Import-by-Query: Ontology Reasoning under Access Limitations

Bernardo Cuenca Grau, Boris Motik, and **Yevgeny Kazakov**
Computing Laboratory
University of Oxford, UK

Abstract

To enable ontology reuse, the Web Ontology Language (OWL) allows an ontology $\mathcal{K}_v$ to *import* an ontology $\mathcal{K}_h$. To reason with such a $\mathcal{K}_v$, a reasoner needs physical access to the axioms of $\mathcal{K}_h$. For copyright and/or privacy reasons, however, the authors of $\mathcal{K}_h$ might not want to publish the axioms of $\mathcal{K}_h$; instead, they might prefer to provide an *oracle* that can answer a (limited) set of queries over $\mathcal{K}_h$, thus allowing $\mathcal{K}_v$ to import $\mathcal{K}_h$ "by query." In this paper, we study *import-by-query* algorithms, which can answer questions about $\mathcal{K}_v \cup \mathcal{K}_h$ by accessing only $\mathcal{K}_v$ and the oracle. We show that no such algorithm exists in general, and present restrictions under which importing by query becomes feasible.

1 Introduction

The Web Ontology Language (OWL) and its revision OWL 2 are widely used ontology languages whose formal underpinnings are provided by description logics (DLs) [Baader *et al.*, 2007]—a family of knowledge representation formalisms with well-understood formal properties. Ontologies are used, for example, in several countries to describe electronic patient records (EPR). In such a system, patients' data typically involves ontological descriptions of human anatomy, medical conditions, drugs and treatments, and so on. The latter domains have already been described in well-established *reference ontologies* such SNOMED-CT and GALEN. In order to save resources, increase interoperability between applications, and rely on experts' knowledge, an EPR application should preferably reuse these reference ontologies.

For example, assume that some reference ontology $\mathcal{K}_h$ describes concepts such as the "ventricular septum defect." An EPR application might reuse the concepts and roles from $\mathcal{K}_h$ to define its own ontology $\mathcal{K}_v$ of concepts such as "patients having a ventricular septum defect." It is generally accepted that ontology reuse should be modular—that is, the axioms of $\mathcal{K}_v$ should not affect the meaning of the symbols reused from $\mathcal{K}_h$ [Lutz *et al.*, 2007; Cuenca Grau *et al.*, 2008].

To enable reuse, OWL allows $\mathcal{K}_v$ to *import* $\mathcal{K}_h$. OWL reasoners deal with imports by internally merging the axioms of the two ontologies; thus, to process $\mathcal{K}_v \cup \mathcal{K}_h$, an EPR application would require physical access to the axioms of $\mathcal{K}_h$. The vendor of $\mathcal{K}_h$, however, might be reluctant to distribute the axioms of $\mathcal{K}_h$, as doing this might allow the competitors to plagiarize $\mathcal{K}_h$. Moreover, $\mathcal{K}_h$ might contain information that is sensitive from a privacy point of view and should not be shared. Finally, the vendor of $\mathcal{K}_h$ might impose different costs for reusing parts of $\mathcal{K}_h$. To reflect this situation, we say that $\mathcal{K}_h$ is *hidden* and, by analogy, $\mathcal{K}_v$ is *visible*.

This problem could be addressed if $\mathcal{K}_h$ were made accessible via an *oracle* (i.e., a limited query interface), thus allowing $\mathcal{K}_v$ to import $\mathcal{K}_h$ "by query." In this paper, we study *import-by-query* algorithms, which can answer questions about $\mathcal{K}_v \cup \mathcal{K}_h$ by accessing only $\mathcal{K}_v$ and the oracle. We focus on schema reasoning problems, such as concept subsumption and satisfiability, which are useful during ontology development; this is in contrast to the information integration [Lenzerini, 2002] and peer-to-peer [Calvanese *et al.*, 2004] scenarios, which focus on the reuse of data.

We proceed as follows. In Section 3 we formalize the import-by-query problem and fix the appropriate query language. Then, in Section 4 we show that no import-by-query algorithm exists in general even if $\mathcal{K}_v$ and $\mathcal{K}_h$ are expressed in the light-weight description logic $\mathcal{EL}$ [Baader *et al.*, 2005]. In Section 5, we present such an algorithm for the case when $\mathcal{K}_v$ reuses only atomic concepts from $\mathcal{K}_h$, and this is done in a modular way. Under certain assumptions, our algorithm is worst-case optimal; however, it is unlikely to be suitable for practice. Therefore, for the case when $\mathcal{K}_h$ is expressed in a Horn DL [Hustadt *et al.*, 2005], we present a practical algorithm that extends the state-of-the-art tableaux algorithms [Kutz *et al.*, 2006]. Finally, in Section 6 we extend our results to the case when $\mathcal{K}_v$ also reuses roles from $\mathcal{K}_h$, but this is done in a syntactically restricted way. Our results may also increase the performance of reasoning: if $\mathcal{K}_v$ is non-Horn but $\mathcal{K}_h$ is, then $\mathcal{K}_v \cup \mathcal{K}_h$ can be reasoned with by applying a general-purpose tableau algorithm only to $\mathcal{K}_v$ and using a more efficient algorithm for $\mathcal{K}_h$.

2 Preliminaries

The formal underpinnings of OWL 2 are provided by the DL $\mathcal{SROIQ}$ [Kutz *et al.*, 2006]. The syntax of $\mathcal{SROIQ}$ is defined w.r.t. a *signature* Σ, which is the union of disjoint countable sets of *atomic concepts*, *atomic roles*, and *individuals*. A *role* is either an atomic role or an *inverse role* R^- for R an atomic role. For R and R_i roles, a *role inclusion axiom* has

Table 1: Model-Theoretic Semantics of $\mathcal{SROIQ}$

Interpretation of Roles
$(R^-)^I = \{\langle y, x\rangle \mid \langle x, y\rangle \in R^I\}$
Interpretation of Concepts
$\top^I = \triangle^I$
$\{a\}^I = \{a^I\}$
$(C_1 \sqcap C_2)^I = C_1^I \cap C_2^I$
$(\exists R.\mathsf{Self})^I = \{x \mid \langle x, x\rangle \in R^I\}$
$(\exists R.C)^I = \{x \mid \exists y : \langle x, y\rangle \in R^I \wedge y \in C^I\}$
$(\geqslant n\, R.C)^I = \{x \mid \sharp\{y \mid \langle x, y\rangle \in R^I \wedge y \in C^I\} \geq n\}$
Satisfaction of Axioms in an Interpretation
$I \models C \sqsubseteq D$ iff $C^I \subseteq D^I$
$I \models R_1 \ldots R_n \sqsubseteq R$ iff $R_1^I \circ \ldots \circ R_n^I \subseteq R^I$
$I \models \mathsf{Dis}(R_1, R_2)$ iff $R_1^I \cap R_2^I = \emptyset$
$I \models C(a)$ iff $a^I \in C^I$
$I \models R(a, b)$ iff $\langle a^I, b^I\rangle \in R^I$
$I \models a \not\approx b$ iff $a^I \neq b^I$

