1'^{\mathcal{I},\mathcal{M}(\Sigma)}$. For this note that if $b' \in \mathsf{Inst}(\{b \in N_I | \Sigma \models P(a', b)\})$ for individuals a' and b' then it immediately follows that $b' \in \mathsf{Inst}(\{b \in \mathsf{IN}_P | \Sigma \models (a', b)\})$ and vice versa. Now for $x \in \Delta^\mathcal{I}$, $x \in C_1^{\mathcal{I},\mathcal{M}(\Sigma)}$ if and only if there is an individual $a' \in N_I$ such that $x = a'^{\mathcal{I},\mathcal{M}(\Sigma)}$ and $x \in \exists P.(\{b \in N_I \mid \Sigma \models P(a', b) \sqcap \tilde{\Phi}_\Sigma(D)\})^{\mathcal{I},\mathcal{M}(\Sigma)}$. This is the case if and only if $x \in \exists P.(\{b \in \mathsf{IN}_P \mid \Sigma \models P(a', b) \sqcap \tilde{\Phi}_\Sigma(D)\})^{\mathcal{I},\mathcal{M}(\Sigma)}$ which is equivalent to $x \in C_1'^{\mathcal{I},\mathcal{M}(\Sigma)}$.
In the same way, we can prove that $x \in C_2^{\mathcal{I},\mathcal{M}(\Sigma)}$ if and only if $x \in C_2'^{\mathcal{I},\mathcal{M}(\Sigma)}$. Similarly, $x \in C_3^{\mathcal{I},\mathcal{M}(\Sigma)}$ if and only if $x \in C_3'^{\mathcal{I},\mathfrak{M}(\Sigma)}$. Consequently, this establishes the proof of the correctness of Rule 3. □

– ***Rule 4 (Number Restriction):***
Similar to Rule 3, $\hat{\Phi}_\Sigma$ maps the concept $\geq n\mathbf{K}S.D$ to

$$\begin{cases} \bigsqcup_{a \in \mathsf{OUT}_{P,n}} \{a\} \sqcap \geqslant nS.(\{b \in \mathsf{IN}_{P,n} \mid \Sigma \models S(a,b) \sqcap \tilde{\Phi}_\Sigma(D)\}) \\ \sqcup \{a \in \mathsf{OUT}_{P,n} \mid \Sigma \models \top \sqsubseteq \exists S^-.\{a\}\} \sqcap \geqslant nS.\tilde{\Phi}_\Sigma(D) \\ \sqcup \geqslant nS.(\{b \in \mathsf{IN}_{P,n} \mid \Sigma \models \top \sqsubseteq \exists S.\{b\}\} \sqcap \tilde{\Phi}_\Sigma(D)) \\ \sqcup \begin{cases} \geqslant (n-1)S.(\{b \in \mathsf{IN}_{P,n-1} \mid \Sigma \models \top \sqsubseteq \exists S.\{b\}\} \sqcap \tilde{\Phi}_\Sigma(D)) \sqcap \\ \tilde{\Phi}_\Sigma(D) \sqcap \neg\{a \mid a \in N_I\} & \text{if } \Sigma \models \top \sqsubseteq \exists S.\mathsf{Self} \\ \bot & \text{otherwise} \end{cases} \end{cases}$$

where $\mathsf{OUT}_{P,n} = \geq nP.\tilde{\Phi}(D)$ and $\mathsf{IN}_{P,n} = \geq n.P^-.\top$.

