Importing Ontologies with Hidden Content

Bernardo Cuenca Grau and Boris Motik

Computing Laboratory
University of Oxford, UK

1 Introduction

The Web Ontology Language (OWL) and its revision OWL 2 are widely used ontology languages whose formal underpinnings are given by description logics (DLs). OWL ontologies are used, for example, in several countries to describe electronic patient records (EPR). Patients' data typically involves descriptions of human anatomy, medical conditions, drugs, and so on. These domains have 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 a 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 $\mathcal{K}_h$ [1, 2].

To enable reuse, OWL allows $\mathcal{K}_v$ to *import* $\mathcal{K}_h$. OWL reasoners deal with imports by 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 competitors to plagiarize $\mathcal{K}_h$. Moreover, $\mathcal{K}_h$ might contain information that is sensitive from a privacy point of view. Finally, the vendor of $\mathcal{K}_h$ might impose different costs on parts of $\mathcal{K}_h$. To reflect this situation, we call the ontology $\mathcal{K}_h$ *hidden* and, by analogy, $\mathcal{K}_v$ *visible*.

To enable reuse without providing physical access to the axioms, $\mathcal{K}_h$ could be 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 solve certain reasoning tasks on $\mathcal{K}_v \cup \mathcal{K}_h$ by accessing only $\mathcal{K}_v$ and the oracle. We focus on schema reasoning problems, such as concept subsumption and satisfiability; this is in contrast to the information integration [3] and peer-to-peer [4] scenarios, which focus on the reuse of data.

In our recent work [5], we studied the import-by-query problem when the oracle query language is concept satisfiability (or, equivalently, concept subsumption). We showed that no import-by-query algorithm exists even if $\mathcal{K}_v$ and $\mathcal{K}_h$ are expressed in the lightweight DL $\mathcal{EL}$ [6]. This negative result holds if $\mathcal{K}_v$ reuses an atomic role from $\mathcal{K}_h$, so we studied the case when $\mathcal{K}_v$ reuses only atomic concepts from $\mathcal{K}_h$ in a modular way [2]. We presented an import-by-query

algorithm for the case when the aforementioned assumptions are satisfied and $\mathcal{K}_v$ and $\mathcal{K}_h$ are expressed in $\mathcal{SROIQ}$ [7] and $\mathcal{SRIQ}$ [8], respectively. We also proposed an algorithm that is better suited for practice for the case where $\mathcal{K}_h$ is expressed in a Horn DL. Finally, we extended these results to the case when $\mathcal{K}_v$ reuses atomic roles from $\mathcal{K}_h$ in a syntactically restricted way.

The conditions in [5] on the usage of imported roles, however, can be limiting in practice, so in this paper we investigate ways of relaxing them. In particular, we study a different oracle query language that is based on ABox query answering, rather than concept subsumption or concept satisfiability checking. We present an import-by-query algorithm that uses such a query language and that is applicable if $\mathcal{K}_v$ and $\mathcal{K}_h$ are expressed in $\mathcal{EL}$ and the only restriction is that $\mathcal{K}_v$ imports concepts and roles from $\mathcal{K}_h$ in a modular way. We also establish links between this result and the negative result from [5]. In particular, we show that, if the axioms of $\mathcal{K}_v$ involving the roles imported from $\mathcal{K}_h$ are weakly acyclic [9], an import-by-query algorithm based on concept subsumption exists.

2 Preliminaries

In this section, we present an overview of the DL $\mathcal{EL}$ [6] and a suitable reasoning algorithm. For simplicity, we restrict ourselves to the basic version of $\mathcal{EL}$; however, our results can be easily extended to $\mathcal{EL}$++. We also recapitulate the notion of locality [2], which ensures modular ontology reuse. Finally, we define a restricted form of $\mathcal{EL}$ axioms that guarantees locality.

2.1 The Description Logic $\mathcal{EL}$

The syntax of $\mathcal{EL}$ is defined w.r.t. a *signature*, which is the union of disjoint countable sets of *atomic concepts*, *atomic roles*, and *individuals*. The set of $\mathcal{EL}$ *concepts* is the smallest set containing $\top$, $\bot$, A, $C_1 \sqcap C_2$, and $\exists r.C$, for A an atomic concept, C, C_1, and C_2 $\mathcal{EL}$ concepts, and r an atomic role.