the form $R_1 \ldots R_n \sqsubseteq R$, and a *role disjointness axiom* has the form $\mathsf{Dis}(R_1, R_2)$. The set of *concepts* is the smallest set containing $\top$, A, $\{a\}$, $\neg C$, $C_1 \sqcap C_2$, $\exists R.C$, $\exists R.\mathsf{Self}$, and $\geqslant n\, R.C$, for A an atomic concept, a an individual, C, C_1, and C_2 concepts, R a role, and n a nonnegative integer. Concepts of the form $\{a\}$ are called *nominals*. Furthermore, $\bot$ is an abbreviation for $\neg\top$, $C_1 \sqcup C_2$ for $\neg(\neg C_1 \sqcap \neg C_2)$, $\forall R.C$ for $\neg(\exists R.\neg C)$, and $\leqslant n\, R.C$ for $\neg(\geqslant n{+}1\, R.C)$. A *concept inclusion axiom* has the form $C_1 \sqsubseteq C_2$ for C_1 and C_2 concepts, and a *concept equivalence* $C_1 \equiv C_2$ is an abbreviation for $C_1 \sqsubseteq C_2$ and $C_2 \sqsubseteq C_1$. A TBox $\mathcal{T}$ is a finite set of concept inclusion, role inclusion, and role disjointness axioms. An *assertion* has the form $C(a)$, $R(a, b)$, or $a \not\approx b$, for C a concept, R a role, and a and b individuals. An ABox $\mathcal{A}$ is a finite set of assertions. A $\mathcal{SROIQ}$ knowledge base is a pair $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ where $\mathcal{T}$ is a TBox and $\mathcal{A}$ is an ABox. By a suitable syntactic test, certain roles in $\mathcal{K}$ can be identified as being *simple*. To ensure decidability of reasoning, the role axioms in $\mathcal{T}$ must satisfy a syntactic restriction which we omit for brevity, and simple roles must not occur in $\geqslant n\, R.C$, $\exists R.\mathsf{Self}$, and role disjointness axioms. The definition of $\mathcal{SROIQ}$ by [Kutz *et al.*, 2006] provides other constructs, all of which are expressible by the ones presented above.

A *interpretation* $I = (\triangle^I, \cdot^I)$ consists of a nonempty *domain* set $\triangle^I$ and a function $\cdot^I$ that assigns an object $a^I \in \triangle^I$ to each individual a, a set $A^I \subseteq \triangle^I$ to each atomic concept A, and a relation $R^I \subseteq \triangle^I \times \triangle^I$ to each atomic role R. Table 1 defines the extension of $\cdot^I$ to roles and concepts, and the satisfaction of axioms in I. An interpretation I is a *model* of $\mathcal{K}$, written $I \models \mathcal{K}$, if I satisfies all axioms in $\mathcal{K}$; if such I exists, then $\mathcal{K}$ is *satisfiable*. A concept C is *satisfiable* w.r.t. $\mathcal{K}$ if a model I of $\mathcal{K}$ exists such that $C^I \neq \emptyset$. A nonempty set of interpretations S is *compatible* if for each $I_1, I_2 \in S$ we have $\triangle^{I_1} = \triangle^{I_2}$ and $a^{I_1} = a^{I_2}$ for each individual a; the *intersection* of such S is defined in the obvious way.

$\mathcal{SRIQ}$ is obtained from $\mathcal{SROIQ}$ by disallowing nominals. $\mathcal{EL}$ [Baader *et al.*, 2005] supports only concepts of the form $\top$, $\bot$, A, $C_1 \sqcap C_2$, and $\exists R.C$ for A an atomic concept and R an atomic role, and it supports no axioms about roles. Significant effort has been devoted to the development of DL languages with good computational properties, such as $\mathcal{EL}$, DL-Lite [Calvanese *et al.*, 2007], and Horn-$\mathcal{SHIQ}$ [Hustadt *et al.*, 2005]. Each knowledge base $\mathcal{K}$ expressed in one of these languages is *Horn* in the sense that the intersection of every compatible set of models of $\mathcal{K}$ is also a model of $\mathcal{K}$.

Table 2: Example Knowledge Bases

Hidden Knowledge Base $\mathcal{K}_h$	
γ_1	$\mathbf{CHD\_Heart} \equiv \mathbf{Heart} \sqcap \exists\mathbf{cond}.\mathbf{CHD}$
γ_2	$\mathbf{VSD\_Heart} \equiv \mathbf{Heart} \sqcap \exists\mathbf{cond}.\mathbf{VSD}$
γ_3	$\mathbf{VSD} \sqsubseteq \mathbf{CHD}$
γ_4	$\mathbf{AS} \sqsubseteq \mathbf{CHD}$
Visible Knowledge Base $\mathcal{K}_v$	
δ_1	$\mathit{CHD\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.\mathbf{CHD\_Heart}$
δ_2	$\mathit{VSD\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.\mathbf{VSD\_Heart}$
δ_3	$\mathit{AS\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.(\mathbf{Heart} \sqcap \exists\mathbf{cond}.\mathbf{AS})$
δ_4	$\mathit{EA\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.(\mathbf{Heart} \sqcap \exists\mathbf{cond}.\mathit{EA})$
δ_5	$\mathit{EA} \sqsubseteq \mathbf{CHD}$

For α a concept, a role, an axiom, or a knowledge base, $\mathsf{sig}(\alpha)$ is the *signature* of α—that is, the set of atomic concepts, atomic roles, and individuals occurring in α. A *position* p is a finite sequence of integers. The empty position is denoted with ϵ. If a position p_1 is a proper prefix of a position p_2, then and p_1 is *above* p_2, and p_2 is *below* p_1. The subterm $\alpha|_p$ of a concept or axiom α at a position p is defined as follows: $\alpha|_\epsilon = \alpha$; $(C_1 \bowtie C_2)|_{ip} = C_i|_p$ for $\bowtie \in \{\sqcap, \sqsubseteq\}$ and $i \in \{1, 2\}$; and $\alpha|_{1p} = C|_p$ for α of the form $\neg C$, $\exists R.C$, $\geqslant n\, R.C$, or $C(a)$. The *concept closure* $\mathsf{cls}(\mathcal{K})$ of $\mathcal{K} = \langle \mathcal{T}, \mathcal{A}\rangle$ is the smallest set that contains all subterms of $\neg C \sqcup D$ for each $C \sqsubseteq D \in \mathcal{T}$ and of C for each $C(a) \in \mathcal{A}$.