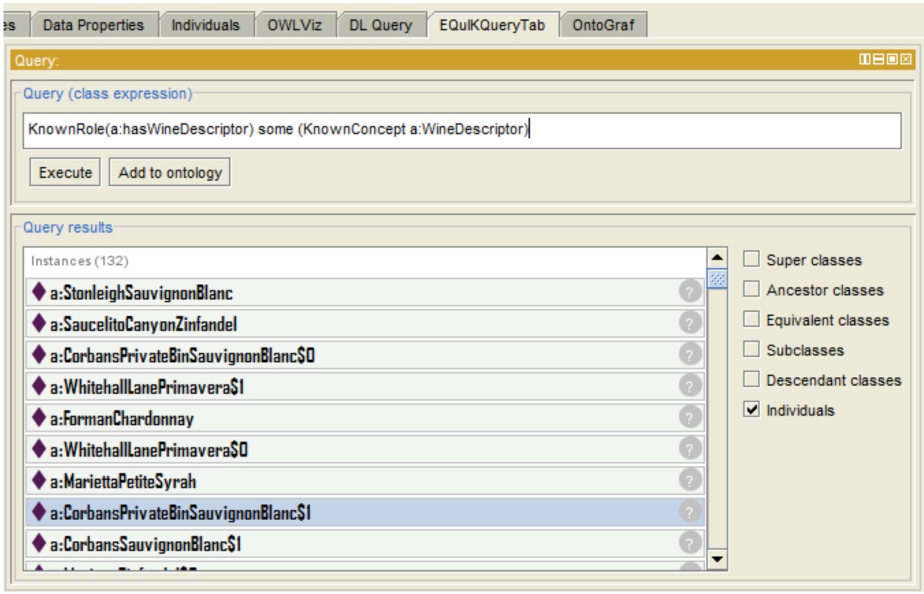
Proof. Similar to Rule 3. □

These rules suffice in the sense that for most of the remaining constructs, we use their dual which correspond to one of the rules discussed above. After showing that the rules preserve the semantics, we need to show that they indeed improve the run time of EQuIKa. For this we performed several experiments, which are discussed in Section 6.

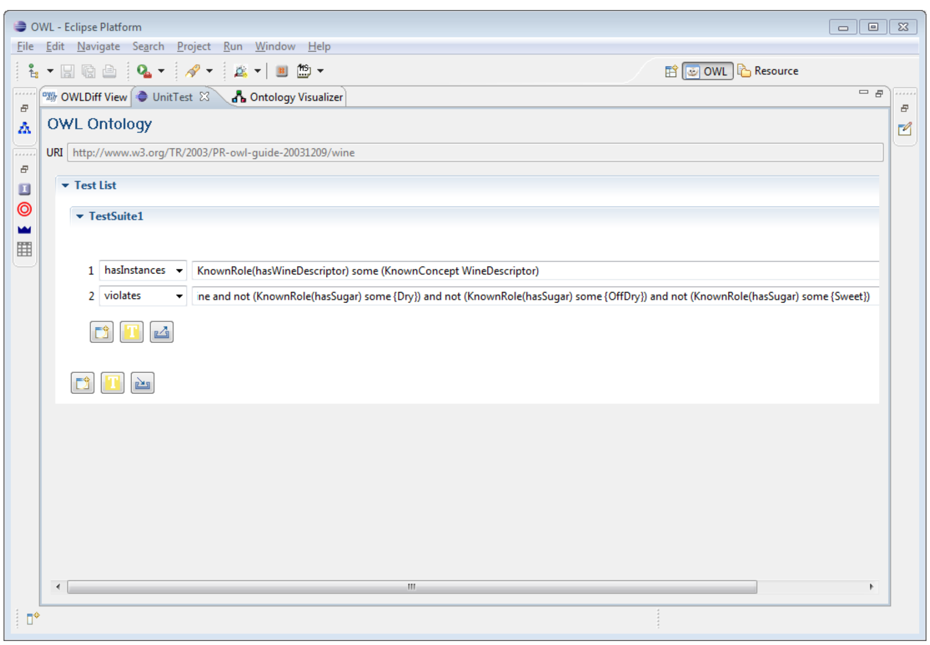
5 Implementation

The EQuIKa system is implemented on top of the OWL-API.[4] It can be used as an API as well as within Protégé or NeOn Toolkit. The following considerations and design decisions underly our implementation:

[4] http://owlapi.sourceforge.net/



(a) Epistemic Querying in Protégé



(b) Integrity Constraint Checking within the NeOn Toolkit

Fig. 1. EQuIKa integration in ontology development tools.