A *concept inclusion* (GCI) has the form $C_1 \sqsubseteq C_2$ for C_1 and C_2 $\mathcal{EL}$ 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 GCIs. An *assertion* has the form $C(a)$ or $r(a, b)$, for C an $\mathcal{EL}$ concept, r an atomic role, and a and b individuals. An *axiom* is either a GCI or an assertion. An *ABox* $\mathcal{A}$ is a finite set of assertions. An $\mathcal{EL}$ *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. For α a concept, an axiom, or a set of axioms, $\mathsf{sig}(\alpha)$ is the *signature* of α—that is, the set of atomic concepts, roles, and individuals occurring in α.

The semantics of $\mathcal{EL}$ is given by means of *interpretations*. The definitions of an $\mathcal{EL}$ interpretation $I = (\triangle^I, \cdot^I)$, satisfaction of an axiom, TBox, ABox, or KB $\mathcal{S}$ in an interpretation I (written $I \models \mathcal{S}$), entailment of an axiom α from $\mathcal{S}$ (written $\mathcal{S} \models \alpha$), and concept subsumption (written $\mathcal{K} \models C \sqsubseteq D$), are standard and can be found in [6]. *Classification* of $\mathcal{K}$ is the problem of determining whether $\mathcal{K} \models C \sqsubseteq D$ for each $C, D \in \mathsf{sig}(\mathcal{K}) \cup \{\top, \bot\}$. For each satisfiable $\mathcal{EL}$ knowledge base $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$, it is well known that $\langle \mathcal{T}, \mathcal{A} \rangle \models C \sqsubseteq D$ iff $\langle \mathcal{T}, \emptyset \rangle \models C \sqsubseteq D$; hence, classification is usually considered w.r.t. a TBox only.

Table 1. $\mathcal{EL}$ Rules

R1	If $\exists r.C(x) \in \mathcal{A}_i$ and $r(x, x_C) \notin \mathcal{A}_i$ then $\mathcal{A}_{i+1} := \mathcal{A}_i \cup \{r(x, x_C)\}$.
R2	If 1. $A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_m.B_m \sqsubseteq C \in \mathcal{T}$ with $k \geq 0$ and $n \geq 0$, and 2. $\mathcal{A}_i$ contains individuals $x, y_1, \ldots, y_m$ such that 2.1 $\{A_j(x) \mid 1 \leq j \leq k\} \cup \{r_j(x, y_j), B_j(y_j) \mid 1 \leq j \leq m\} \subseteq \mathcal{A}_i$, and 2.2 $C(x) \notin \mathcal{A}_i$ then $\mathcal{A}_{i+1} := \mathcal{A}_i \cup \{C(x)\}$.

2.2 Reasoning with $\mathcal{EL}$ Knowledge Bases

We next present two algorithms for reasoning with $\mathcal{EL}$ KBs. First, we present the algorithm $\mathsf{sat}(\mathcal{T}, \mathcal{A})$ that can be used to compute ABox assertions entailed by $\mathcal{T} \cup \mathcal{A}$. Then, we show how to use $\mathsf{sat}(\mathcal{T}, \mathcal{A})$ to obtain the $\mathcal{EL}$ classification algorithm $\mathsf{el}(\mathcal{T})$. These algorithms are similar to the one in [10] and can be seen as a minor variation of the algorithm in [6].

Let Γ be a signature. Then, $\mathsf{BC}(\Gamma)$ is the set of *basic concepts* for Γ, which is the smallest set of $\mathcal{EL}$ concepts containing $\top$, $\bot$, all atomic concepts in Γ, and all concepts of the form $\exists r.C$ such that $r \in \Gamma$ and $C \in \Gamma \cup \{\top, \bot\}$.[1] An $\mathcal{EL}$ TBox $\mathcal{T}$ is in *normal form* if every GCI $\alpha \in \mathcal{T}$ is of the form

$$C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D \tag{1}$$

with $n \geq 1$, and where C_i and D are basic concepts for $\mathsf{sig}(\mathcal{T})$. An $\mathcal{EL}$ ABox $\mathcal{A}$ is in *normal form* if, in each assertion $C(a) \in \mathcal{A}$, the concept C is basic for $\mathsf{sig}(\mathcal{A})$. An $\mathcal{EL}$ KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ is in *normal form* if both $\mathcal{T}$ and $\mathcal{A}$ are in normal form. Given an ABox $\mathcal{A}$ in normal form and $\Gamma \subseteq \mathsf{sig}(\mathcal{A})$, we denote with $\mathcal{A}|_\Gamma$ the set of assertions $C(a)$ and $r(a, b)$ in $\mathcal{A}$ such that $C \in \mathsf{BC}(\Gamma)$ and $r \in \Gamma$.