3 Importing Ontologies by Query

To illustrate the notion of import-by-query, Table 2 shows a reference knowledge base $\mathcal{K}_h$ whose axioms are to be kept hidden, but that is reused in a visible knowledge base $\mathcal{K}_v$. The hidden knowledge base $\mathcal{K}_h$ provides concepts describing organs such as **Heart**, and medical conditions such as **CHD** (congenital heart defect), **VSD** (ventricular septum defect), and **AS** (aortic stenosis). Furthermore, the role **cond** relates organs to medical conditions and is used to define concepts such as **CHD_Heart** (a heart with a congenital heart disorder) and **VSD_Heart** (a heart with a ventricular septal defect). The shared symbols of $\mathcal{K}_h$ are written in bold font. In addition to these, $\mathcal{K}_h$ might contain nonshared symbols; however, for the sake of brevity, we do not show any axioms involving such symbols. The visible knowledge base $\mathcal{K}_v$ provides the concept Pat representing patients, and it defines various types of patients by relating the organs from $\mathcal{K}_h$ with the patients using the $\mathit{hasOrgan}$ role. In addition, $\mathcal{K}_v$ extends the list of defects in $\mathcal{K}_h$ by EA (Ebstein's anomaly). The symbols private to $\mathcal{K}_v$ are written in italic font.

When reusing ontologies, it is commonly accepted that $\mathcal{K}_v$ should not affect the meaning of the symbols reused from $\mathcal{K}_h$—that is, $\mathcal{K}_v \cup \mathcal{K}_h \models \alpha$ should imply $\mathcal{K}_h \models \alpha$ for each axiom α containing only the reused symbols [Lutz *et al.*, 2007; Cuenca Grau *et al.*, 2008]. This is guaranteed if the TBox $\mathcal{T}_v$ of $\mathcal{K}_v$ is *local* w.r.t. the set Γ of concepts and roles imported from $\mathcal{K}_h$—that is, if $I \models \mathcal{T}_v$ for each interpretation I in which, for each concept or role $X \notin \Gamma$, we have $X^I = \emptyset$. For example, δ_1 is local w.r.t. $\{\mathbf{CHD\_Heart}\}$ because δ_1 is satisfied in any interpretation that interprets the nonshared symbols as $\emptyset$. [Cuenca Grau *et al.*, 2008] have shown how to check this condition using a DL reasoner.

To formalize the notion of import-by-query, we introduce the notion of a Γ-oracle, which is responsible for advertising the shared signature Γ of $\mathcal{K}_h$ and answering satisfiability of (not necessarily atomic) concepts w.r.t. $\mathcal{K}_h$. Concept satisfiability is available in all DL reasoners known to us, so it provides us with a natural query language for Γ-oracles; we leave the investigation of richer query languages to future work.

Definition 1. *Let $\mathcal{K}$ be a KB and $\Gamma \subseteq \mathsf{sig}(\mathcal{K})$ a signature. The* Γ-oracle *for $\mathcal{K}$ is the function $\Omega_\mathcal{K}$ defined for each concept C (in the same DL as $\mathcal{K}$) with $\mathsf{sig}(C) \subseteq \Gamma$ such that $\Omega_\mathcal{K}(C) = \mathsf{t}$ if C is satisfiable w.r.t. $\mathcal{K}$, and $\Omega_\mathcal{K}(C) = \mathsf{f}$ otherwise.*

An import-by-query algorithm checks whether $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable; other relevant reasoning problems, such as concept subsumption, can be solved using the well-known transformations. The notion of an algorithm in the following definition can be made precise using a formal computation model such as Turing machines in the obvious way.

Definition 2. *An* import-by-query algorithm *takes a Γ-oracle $\Omega_{\mathcal{K}_h}$ and a KB $\mathcal{K}_v$ with $\mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h) \subseteq \Gamma$ as input, and it terminates after a finite number of computation steps returning t iff $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable.*

4 The Limits of Import-by-Query Reasoning

We next show that no import-by-query algorithm exists even for a light-weight DL such as $\mathcal{EL}$.

Theorem 1. *No import-by-query algorithm exists if $\mathcal{K}_v$ and $\mathcal{K}_h$ are in $\mathcal{EL}$, Γ is allowed to contain at least one atomic role, and the TBox of $\mathcal{K}_v$ is local in Γ.*

Proof. Consider an application of an import-by-query algorithm to $\mathcal{K}_v$ given in (1) and $\Gamma = \{R\}$. Clearly, the TBox of $\mathcal{K}_v$ is local in Γ. Since the algorithm terminates on all inputs, the number of questions posed to any Γ-oracle is bounded by some integer m and, consequently, the quantifier depth of each concept C passed to the Γ-oracle is bounded by an integer n, where both m and n depend only on Γ and $\mathcal{K}_v$. Let $\mathcal{K}_h^1$ and $\mathcal{K}_h^2$ be as in (2) and (3), respectively.

$$\mathcal{K}_v = \{A(a),\ A \sqsubseteq \exists R.A\} \tag{1}$$

$$\mathcal{K}_h^1 = \emptyset \tag{2}$$

$$\mathcal{K}_h^2 = \{\ \underbrace{\exists R.\ldots\exists R}_{n+1 \text{ times}}.\top \sqsubseteq \bot\ \} \tag{3}$$

For each $\mathcal{EL}$ concept C of quantifier depth at most n with $\mathsf{sig}(C) \subseteq \Gamma$, we have $\mathcal{K}_h^1 \models C \sqsubseteq \bot$ iff $\mathcal{K}_h^2 \models C \sqsubseteq \bot$, so $\Omega_{\mathcal{K}_h^1}(C) = \Omega_{\mathcal{K}_h^2}(C)$. Thus, when applied to $\mathcal{K}_v$ and $\Omega_{\mathcal{K}_h^1}$, the algorithm returns the same value as when it is applied to $\mathcal{K}_v$ and $\Omega_{\mathcal{K}_h^2}$. Since $\mathcal{K}_v \cup \mathcal{K}_h^1$ is satisfiable but $\mathcal{K}_v \cup \mathcal{K}_h^2$ is not, the algorithm does not satisfy Definition 2. $\square$

5 Importing Atomic Concepts

The proof of Theorem 1 relies on the fact that $\mathcal{K}_v$ reuses a role from $\mathcal{K}_h$. We now present an import-by-query algorithm for the case when no role is reused. In our example, this allows one to express axioms δ_1, δ_2, and δ_5, which, together with $\mathcal{K}_h$, allow us to conclude $VSD\_Pat \sqsubseteq CHD\_Pat$.