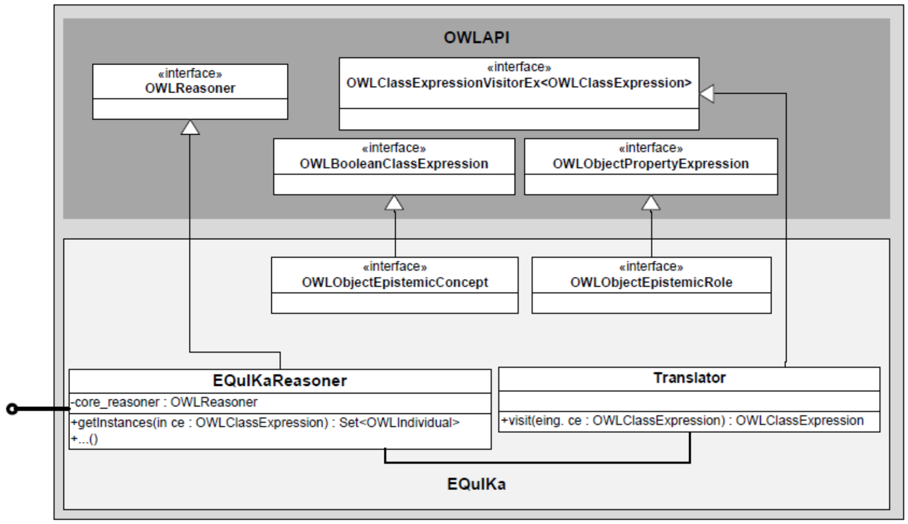


Fig. 2. The *EQuIKa*-system extending the OWL-API

- Since the standard OWL-API does not support epistemic constructs, we extended several classes of the API. The **K**-operator syntactically behaves similar like the *complement construct* (¬) for concepts and like the *inverse role construct* for roles. We therefore followed the same implementation patterns.
- For parsing we created an EpistemicSyntaxParser based on the ManchesterOWLSyntaxOntologyParser. The **K**-operator is expressed by the token `KnownConcept` for concepts and by the token `KnownRole` for the roles.
- We implemented the translation function in a recursive fashion. For this, we implemented a visitor pattern by extending the OWLClassExpressionVisitor class in order to handle the epistemic operator.
- In order to support epistemic querying within the Protégé editor, we implemented an additional tab based on the DL Query tab. Figure 1(a) shows a snapshot of epistemic querying in Protégé.
- In order to support epistemic querying within the NeOn Toolkit, we extended the unit testing component of the ontology evolution plugin CHRONOS[5]. Figure 1(b) shows a snapshot of constraint checking in NeOn.

The class diagram for EQuIKa is displayed in Figure 2. The new types OWLObjectEpistemicConcept and OWLObjectEpistemicRole are derived from

[5] http://chronos-update.fzi.de

Table 3. Concepts used for instance retrieval experiments.

EC_1	$\exists\mathbf{K}hasWineDescriptor.\mathbf{K}WineDescriptor$
EC_2	$\exists\mathbf{K}hasWineDescriptor.\mathbf{K}WineDescriptor \sqcap \exists\mathbf{K}madeFromFruit.\mathbf{K}WineGrape$
EC_3	$\mathbf{K}RoseWine$
EC_4	$\mathbf{K}RoseWine \sqcap \mathbf{K}WhiteWine$
EC_5	$\mathbf{K}RoseWine \sqcap \mathbf{K}WhiteWine \sqcap \{r_1, \ldots, r_{108}\}$
EC_6	$\mathbf{K}Wine \sqcap \neg\exists\mathbf{K}hasSugar.\{Dry\} \sqcap \neg\exists\mathbf{K}hasSugar.\{OffDry\} \sqcap \neg\exists\mathbf{K}hasSugar.\{Sweet\}$

the respective standard types OWLBooleanClassExpression and OWLObjectPropertyExpression to fit the design of the OWL-API. As our translation method depends on intermediate calls to a standard reasoner, the class EQuIKaReasoner implements the OWLReasoner interface. As already mentioned, EQuIKa translates an epistemic concept into a **K**-free one in a recursive fashion using the class Translator that implements the OWLClassExpressionVisitor.

Since Protégé and NeOn can utilize any reasoner that implements the OWLReasoner interface, EQuIkaReasoner can be easily integrated. Last but not least, EQuIKa has been shared on googlecode for testing purposes.[6] The plugin is provided as jar file[7] that can be installed via the Protégé 4.1 plugin folder.

6 Evaluation

For the purpose of evaluation, we performed several experiments with the following setup:

- We used an IBM Thinkpad T60 dual core, with 2 GHz per core, Windows 7 (32-bit) as the operating system, and a total of 2 GB memory.
- For benchmark tests, we used a populated version of the Wine ontology[8], that contains 483 individuals and uses most of the OWL 2 DL constructs. These ontologies can be downloaded along with EQuIKa.
- To evaluate the performance of EQuIKa, we constructed several epistemic concepts and translated them into **K**-free ones. These concepts are given in Table 3 where $r_1, \ldots, r_{108}$ are individuals representing wine regions in the ontologies.