For each concept $C \in \mathsf{sig}(\mathcal{T})$, let x_C be an individual uniquely associated with C. The algorithm $\mathsf{sat}(\mathcal{T}, \mathcal{A})$ accepts an $\mathcal{EL}$ TBox $\mathcal{T}$ and ABox $\mathcal{A}$ and produces a sequence $\mathcal{A}_0, \ldots \mathcal{A}_n$ of ABoxes, called a *run*, s.t. the following conditions hold:

1. $\mathcal{A}_0 := \mathcal{A} \cup \{C(x_C) \mid C \in \mathsf{sig}(\mathcal{T})\}$.
2. $\mathcal{A}_i$ is obtained from $\mathcal{A}_{i-1}$ by applying a rule from Table 1 for each $1 \leq i \leq n$.
3. No rule from Table 1 is applicable to $\mathcal{A}_n$.

For each $\mathcal{T}$ and $\mathcal{A}$, the ABox $\mathcal{A}_n$ is unique across all runs of $\mathsf{sat}(\mathcal{T}, \mathcal{A})$, so we often write $\mathsf{sat}(\mathcal{T}, \mathcal{A}) = \mathcal{A}_n$. Furthermore, by modifying slightly the proofs from [6] and [10], it is straightforward to show that the following properties hold:

- **P1**: For each TBox $\mathcal{T}$, ABox $\mathcal{A}$, basic concept C for $\mathsf{sig}(\mathcal{T})$, and each individual x in $\mathcal{A}$, we have $C(x) \in \mathsf{sat}(\mathcal{T}, \mathcal{A})$ if and only if $\mathcal{T} \cup \mathcal{A} \models C(x)$.
- **P2**: For all TBoxes $\mathcal{T}$ and $\mathcal{T}'$, and all ABoxes $\mathcal{A}$ and $\mathcal{A}'$,
  - $\mathcal{A} \subseteq \mathsf{sat}(\mathcal{T} \cup \mathcal{T}', \emptyset)$ implies $\mathsf{sat}(\mathcal{T}, \mathcal{A}) \subseteq \mathsf{sat}(\mathcal{T} \cup \mathcal{T}', \emptyset)$, and
  - $\mathcal{A}' \subseteq \mathsf{sat}(\mathcal{T}, \mathcal{A})$ implies $\mathsf{sat}(\mathcal{T}, \mathcal{A} \cup \mathcal{A}') \subseteq \mathsf{sat}(\mathcal{T}, \mathcal{A})$.

[1] Note that this notion of basic concept differs from the one introduced in [6].

The algorithm $\mathsf{el}(\mathcal{T})$ accepts a TBox $\mathcal{T}$ and returns $C \sqsubseteq D$ for each pair of concepts C and D from $\mathsf{sig}(\mathcal{T}) \cup \{\top, \bot\}$ such that $D(x_C) \in \mathsf{sat}(\mathcal{T}, \emptyset)$. Properties **P1** and **P2** straightforwardly imply that $C \sqsubseteq D \in \mathsf{el}(\mathcal{T})$ iff $\mathcal{T} \models C \sqsubseteq D$.

2.3 Locality

When a TBox $\mathcal{T}_v$ reuses a TBox $\mathcal{T}_h$, it is commonly accepted [1, 2] that $\mathcal{T}_v$ should not affect the meaning of the symbols reused from $\mathcal{T}_h$—that is, $\mathcal{T}_v \cup \mathcal{T}_h \models \alpha$ should imply $\mathcal{T}_h \models \alpha$ for each axiom α containing only the reused symbols. This is guaranteed if $\mathcal{T}_v$ is local w.r.t. the set of concepts and roles Γ imported from $\mathcal{T}_h$ [2]. More precisely, for Γ a signature such that $\{\top, \bot\} \subseteq \Gamma$, a TBox $\mathcal{T}$ is *local* w.r.t. Γ if, for each interpretation I such that $X^I = \emptyset$ for each atomic concept and each atomic role $X \notin \Gamma$, we have $I \models \mathcal{T}$.