5.1 Interfacing Models Point-Wise

The following definition identifies valid inputs for our algorithm. In particular, we allow $\mathcal{K}_v$ to be any OWL 2 ontology that reuses the symbols of $\mathcal{K}_h$ in a local way; however, we disallow the usage of nominals in $\mathcal{K}_h$ for technical reasons.

Definition 3. *Let $\mathcal{K}_v = \langle \mathcal{T}_v, \mathcal{A}_v \rangle$ and $\mathcal{K}_h = \langle \mathcal{T}_h, \mathcal{A}_h \rangle$ be KBs such that $\Gamma = \mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h)$ contains only atomic concepts. Then, $\mathcal{K}_h$ is* safe for import-by-query *into $\mathcal{K}_v$ if $\mathcal{K}_v$ is in $\mathcal{SROIQ}$, $\mathcal{K}_h$ is in $\mathcal{SRIQ}$, and $\mathcal{T}_v$ is local w.r.t. Γ.*

Our core observation is that a model of $\mathcal{K}_v \cup \mathcal{K}_h$ can be obtained by taking a model I of $\mathcal{K}_v$ and extending it at each point $x \in \triangle^I$ with a fresh model J_x of $\mathcal{K}_h$ that contains a point $y \in \triangle^{J_x}$ such that x and y coincide on the interpretation of the concepts in Γ. This is a consequence of the fact that (*i*) $\mathcal{K}_v$ uses the concepts from Γ in a local way, and (*ii*) $\mathcal{K}_h$ does not contain nominals, so the union of all models J_x is also a model of $\mathcal{K}_h$. To formalize this idea, we use the following notion: for $S = \{D_1, \ldots, D_n\}$ a nonempty finite set of concepts, a *selection* w.r.t. S is a concept of the form $L_1 \sqcap \ldots \sqcap L_n$ where each L_i is either D_i or $\neg D_i$; furthermore, $\top$ is the only selection w.r.t. $S = \emptyset$.

Lemma 1. *Let $\mathcal{K}_h$ be safe for import-by-query into $\mathcal{K}_v$, and let $\Gamma = \mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h)$. Then, $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable iff a model I of $\mathcal{K}_v$ exists such that $\Omega_{\mathcal{K}_h}(C) = \mathsf{t}$ for each selection C w.r.t. Γ such that $C^I \neq \emptyset$.*

Proof. ($\Rightarrow$) If I is a model of $\mathcal{K}_v \cup \mathcal{K}_h$, then clearly $I \models \mathcal{K}_v$, and $\Omega_{\mathcal{K}_h}(C) = \mathsf{t}$ for each selection C w.r.t. Γ with $C^I \neq \emptyset$.

($\Leftarrow$) Let $I = (\triangle^I, \cdot^I)$ be a model of $\mathcal{K}_v$ and consider each $x \in \triangle^I$ and the selection C w.r.t. Γ such that $x \in C^I$. Since $\Omega_{\mathcal{K}_h}(C) = \mathsf{t}$, an interpretation $J_x = (\triangle^{J_x}, \cdot^{J_x})$ exists such that $J_x \models \mathcal{K}_h$ and $y \in C^{J_x}$ for some $y \in \triangle^{J_x}$. W.l.o.g. we assume that $y = x$; $\triangle^{J_x} \cap \triangle^I = \{x\}$; $\triangle^{J_{x_1}} \cap \triangle^{J_{x_2}} = \emptyset$ for each $x_1, x_2 \in \triangle^I$ with $x_1 \neq x_2$; and $X^{J_x} = \emptyset$ for each $X \in \mathsf{sig}(\mathcal{K}_v) \setminus \Gamma$. Let $M = (\triangle^M, \cdot^M)$ be such that

$$\triangle^M = \bigcup_{x \in \triangle^I} \triangle^{J_x},$$

$$X^M = \bigcup_{x \in \triangle^I} X^{J_x} \text{ for each atomic concept or role } X, \text{ and}$$

$$a^M = a^{J_x} \text{ for each individual } a \text{ and some (arbitrarily chosen) interpretation } J_x.$$

$\mathcal{SRIQ}$ does not allow for nominals, so it is *invariant under disjoint unions*—that is, the union of any number of disjoint models of $\mathcal{K}_h$ is also a model of $\mathcal{K}_h$ [Baader *et al.*, 2002]; thus, $M \models \mathcal{K}_h$. Furthermore, since $\mathcal{T}_v$ is local in Γ, we have $M \models \mathcal{T}_v \cup \mathcal{K}_h$. Finally, let $N = (\triangle^N, \cdot^N)$ be an interpretation defined by $\triangle^N = \triangle^M$ and

$$X^N = \begin{cases} X^I & \text{for each } X \in \mathsf{sig}(\mathcal{K}_v) \setminus \Gamma \\ X^M & \text{for each } X \in \mathsf{sig}(\mathcal{K}_h) \end{cases}.$$

Algorithm 1 Import-by-Query Algorithm

Algorithm: $\mathsf{ibq}(\mathcal{K}_v, \Omega_{\mathcal{K}_h}, S)$
Inputs: a knowledge base $\mathcal{K}_v$, a Γ-oracle $\Omega_{\mathcal{K}_h}$, and a set of concepts S over the signature Γ

1 Compute the set N of all axioms of the form $C \sqsubseteq \bot$ such that C is a selection w.r.t. S with $\Omega_{\mathcal{K}_h}(C) = \mathsf{f}$.
2 Return t iff the $\mathcal{SROIQ}$ knowledge base $\mathcal{K}_v \cup N$ is satisfiable.

N and M have the same domains and they coincide on the interpretation of the symbols in $\mathsf{sig}(\mathcal{K}_h)$, so $N \models \mathcal{K}_h$. To show that $N \models \mathcal{K}_v$, we first prove the following claim ($\star$): for each $C \in \mathsf{cls}(\mathcal{K}_v)$, we have $C^N = C^I \cup (C^M \setminus \triangle^I)$. The proof of ($\star$) is by induction on the structure of concepts, so consider each $C \in \mathsf{cls}(\mathcal{K}_v)$.

If C is an atomic concept with $C \in \Gamma$, then by the definition of M we have $C^I = C^M \cap \triangle^I$, so $C^I \cup (C^M \setminus \triangle^I) = (C^M \cap \triangle^I) \cup (C^M \setminus \triangle^I) = C^M$; by the definition of N, we have $C^M = C^N$, which implies ($\star$).

If C is a nominal or an atomic concept with $C \notin \Gamma$, then $C^M = \emptyset$ and $C^N = C^I$, which trivially imply ($\star$).