To the best of our knowledge, EQuIKa is the only reasoner of its nature for epistemic query answering, such that there is no other existing reasoner with

6 http://code.google.com/p/epistemicdl/
7 https://epistemicdl.googlecode.com/svn/EpistemicQueryTab/equika.protege.querytab.jar
8 https://code.google.com/p/epistemicdl/source/browse/trunk/EQuIK/wine_1.owl

these capabilities against which we could compare EQuIKa's performance. To give an impression about the runtime behavior, we performed two kind of experiments and as a measure, we consider the time required to translate the epistemic concepts (given in Table 3) to **K**-free equivalent ones and the instance retrieval time of the translated concept. In the first series of experiments, we evaluated the

Table 4. Results of instance retrieval experiments.

Concept	EQuIKa-N			EQuIKa-O		
	T_{trans}	T_{inst}	$\#_{inst}$	T_{trans}	T_{inst}	$\#_{inst}$
EC_1	4	192.7	132	21	97.8	132
EC_2	9	198.9	3	3	37.5	3
EC_3	110	110.1	3	26	26.5	3
EC_4	203	211.7	0	122	122.1	0
EC_5	206	400.6	0	121	121.9	0
EC_6	13	–	–	0.5	487.3	119

benefit of the optimization rules introduced in Section 4. We implemented two versions of EQuIKa; a naive one called EQuIKa-N implementing the translation function of Definition 7 as is and an optimized one called EQuIKa-O where the optimization rules were used. The corresponding results are shown in Table 4 where T_{trans}, T_{inst} and $\#_{inst}$ represent the translation time, instance retrieval time and the number of instances respectively. One can see that T_{inst} for EQuIKa-O is far less than for EQuIKa-N. In particular for concept $EC6$, EQuIKa-N did not responded for almost an hour and we stopped it, whereas EQuIKa-O translated $EC6$ and retrieved its instances in few seconds. This shows that the optimization rules introduced are of high importance toward the feasibility of EQuIKa in practice.

In the second series of experiments, we evaluated the computation time of EQuIKa-O in general to provide an impression of how the cost of epistemic querying relates to standard reasoning tasks. For this purpose, we consider non-epistemic concepts $C1, \ldots, C_6$ where each C_i is obtained by dropping **K** in EC_i for $1 \leq i \leq 6$. Note that an epistemic concept EC_i and the corresponding C_i are semantically different concepts. Table 5 shows the results of our experiments. It can be seen that even when comparing to the **K**-free counterpart of the epistemic concepts, the computation time of EQuIKa-O is roughly in the same order of magnitude. This indicates that an explosion of reasoning runtime which often occurs when nonmonotonic features are added to DLs can be avoided in our case.

Table 5. Evaluation epistemic vs. standard instance retrieval

Concept	$t_{(C_i)}$	$\|C_i\|$	Concept	$t_{T(EC_i)}$	$t_{(EC_i)}$	$\|EC_i\|$
C_1	2.18	159	EC_1	20	95.7	132
C_2	41.9	159	EC_2	3	36.5	3
C_3	10.7	3	EC_3	10	10.8	3
C_4	2.68	0	EC_4	2	2.9	0
C_5	0.2	0	EC_5	2	2.9	0
C_6	61.1	80	EC_5	0.5	487.3	119

7 Related Work

In [13], another approach to IC checking is proposed based on the notion of the so-called *Minimal Equality* models, where ICs are OWL axioms interpreted under *IC-interpretations*. According to this approach, an ontology Σ satisfies a set of constraints C if and only if $\Sigma \models_{IC} \alpha$ for each $\alpha \in C$ i.e., $\mathcal{I}, Mod_{ME}(\Sigma) \models \alpha$ for all $\mathcal{I} \in Mod_{ME}(\Sigma)$, where $Mod_{ME}(\Sigma)$ is the set of all minimal equality models of Σ (see [13] for details). Comparing to our semantics, note that for each $\mathcal{I} \in Mod_{ME}(\Sigma)$, by definition, $\mathcal{M}(\Sigma)$ (c.f. Section 3) contains all the extended interpretations obtained by extending $\mathcal{I}$ with all possible surjective mappings from $N_I \cup \mathbb{N}$ to $\Delta^{\mathcal{I}}$. This allows us to define a translation function that converts every IC $\alpha \in C$ into a $\mathcal{SROIQK}$ axiom $\alpha_{\mathbf{K}}$ by placing the $\mathbf{K}$-operator in front of every occurrence of concept names and role names in α, such that α is satisfied by Σ, i.e., $\Sigma \models_{IC} \alpha$ if and only if $\Sigma \Vdash \alpha_{\mathbf{K}}$.