An $\mathcal{EL}$ TBox $\mathcal{T}$ in normal form is *safe* for Γ if $\mathsf{sig}(C_1 \sqcap \ldots \sqcap C_n) \not\subseteq \Gamma$ for each axiom of the form (1) in $\mathcal{T}$. It is easy to see that $\mathcal{T}$ is then local for Γ.

3 Importing Ontologies by Query

To illustrate the notion of import-by-query, Table 2 shows a TBox $\mathcal{T}_h$ whose axioms are to be kept hidden, but that is reused in a visible TBox $\mathcal{T}_v$. The TBox $\mathcal{T}_h$ provides concepts describing the structure of organs such as **Heart**, and medical conditions associated to them such as **CHD** (congenital heart defect), **VSD** (ventricular septum defect), and **AS** (aortic stenosis). Furthermore, the role **part** relates organs with their parts, and **cond** relates organs with medical conditions. The former role is used to define concepts such as **Heart** (an organ one of whose parts is the tricuspid valve); the latter 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 $\mathcal{T}_h$ are written in bold font. The TBox $\mathcal{T}_h$ might also contain nonshared symbols; however, for simplicity, we do not show any axioms involving such symbols. The TBox $\mathcal{T}_v$ provides the concept *Pat* representing patients, and it defines types of patients by relating the organs from $\mathcal{T}_h$ with the patients using the *hasOrgan* role. In addition, $\mathcal{T}_v$ extends the list of defects in $\mathcal{T}_h$ by *EA* (Ebstein's anomaly), which is a congenital heart defect in which the opening of the tricuspid valve is displaced (axioms δ_7 and δ_8), and defines a patient with tricuspid valve disease (*TVD_Pat*) as a patient with a heart that has a congenital heart defect and an abnormal tricuspid valve (δ_9). The symbols private to $\mathcal{T}_v$ are written in italic font.

As already mentioned, we assume that $\mathcal{T}_v$ is local w.r.t. the imported symbols. For example, δ_1 is local w.r.t. {**CHD_Heart**} because δ_1 is satisfied in any interpretation that interprets the nonshared symbols as $\emptyset$. Note also that, according to the definition in Section 2, $\mathcal{T}_v$ is safe for the imported signature.

We next introduce two types of *oracles*, which are responsible for advertising the shared signature Γ of $\mathcal{T}_h$ and answering queries w.r.t. $\mathcal{T}_h$.

Table 2. Example Knowledge Bases

Hidden Knowledge Base $\mathcal{T}_h$	
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	$\mathbf{AS} \sqsubseteq \mathbf{CHD}$
γ_5	$\mathbf{Heart} \sqsubseteq \mathbf{Organ} \sqcap \exists \mathbf{part}.\mathbf{Tric\_Valve}$
Visible Knowledge Base $\mathcal{T}_v$	
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	$\mathit{EA\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.\mathit{EA\_Heart}$
δ_5	$\mathit{EA} \sqsubseteq \mathbf{CHD}$
δ_6	$\mathit{EA\_Heart} \equiv \mathbf{Heart} \sqcap \exists \mathbf{cond}.\mathit{EA}$
δ_7	$\mathbf{Heart} \sqcap \exists \mathbf{cond}.\mathit{EA} \sqsubseteq \exists \mathbf{part}.\mathit{Ab\_Tric\_Valve}$
δ_8	$\mathit{Ab\_Tric\_Valve} \equiv \mathbf{Tric\_Valve} \sqcap \exists \mathbf{part}.\mathit{Displ\_Opening}$
δ_9	$\mathit{TVD\_Pat} \equiv \mathit{Pat} \sqcap \exists \mathit{hasOrgan}.(\mathbf{CHD\_Heart} \sqcap \mathbf{part}.\mathit{Ab\_Tric\_Valve})$