If $C = \neg D$, then $C^N = (\triangle^I \cup (\triangle^M \setminus \triangle^I)) \setminus D^N = (\triangle^I \setminus D^N) \cup ((\triangle^M \setminus \triangle^I) \setminus D^N)$. By applying the induction hypothesis, the first disjunct reduces to $\triangle^I \setminus D^I$, and, since, $\triangle^M \setminus D^N = \triangle^M \setminus (D^I \cup (D^M \setminus \triangle^I)) = (\triangle^M \setminus D^I) \setminus (D^M \setminus \triangle^I) = \triangle^M \setminus D^M$, the second one reduces to $(\triangle^M \setminus D^M) \setminus \triangle^I$. But then, ($\star$) holds.

If $C = D_1 \sqcap D_2$, then $C^N = D_1^N \cap D_2^N$, which is equal to $(D_1^I \cup (D_1^M \setminus \triangle^I)) \cap (D_2^I \cup (D_2^M \setminus \triangle^I))$ by the induction hypothesis; but $(D_1^M \setminus \triangle^I) \cap D_2^I = (D_2^M \setminus \triangle^I) \cap D_1^I = \emptyset$, so $C^N = (D_1^I \cap D_2^I) \cup ((D_1^M \setminus \triangle^I) \cap (D_2^M \setminus \triangle^I))$; finally, $(D_1^M \setminus \triangle^I) \cap (D_2^M \setminus \triangle^I) = (D_1^M \cap D_2^M) \setminus \triangle^I$.

If $C = \geqslant n\, R.D$ or $C = \exists R.\mathsf{Self}$, since $R \notin \mathsf{sig}(\mathcal{K}_h)$, we have $R^M = \emptyset$ and $C^M = \emptyset$; furthermore, $R^N = R^I$ and $D^I \subseteq D^N$ by the induction hypothesis, so $C^N = C^I$. This completes the proof of ($\star$).

Consider now each axiom α in $\mathcal{K}_v$. For α a concept inclusion axiom, we assume w.l.o.g. that it is of the form $\top \sqsubseteq C$. By ($\star$), $C^N = C^I \cup (C^M \setminus \triangle^I)$. Since $I \models \alpha$, we have $C^I = \triangle^I$; furthermore, since $\mathcal{T}_v$ is local w.r.t. Γ, we have $M \models \alpha$, so $C^M = \triangle^M$; thus, $C^N = \triangle^N$, so $N \models \alpha$. For α a role assertion, a role inclusion, or a role disjointness axiom, we have $N \models \alpha$ because N coincides with I on the interpretation of all roles from $\mathsf{sig}(\mathcal{K}_v)$. For $\alpha = C(a)$, we have $a^N \in C^N$ by ($\star$) and $a \notin \Gamma$. Finally, for $\alpha = a \not\approx b$, we have $a^N \neq b^N$ because $\{a, b\} \cap \Gamma = \emptyset$. Thus, $N \models \mathcal{K}_v$. □

Lemma 1 motivates Algorithm 1.

Theorem 2. *Let $\mathcal{K}_h$ be safe for import-by-query into $\mathcal{K}_v$, $\Gamma = \mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h)$, and $\Omega_{\mathcal{K}_h}$ the Γ-oracle for $\mathcal{K}_h$. Then, $\mathsf{ibq}(\mathcal{K}_v, \Omega_{\mathcal{K}_h}, \Gamma)$ is an import-by-query algorithm, and it can be implemented such that it runs in* N2ExpTime *with an exponential number of calls to $\Omega_{\mathcal{K}_h}$.*

Proof. That $\mathsf{ibq}(\mathcal{K}_v, \Omega_{\mathcal{K}_h}, \Gamma)$ is an import-by-query algorithm is a direct consequence of Lemma 1. Furthermore, the number of selections w.r.t. Γ is exponential in the size of Γ, so N can be computed by an exponential number of calls to $\Omega_{\mathcal{K}_h}$. Let $\mathsf{ria}(\cdot)$ be the transformation by [Kazakov, 2008] for eliminating role inclusion axioms from $\mathcal{SROIQ}$ KBs. Then, $\mathsf{ria}(\mathcal{K}_v)$ is equisatisfiable with and exponentially larger than $\mathcal{K}_v$ [Kazakov, 2008]. Furthermore, N contains the same concepts as $\mathsf{ria}(\mathcal{K}_v)$ and no role inclusions axioms, so $\mathsf{ria}(\mathcal{K}_v \cup N) = \mathsf{ria}(\mathcal{K}_v) \cup N = \mathcal{K}'$. Thus, $\mathcal{K}'$ is equisatisfiable with and exponentially larger than $\mathcal{K}_v$. We can check satisfiability of $\mathcal{K}'$ by transforming $\mathcal{K}'$ polynomially into an equisatisfiable formula φ of the two-variable fragment with counting, and deciding the satisfiability of φ in NExpTime [Pratt-Hartmann, 2005]. Clearly, the overall algorithm runs in N2ExpTime with exponentially many calls to $\Omega_{\mathcal{K}_h}$. □

5.2 Importing Horn Ontologies

Algorithm 1 is unlikely to be suitable for practice because Step 1 is exponential in the size of Γ. In this section, we present a practical algorithm for the case when $\mathcal{K}_h$ is Horn.[1] This algorithm calls the Γ-oracle "on demand," which makes it "more goal-oriented." The correctness of the algorithm is based on the following observation about Horn KBs.

Proposition 1. *Let $\mathcal{K}$ be a Horn knowledge base, C a conjunction of atomic concepts, and $A_1, \ldots, A_n$ atomic concepts such that $C \sqcap \neg A_i$ is satisfiable w.r.t. $\mathcal{K}$ for each $1 \leq i \leq n$. Then, $C \sqcap \neg A_1 \sqcap \ldots \sqcap \neg A_n$ is satisfiable w.r.t. $\mathcal{K}$ as well.*

Proof. Let $\mathcal{K}_i = \mathcal{K} \cup \{C(a), \neg A_i(a)\}$ for $1 \leq i \leq n$ and a an individual not occurring in $\mathcal{K}$. Let I_i be a model of each $\mathcal{K}_i$; w.l.o.g. we assume that the set $S = \{I_i \mid 1 \leq i \leq n\}$ is compatible (e.g., we can select I_i to be Herbrand models of $\mathcal{K}_i$). Let J be the intersection of S. Since $\mathcal{K}$ is Horn, we have $J \models \mathcal{K}$. Furthermore, $a^J \in C^J$ and $a^J \notin A_i^J$ for each $1 \leq i \leq n$; therefore, $a^J \in (C \sqcap \neg A_1 \sqcap \ldots \sqcap \neg A_n)^J$. □

We extend the tableau algorithms used in many state-of-the-art DL reasoners. Our extension, however, is largely independent from the intricacies of these algorithms, so we introduce an abstraction of a *tableau algorithm* as a tuple $\mathsf{T} = \langle \mathsf{C}, \mathsf{R} \rangle$ with the following structure.