Thus our formalism is more general in the sense that IC as in [13] can be expressed via $\mathcal{SROIQK}$ axioms, meanwhile we allow for non-epistemic (standard) concepts and roles in epistemic queries. This gives us the added value of tuning ICs per requirement as discussed in Section 2. Another feature of our semantics is that we do not enforce the unique name assumption(UNA) rather we have weak UNA in the sense as discussed in [13]. The enforcement of UNA and less expressiveness in traditional epistemic extension of DL (e.g., [2]), is a serious limitation for using such formalisms in practice as pointed out in [13]. This, indeed is not the case for our approach thanks to our more flexible revised semantics.

Despite the restricted expressiveness, we would have liked to compare Pellet ICV – a system based on the approach of [13] – to our system EQuIKa. Unfortunately, details of the tests conducted for Pellet ICV were not available at the time of compiling this report.

8 Conclusion

In this paper, we have motivated the importance of epistemic querying of OWL ontologies for purposes like ontology introspection, IC checking etc. We have presented a system, called EQuIKa, implemented in conformance with the common OWL interfaces such that any off-the-shelf reasoner can be used as its backbone. To support convenient deployment of our tool in the course of the ontology development process, our system also features a user front-end realized as a plugin for the Protégé and NeOn Toolkit ontology editors.

EQuIKa is based on a reduction of epistemic queries to standard reasoning. In order to assure its practical feasibility, we have presented and implemented several optimization rules leading to a speed-up of EQuIKa by one to several orders of magnitude. We performed several experiments checking the computation time of EQuIKa and also evaluated EQuIKa against the standard reasoning task of instance retrieval. We found that EQuIKa performs in the same scale as the standard counter part, witnessing that epistemic querying can be efficiently realized in practice.

Avenues for future research are manifold: we will carry out a more extensive evaluation of our system with data stemming from ontology design scenarios from industry projects' use cases. These experiments will provide a clearer view on which – already implemented or still to be defined – optimizations and heuristics in EQuIKa pay off in practice. On the theoretical side, we want to investigate the practical benefits of epistemic constructors as part of the ontology modeling language and try to extent our framework to this case. This is clearly a non-trivial task, since favorable model-theoretic properties get lost.

References

1. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A., Nutt, W.: Adding epistemic operators to concept languages. In: Bernhard Nebel, Charles Rich, W.R.S. (ed.) Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning (KR'92). pp. 342–353. Morgan Kaufmann (1992)
2. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A., Nutt, W.: An epistemic operator for description logics. Artificial Intelligence 100(1-2), 225–274 (1998)
3. Donini, F.M., Nardi, D., Rosati, R.: Autoepistemic description logics. In: Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97). pp. 136–141. Morgan Kaufmann (1997)
4. Grimm, S., Motik, B., Preist, C.: Matching semantic service descriptions with local closed-world reasoning. In: ESWC. pp. 575–589 (2006)
5. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Doherty, P., Mylopoulos, J., Welty, C.A. (eds.) Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06). pp. 57–67. AAAI Press (2006)
6. Levesque, H.J.: Foundations of a functional approach to knowledge representation. Artificial Intelligence 23(2), 155–212 (1984)

7. Mehdi, A., Rudolph, S.: Revisiting semantics for epistemic extensions of description logics. In: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence (AAAI-11). AAAI Press (2011)
8. Mehdi, A., Rudolph, S., Grimm, S.: Epistemic querying of owl knowledge bases. In: Antoniou, G., Grobelnik, M., Simperl, E.P.B., Parsia, B., Plexousakis, D., Leenheer, P.D., Pan, J.Z. (eds.) Proceedings of the 8th Extended Semantic Web Conference (ESWC11). pp. 397–409 (2011)
9. Motik, B., Rosati, R.: Reconciling description logics and rules. J. ACM 57(5) (2010)
10. OWL Working Group, W.: OWL 2 Web Ontology Language: Document Overview. W3C Recommendation (2009), available at http://www.w3.org/TR/owl2-overview/
11. Reiter, R.: What should a database know? Journal of Logic Programming 14(1-2), 127–153 (1992)
12. Schmidt-Schauß, M., Smolka, G.: Attributive concept descriptions with complements. Artificial Intelligence 48(1), 1–26 (1991)
13. Tao, J., Sirin, E., Bao, J., McGuinness, D.L.: Integrity constraints in owl. In: Fox, M., Poole, D. (eds.) AAAI. AAAI Press (2010)