Definition 1. *Let $\mathcal{T}$ be an $\mathcal{EL}$ TBox and Γ be s.t. $\{\top, \bot\} \subseteq \Gamma \subseteq \mathsf{sig}(\mathcal{T})$. A* subsumption *oracle $\Omega^{\mathsf{s}}_{\mathcal{T},\Gamma}$ is a function that, for each pair of $\mathcal{EL}$ concepts C and D with $\mathsf{sig}(C) \subseteq \Gamma$ and $\mathsf{sig}(D) \subseteq \Gamma$, returns a Boolean value as follows:*

$$\Omega^{\mathsf{s}}_{\mathcal{T},\Gamma}(C, D) = \begin{cases} \mathsf{t} & \text{if } \mathcal{T} \models C \sqsubseteq D \\ \mathsf{f} & \text{otherwise} \end{cases}$$

An ABox query *oracle $\Omega^{\mathsf{a}}_{\mathcal{T},\Gamma}$ is a function that, for each normalized $\mathcal{EL}$ ABox $\mathcal{A}$ with $\mathsf{sig}(\mathcal{A}) \subseteq \Gamma$, a concept $C \in \mathsf{BC}(\Gamma)$, and an individual x occurring in $\mathcal{A}$, returns a Boolean value as follows:*

$$\Omega^{\mathsf{a}}_{\mathcal{T},\Gamma}(\mathcal{A}, C, x) = \begin{cases} \mathsf{t} & \text{if } \langle \mathcal{T}, \mathcal{A} \rangle \models C(x) \\ \mathsf{f} & \text{otherwise} \end{cases}$$

Without any further qualification, the generic term *oracle* refers either to the subsumption of the ABox query oracle.

A subsumption oracle is able to answer subsumption queries between (possibly complex) $\mathcal{EL}$ concepts, and it is analogous to the concept satisfiability oracle we considered in [5]. Concept subsumption is available as a reasoning service in all DL reasoners known to us, so it provides us with a natural oracle query language. Our results from [5], however, suggest that the way in which the roles from $\mathcal{T}_h$ are used in $\mathcal{T}_v$ must be restricted in import-by-query algorithms based on subsumption oracles; we summarize these results in Section 4.

As we show in Section 5, the use of ABox query oracles allows us to overcome these limitations. Intuitively, an ABox query accepts an ABox $\mathcal{A}$ containing only symbols in Γ and "completes it" with the (basic) concepts over Γ that are entailed by $\mathcal{A} \cup \mathcal{T}_h$. Roughly speaking, the ABox represents the information that has already been deduced; when "handed over" to the oracle, the oracle then completes the ABox with the relevant information entailed by $\mathcal{T}_h$.

An import-by-query (classification) algorithm checks the subsumption relations between the atomic concepts in $\mathsf{sig}(\mathcal{T}_v)$ w.r.t. $\mathcal{T}_v \cup \mathcal{T}_h$ by using an oracle.

Definition 2. *An* import-by-query algorithm *takes as input an $\mathcal{EL}$ TBox $\mathcal{T}_v$ and an oracle $\Omega_{\mathcal{T}_h,\Gamma}$ with $\mathsf{sig}(\mathcal{T}_v) \cap \mathsf{sig}(\mathcal{T}_h) \subseteq \Gamma$, and it returns $C \sqsubseteq D$ for each C and D such that $\{C, D\} \subseteq \mathsf{sig}(\mathcal{T}_v) \cup \{\top, \bot\}$ and $\mathcal{T}_v \cup \mathcal{T}_h \models C \sqsubseteq D$.*

4 Import-by-Query and Subsumption Oracles

We next summarize our results from [5]. We first show that no general import-by-query algorithm based on subsumption oracles exists.