- C assigns to each ABox $\mathcal{A}$ a value from $\{\mathsf{t}, \mathsf{f}\}$ such that $\mathsf{C}(\mathcal{A}) = \mathsf{t}$ only if $\mathcal{A}$ is unsatisfiable. $\mathcal{A}$ contains a *clash* if $\mathsf{C}(\mathcal{A}) = \mathsf{t}$; otherwise, $\mathcal{A}$ is *clash-free*.
- R is a set of *derivation rules*, where each $\rho \in \mathsf{R}$ assigns to each pair $(\mathcal{T}, \mathcal{A})$ a set of n-tuples of ABoxes (tuples in this set can vary in arity). A rule ρ is *applicable* to $\mathcal{T}$ and $\mathcal{A}$ if $\rho(\mathcal{T}, \mathcal{A}) \neq \emptyset$.

A *derivation* for $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ by $\mathsf{T} = \langle \mathsf{C}, \mathsf{R} \rangle$ is a pair $\langle \Theta, \sigma \rangle$ where Θ is a finitely branching tree and σ labels each node v of Θ with an ABox $\sigma(v)$ such that (*i*) $\sigma(v) = \mathcal{A}$ for v the root of Θ; (*ii*) if $\mathsf{C}(\sigma(v)) = \mathsf{t}$ or no derivation rule in R is applicable to $(\mathcal{T}, \sigma(v))$, then v is a leaf of Θ; (*iii*) if $\mathsf{C}(\sigma(v)) = \mathsf{f}$ and a derivation rule in R is applicable to $(\mathcal{T}, \sigma(v))$, then v has children $v_1, \ldots, v_n$ such that $\langle \sigma(v_1), \ldots, \sigma(v_n) \rangle \in \rho(\mathcal{T}, \sigma(v))$ for some (arbitrarily chosen) derivation rule $\rho \in \mathsf{R}$.

T is *terminating* if, for each $\mathcal{K}$, each derivation for $\mathcal{K}$ by T can be constructed using finitely many steps. T is *sound*

[1] From the infrastructure point of view, the Γ-oracle for $\mathcal{K}_h$ should indicate to clients if $\mathcal{K}_h$ is (known to be) Horn.

if, for each model I of each $\langle \mathcal{T}, \mathcal{A} \rangle$, each derivation rule $\rho \in \mathsf{R}$, and each $\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle \in \rho(\mathcal{T}, \mathcal{A})$, an interpretation I' exists such that $X^I = X^{I'}$ for each $X \notin \mathsf{sig}(\mathcal{A}_i) \setminus \mathsf{sig}(\mathcal{A})$, and $I' \models \langle \mathcal{T}, \mathcal{A}_i \rangle$ for some $1 \leq i \leq n$. T is *complete* if a partial function M mapping ABoxes to interpretations exists such that, in each derivation (Θ, σ) for $\mathcal{K}$ by T, and for each leaf v of (Θ, σ) such that $\mathcal{A}' = \sigma(v)$ is clash-free, the value of $\mathsf{M}(\mathcal{A}')$ is defined and $\mathsf{M}(\mathcal{A}') \models \mathcal{K}$. Furthermore, we assume that $\mathsf{M}(\mathcal{A}') = (\triangle^I, \cdot^I)$ always satisfies the following property ($\Diamond$): for each conjunction of atomic concepts $C = A_1 \sqcap \ldots \sqcap A_n$ such that $C^I \neq \emptyset$ and $(C \sqcap B)^I = \emptyset$ for each atomic concept B not occurring in C, an individual s exists such that $A_i(s) \in \mathcal{A}'$ for each $1 \leq i \leq n$, and $B(s) \notin \mathcal{A}'$ for each atomic concept B not occurring in C. Intuitively, ($\Diamond$) ensures that conjunctive concepts are interpreted in $\mathsf{M}(\mathcal{A}')$ in accordance with their instantiations in $\mathcal{A}'$. Most tableau algorithms used in practice are sound, complete, and terminating; furthermore, all such algorithms known to us satisfy ($\Diamond$).

We now show how to extend T to an import-by-query algorithm for the case when $\mathcal{K}_h$ is Horn.

Definition 4. *Let $\mathsf{T} = \langle \mathsf{C}, \mathsf{R} \rangle$ be a sound, complete, and terminating tableau algorithm, Γ a set of atomic concepts, and $\Omega_{\mathcal{K}_h}$ a Γ-oracle. The tableau algorithm $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ is obtained by extending T with the* ask-rule *as follows: $\mathsf{ask}(\mathcal{T}, \mathcal{A})$ is defined for each $(\mathcal{T}, \mathcal{A})$ as the smallest set such that, for each individual s in $\mathcal{A}$, the concept C obtained as the conjunction of all $A_i \in \Gamma$ with $A_i(s) \in \mathcal{A}$, and each $B \in \Gamma \cup \{\bot\}$ with $\Omega_{\mathcal{K}_h}(C \sqcap \neg B) = \mathsf{f}$, we have*

$$\langle \mathcal{A} \cup \{B(s)\} \rangle \in \mathsf{ask}(\mathcal{T}, \mathcal{A}).$$

Intuitively, the ask-rule deterministically adds $B(s)$ to each ABox that contains assertions $A_1(s) \ldots, A_n(s)$ such that $\mathcal{K}_h \models A_1 \sqcap \ldots \sqcap A_n \models B$.

Theorem 3. *Let $\mathcal{K}_h = \langle \mathcal{T}_h, \mathcal{A}_h \rangle$ be a Horn knowledge base that is safe for import-by-query into $\mathcal{K}_v = \langle \mathcal{T}_v, \mathcal{A}_v \rangle$, let $\Gamma = \mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h)$, let $\Omega_{\mathcal{K}_h}$ be the Γ-oracle for $\mathcal{K}_h$, and let T be a sound, complete, and terminating tableau algorithm. Then, $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ satisfies the following two claims:*

1. *if $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable, then each derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ contains a branch on which all nodes are labeled with clash-free ABoxes; and*
2. *if a derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ contains a leaf labeled with a clash-free ABox, then $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable.*

Proof. (Claim 1) Assume that I is a model of $\mathcal{K}_v \cup \mathcal{K}_h$, and consider each derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$. We assume w.l.o.g. that the derivation rules of $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ do not introduce assertions involving symbols from $\mathsf{sig}(\mathcal{K}_h) \setminus \Gamma$. Consider now the tuple $\langle \mathcal{A}_1, \ldots, \mathcal{A}_n \rangle$ obtained from $\mathcal{A}_v$ by an application of a derivation rule of $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$. If the derivation rule is from T, since T is sound, I can be extended to a model I' of some $\langle \mathcal{T}_v, \mathcal{A}_i \rangle$; since this extension does not involve the symbols in $\mathsf{sig}(\mathcal{K}_h)$, we have $I' \models \mathcal{K}_h$ as well. For the ask-rule, $n = 1$ and $s^I \in C^I$, so $\Omega_{\mathcal{K}_h}(C \sqcap \neg B) = \mathsf{f}$ implies $s^I \in B^I$ and $I \models \langle \mathcal{T}_v, \mathcal{A}_1 \rangle \cup \mathcal{K}_h$. By repeating this claim inductively, we conclude that the derivation contains a branch on which each node is labeled with an ABox $\mathcal{A}'$ such that $\langle \mathcal{T}_v, \mathcal{A}' \rangle \cup \mathcal{K}_h$ is satisfiable; thus, each $\mathcal{A}'$ is clash-free.