Theorem 3. *No import-by-query algorithm based on subsumption oracles exists, if $\mathcal{T}_v$ and $\mathcal{T}_h$ are in $\mathcal{EL}$, Γ contains only one atomic role, and $\mathcal{T}_v$ is local in Γ.*

Proof. Assume that an import-by-query algorithm exists that terminates on all inputs, and consider $\mathcal{T}_v$ as in (2) with $\Gamma = \{r\}$. Clearly, $\mathcal{T}_v$ is local w.r.t. Γ. Since the algorithm terminates, the number of questions posed to any subsumption oracle is bounded by some integer m, and the quantifier depth of each concept C passed to the oracle is bounded by an integer n, where both m and n depend only on Γ and $\mathcal{T}_v$. Let $\mathcal{T}_h^1$ and $\mathcal{T}_h^2$ be as in (3) and (4), respectively.

$$\mathcal{T}_v = \{A \sqsubseteq \exists r.A\} \tag{2}$$

$$\mathcal{T}_h^1 = \emptyset \tag{3}$$

$$\mathcal{T}_h^2 = \{ \underbrace{\exists r. \ldots \exists r}_{n+1 \text{ times}} .\top \sqsubseteq \bot \} \tag{4}$$

For each C with $\mathsf{sig}(C) \subseteq \Gamma$ and quantifier depth at most n , $\mathcal{T}_h^1 \models C \sqsubseteq \bot$ iff $\mathcal{T}_h^2 \models C \sqsubseteq \bot$, so $\Omega^{\mathsf{s}}_{\mathcal{T}_h^1,\Gamma}(C, \bot) = \Omega^{\mathsf{s}}_{\mathcal{T}_h^2,\Gamma}(C, \bot)$. When applied to $\mathcal{T}_v$ and $\Omega^{\mathsf{s}}_{\mathcal{T}_h^1,\Gamma}$, the algorithm returns the same value as when applied to $\mathcal{T}_v$ and $\Omega^{\mathsf{s}}_{\mathcal{T}_h^2,\Gamma}$. Since $\mathcal{T}_v \cup \mathcal{T}_h^1 \not\models A \sqsubseteq \bot$ but $\mathcal{T}_v \cup \mathcal{T}_h^2 \models A \sqsubseteq \bot$, it does not satisfy Definition 2. □

This negative result relies on the presence of an atomic role r in the reused signature Γ and the fact that r is used in $\mathcal{T}_v$ in a "cyclic" axiom $A \sqsubseteq \exists r.A$. To circumvent this problem, we studied in [5] the case where Γ contains only atomic concepts and proposed an import-by-query algorithm that applies even when $\mathcal{T}_v$ and $\mathcal{T}_h$ are given in a very expressive DL.

In [5] we also studied the case where both concepts and roles are imported, but with the condition that $\mathsf{sig}(C) \subseteq \Gamma$ for each concept C appearing in $\mathcal{T}_v$ in the scope of a quantifier over r from Γ. For example, if $r \in \Gamma$, then $\mathcal{T}_v$ is allowed contain the concept $\exists r.A$ only if $A \in \Gamma$. The algorithm in [5] for this case applies to very expressive DLs. In our example, the conditions in [5] allow us to write axioms δ_1–δ_5, which together with $\mathcal{T}_h$ allow us to conclude $VSD\_Pat \sqsubseteq CHD\_Pat$ (using δ_1, δ_2 and δ_5) and $AS\_Pat \sqsubseteq CHD\_Pat$ (using δ_1 and δ_3).

In order to express axioms such as δ_5–δ_9, we next investigate ways of relaxing the conditions from [5]. In Section 5, we study the design of import-by-query

Table 3. Additional $\mathcal{EL}$ Rule for an ABox Query Oracle

R3	If for some individual x_C in $\mathcal{A}_i$ and concept $D \in \mathsf{BC}(\Gamma)$, $\Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma}(\mathcal{A}_i|_\Gamma, D, x_C) = \mathsf{t}$ and $D(x_C) \notin \mathcal{A}_i$ then $\mathcal{A}_{i+1} := \mathcal{A}_i \cup \{D(x_C)\}$.

algorithms based on ABox query oracles. We show that, if $\mathcal{T}_v$ and $\mathcal{T}_h$ are in $\mathcal{EL}$, no restrictions other than locality are needed to ensure existence of an import-by-query algorithm. This investigation will also provide us with insight on how to relax the syntactic restrictions from [5] when using subsumption oracles.

5 Import-by-Query with ABox Query Oracles

We next present the import-by-query algorithm $\mathsf{ibq}^{\mathsf{a}}(\mathcal{T}_v, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$, which accepts a normalized $\mathcal{EL}$ TBox $\mathcal{T}_v$ and an ABox query oracle $\Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma}$ s.t. $\mathcal{T}_v$ is safe for Γ.