(Claim 2) Let $\mathcal{A}'$ be a clash-free ABox labeling a leaf of a derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$, and let $\mathsf{M}(\mathcal{A}') = (\triangle^I, \cdot^I)$. Furthermore, let C be a selection w.r.t. Γ such that $C^I \neq \emptyset$, and let D be the conjunction of all atomic concepts that occur positively in C. By ($\Diamond$) and the fact that C is maximal, an individual s in $\mathcal{A}$ exists such that $A_i(s) \in \mathcal{A}'$ for each A_i in D, and $B_j(s) \notin \mathcal{A}'$ for each atomic concept B_j, $1 \leq j \leq n$ that occurs negatively in C. Since the ask-rule is not applicable to $\mathcal{A}'$, then $D \sqcap \neg B_j$ is satisfiable w.r.t. $\mathcal{K}_h$ for each $1 \leq j \leq n$. Since $\mathcal{K}_h$ is Horn, by Proposition 1 we have that $D \sqcap \neg B_1 \sqcap \ldots \sqcap \neg B_n = C$ is satisfiable w.r.t. $\mathcal{K}_h$ as well. But then, $\mathcal{K}_v \cup \mathcal{K}_h$ is satisfiable by Lemma 1. $\square$

Each derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ is clearly finite. Furthermore, the value of $\mathsf{ask}(\mathcal{T}, \mathcal{A})$ can be determined by asking $\Omega_{\mathcal{K}_h}(C)$ for each selection C w.r.t. Γ occurring in $\mathcal{A}$. Therefore, a derivation for $\mathcal{K}_v$ by $\mathsf{T}_{\Gamma, \Omega_{\mathcal{K}_h}}$ can be constructed by a finite number of steps, which provides us with an import-by-query algorithm. Such an algorithm may, in the worst case, make an exponential number of calls to the oracle; however, such calls are made as needed, which makes this algorithm more amenable to implementation than Algorithm 1.

6 Importing Atomic Roles

We now extend the results from Section 5 and allow the reuse of roles under the following syntactic restriction.

Definition 5. *For Γ a set of atomic concepts and roles, we say that a concept is* Γ-modal *if it is of the form $\exists R.\mathsf{Self}$, $\exists R.C$, or $\geqslant n\, R.C$, for $R \in \Gamma$.*

Let $\mathcal{K}_v$ and $\mathcal{K}_h$ be KBs such that $\Gamma = \mathsf{sig}(\mathcal{K}_v) \cap \mathsf{sig}(\mathcal{K}_h)$ contains both concepts and roles. Then, $\mathcal{K}_h$ is safe for import-by-query *into $\mathcal{K}_v$ if, in addition to the conditions from Definition 3, roles from Γ do not occur in role inclusion and disjointness axioms in $\mathcal{K}_v$; for each $\exists R.\mathsf{Self}$ or $\geqslant n\, R.C$ in $\mathcal{K}_v$, if $R \in \Gamma$ then R is simple in $\mathcal{K}_h$; and $\mathsf{sig}(C) \subseteq \Gamma$ for each Γ-modal concept $C \in \mathsf{cls}(\mathcal{K}_v)$.*

For satisfiability of $\mathcal{K}_v \cup \mathcal{K}_h$ to be decidable, only simple roles from $\mathcal{K}_h$ can occur in certain concepts in $\mathcal{K}_v$ [Horrocks *et al.*, 2000]. Thus, the Γ-oracle for $\mathcal{K}_v$ should also advertise to clients which roles in Γ are simple. This is a syntactic check that is provided by most DL reasoners.

In our example, Definition 5 allows us to express δ_3: the role $\mathbf{cond}$ from $\mathcal{K}_h$ occurs in a Γ-modal concept $\exists\mathbf{cond}.\mathbf{AS}$, but $\mathbf{AS}$ is from $\mathcal{K}_h$ as well. This is in contrast to δ_4, in which $\exists\mathbf{cond}.EA$ contains EA that is not from $\mathcal{K}_h$. Note that δ_1, δ_3, and $\mathcal{K}_h$ allow us to conclude $AS\_Pat \sqsubseteq CHD\_Pat$.

By using the appropriate set S, Algorithm 1 is an import-by-query algorithm for the case of shared roles as well. In the following theorem, we say that position p in a concept or axiom α is Γ-*outermost* if $\alpha|_p$ is a Γ-modal concept, and $\alpha|_q$ is not a Γ-modal concept for each position q above p.

Theorem 4. *Let $\mathcal{K}_v$, Γ, and $\Omega_{\mathcal{K}_h}$ be as in Theorem 2 with the difference that Γ can also contain atomic roles, and let*

$$\begin{aligned} S = &\{A \in \Gamma \mid A \text{ is an atomic concept}\} \cup \\ &\{\alpha|_p \mid \alpha \in \mathcal{K}_v \text{ and } p \text{ is } \Gamma\text{-outermost in } \alpha\}. \end{aligned}$$

Then, $\mathsf{ibq}(\mathcal{K}_v, \Omega_{\mathcal{K}_h}, S)$ *is an import-by-query algorithm, and it can be implemented such that it runs in* N2ExpTime *with an exponential number of calls to* $\Omega_{\mathcal{K}_h}$.

Proof. Let Q_D be a fresh atomic concept uniquely associated with each $D \in S$. Furthermore, let $\mathcal{K}'_v$ be the knowledge base obtained from $\mathcal{K}_v$ by replacing in each axiom $\alpha \in \mathcal{K}_v$ the concept $\alpha|_p$ with $Q_{\alpha|_p}$ for each Γ-outermost position p in α. Also, let $\mathcal{K}'_h$ be obtained from $\mathcal{K}_h$ by adding the axiom $Q_C \equiv C$ for each C in S. Finally, let $\Gamma' = \mathsf{sig}(\mathcal{K}'_v) \cap \mathsf{sig}(\mathcal{K}'_h)$, and let $\Omega_{\mathcal{K}'_h}$ be the Γ'-oracle such that $\Omega_{\mathcal{K}'_h}(C_1) = \Omega_{\mathcal{K}_h}(C_2)$ for each C_1 and C_2 where C_2 is obtained from C_1 by replacing all Q_D with D. Since $\mathcal{K}_v$ satisfies the condition from Definition 5 and Γ' contains only atomic concepts, $\mathcal{K}'_h$, $\mathcal{K}'_v$, and $\Omega_{\mathcal{K}'_h}$ satisfy the preconditions of Theorem 2. Furthermore, it is obvious that $\mathsf{ibq}(\mathcal{K}'_v, \Omega_{\mathcal{K}'_h}, \Gamma') = \mathsf{ibq}(\mathcal{K}_v, \Omega_{\mathcal{K}_h}, S)$, so the latter is an import-by-query algorithm. The proof for the algorithm's running time is the same as in Theorem 2. □

When each Γ-modal concept C occurring in $\mathcal{K}_v$ is Horn [Hustadt *et al.*, 2005], the tableau algorithm from Definition 4 can be extended to the case when Γ contains roles by using the set S from Theorem 4 instead of Γ in the ask-rule.

Finally, the results from this section can be extended to the case when $\mathcal{K}_v$ contains concepts of the form $\geqslant n\, R.D$ with $R \in \Gamma$ and $\mathsf{sig}(D) \not\subseteq \Gamma$, provided that the unfolding of D in $\mathcal{K}_v$ and results in a concept containing only symbols from Γ. In our example, the nonshared symbol EA in δ_4 can be unfolded with its definition in δ_5, resulting in $EA\_Pat \equiv Pat \sqcap \exists hasOrgan.(\mathbf{Heart} \sqcap \exists \mathbf{cond}.\mathbf{CHD})$; after this preprocessing step, we can use the import-by-query algorithm to conclude $EA\_Pat \sqsubseteq CHD\_Pat$.

7 Related Work

In a peer-to-peer setting, [Calvanese *et al.*, 2004] consider the problem of answering a query q over two KBs $\mathcal{K}_v$ and $\mathcal{K}_h$ with disjoint signatures and a set $\mathcal{M}$ of mappings of the form $q_h \rightsquigarrow q_v$ by reformulating q as queries that can be evaluated over $\mathcal{K}_v$ and $\mathcal{K}_h$ in isolation. The query reformulation algorithm accesses only $\mathcal{K}_v$ and $\mathcal{M}$; thus, q can be answered by means of an oracle for $\mathcal{K}_h$. In such a setting, however, a satisfiable $\mathcal{K}_h$ cannot affect the subsumption of concepts in $\mathcal{K}_v$. Consider the following example:

$$\begin{array}{ll} \mathcal{K}_h = \{B_h \sqsubseteq A_h\} & \mathcal{M} = \{\ A_h(x) \rightsquigarrow A_v(x), \\ \mathcal{K}_v = \{C_v \sqsubseteq B_v\} & \qquad\quad B_h(x) \rightsquigarrow B_v(x)\ \} \end{array} \quad (4)$$

Now $\mathcal{K}_h \cup \mathcal{K}_v \cup \mathcal{M} \not\models C_v \sqsubseteq A_v$, since the mappings in $\mathcal{M}$ are unidirectional. Thus, whereas [Calvanese *et al.*, 2004] consider simple schemas (i.e., both $\mathcal{K}_h$ and $\mathcal{K}_v$ must be in DL-Lite) and conjunctive query answering, we focus on rich TBoxes and schema reasoning.

[Baader *et al.*, 2002] study the transfer of decidability results when combining decidable logics. In particular, they show how to integrate algorithms that decide satisfiability of $\mathcal{K}_v$ and $\mathcal{K}_h$ independently into an algorithm that decides satisfiability of $\mathcal{K}_v \cup \mathcal{K}_h$, provided that the two KBs do not share roles and do not contain nominals. This situation is similar to the one in Section 5, with the difference that we allow $\mathcal{K}_v$ to contain nominals but require it to be local in Γ.

8 Conclusion

In this paper, we have studied the problem of importing an ontology without knowing its axioms. We have shown that this problem does not have a general solution. Furthermore, we have identified solvable cases, for which we have presented two algorithms. In future work, one might consider relaxing the syntactic restrictions on the usage of roles, particularly if one were to extend the query language of the oracle.

References

[Baader *et al.*, 2002] F. Baader, C. Lutz, H. Sturm, and F. Wolter. Fusions of Description Logics and Abstract Description Systems. *Journal of Artificial Intelligence Research*, 16:1–58, 2002.

[Baader *et al.*, 2005] F. Baader, S. Brandt, and C. Lutz. Pushing the $\mathcal{EL}$ Envelope. In *Proc. IJCAI,*, pages 364–369, 2005.

[Baader *et al.*, 2007] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2nd edition, 2007.

[Calvanese *et al.*, 2004] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. What to Ask to a Peer: Ontolgoy-based Query Reformulation. In *Proc. KR*, pages 469–478, 2004.

[Calvanese *et al.*, 2007] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning*, 9:385–429, 2007.

[Cuenca Grau *et al.*, 2008] B. Cuenca Grau, I. Horrocks, Y. Kazakov, and U. Sattler. Modular Reuse of Ontologies: Theory and Practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.

[Horrocks *et al.*, 2000] I. Horrocks, U. Sattler, and S. Tobies. Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239–263, 2000.

[Hustadt *et al.*, 2005] U. Hustadt, B. Motik, and U. Sattler. Data Complexity of Reasoning in Very Expressive Description Logics. In *Proc. IJCAI*, pages 466–471, 2005.

[Kazakov, 2008] Y. Kazakov. $\mathcal{RIQ}$ and $\mathcal{SROIQ}$ are Harder than $\mathcal{SHOIQ}$. In *Proc. KR*, pages 274–284, 2008.

[Kutz *et al.*, 2006] O. Kutz, I. Horrocks, and U. Sattler. The Even More Irresistible $\mathcal{SROIQ}$. In *Proc. KR*, pages 68–78, 2006.

[Lenzerini, 2002] M. Lenzerini. Data Integration: A Theoretical Perspective. In *Proc. PODS*, pages 233–246, 2002.

[Lutz *et al.*, 2007] C. Lutz, D. Walther, and F. Wolter. Conservative Extensions in Expressive Description Logics. In *Proc. IJCAI*, pages 453–458, 2007.

[Pratt-Hartmann, 2005] I. Pratt-Hartmann. Complexity of the Two-Variable Fragment with Counting Quantifiers. *Journal of Logic, Language and Information*, 14(3):369–395, 2005.