Let $\mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \mathcal{A}, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ be the same as $\mathsf{sat}(\mathcal{T}_v, \mathcal{A})$ (see Section 2), with the difference that rule R3 shown Table 3 is applied alongside rules R1 and R2 from Table 1. We next prove certain properties of this algorithm.

Lemma 4. *Let $\mathcal{A}^{\mathsf{a}} = \mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ and $\mathcal{A}^{\mathsf{el}} = \mathsf{sat}(\mathcal{T}_v \cup \mathcal{T}'_h, \emptyset)$, where $\mathcal{T}'_h$ is the result of transforming $\mathcal{T}_h$ into normal form. The following properties hold:*

- Soundness: $\mathcal{A}^{\mathsf{a}} \subseteq \mathcal{A}^{\mathsf{el}}$*;*
- Completeness: *For each $C \in \mathsf{sig}(\mathcal{T}_v)$ and $D \in \mathsf{BC}(\mathsf{sig}(\mathcal{T}_v))$, we have that $D(x_C) \in \mathcal{A}^{\mathsf{el}}$ implies $D(x_C) \in \mathcal{A}^{\mathsf{a}}$.*

Proof (Soundness). Let $\mathcal{A}_0, \ldots, \mathcal{A}_n$ with $\mathcal{A}_n = \mathcal{A}^{\mathsf{a}}$ be a run of $\mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$. We show by induction on the rule applications that $\mathcal{A}_i \subseteq \mathcal{A}^{\mathsf{el}}$ for each $0 \leq i \leq n$. The induction base ($i = 0$) follows from the initialization conditions of both algorithms. Assume now that $\mathcal{A}_{i-1} \subseteq \mathcal{A}^{\mathsf{el}}$ and let $\mathcal{A}_i$ be obtained from $\mathcal{A}_{i-1}$ by a rule application. If either R1 or R2 is applied, property **P2** from Section 2 implies that $\mathcal{A}_i \subseteq \mathcal{A}^{\mathsf{el}}$. Assume that R3 extends $\mathcal{A}_{i-1}$ with $D(x_C)$. Then, $C \in \Gamma$; $D \in \mathsf{BC}(\Gamma)$; and, by the induction hypothesis, $\mathcal{A}_{i-1}|_\Gamma \subseteq \mathcal{A}^{\mathsf{el}}$. Also, $\Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma}(\mathcal{A}_{i-1}|_\Gamma, D, x_C) = \mathsf{t}$, which implies $\mathcal{T}'_h \cup \mathcal{A}_{i-1}|_\Gamma \models D(x_C)$. But then, property **P1** from Section 2 implies that $D(x_C) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}_{i-1}|_\Gamma)$. Since $\mathcal{A}_{i-1}|_\Gamma \subseteq \mathcal{A}^{\mathsf{el}}$, by **P2** we have that $\mathsf{sat}(\mathcal{T}'_h, \mathcal{A}_{i-1}|_\Gamma) \subseteq \mathcal{A}^{\mathsf{el}}$, which implies our claim. □

Proof (Completeness). Let $\mathcal{A}_0, \ldots, \mathcal{A}_n$ with $\mathcal{A}_n = \mathcal{A}^{\mathsf{el}}$ be a run of $\mathsf{sat}(\mathcal{T}_v \cup \mathcal{T}'_h, \emptyset)$. For $\mathcal{A}_j$ occurring in the run and Σ a signature, let

$$\mathcal{A}_j\|_\Sigma = \{D(x_C) \in \mathcal{A}_j \mid \{C, D\} \subseteq \mathsf{BC}(\Sigma)\} \cup \{r(x_C, x_D) \in \mathcal{A}_j \mid \{r, C, D\} \subseteq \Sigma\}.$$

Completeness follows immediately from statement 2 of the following claim.

CLAIM: The following conditions hold for each $\mathcal{A}_j$ with $1 \leq j \leq n$:

1. $\mathcal{A}_j \subseteq \mathcal{A}^{\mathsf{a}} \cup \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$.

2. $\mathcal{A}_j\|_{\mathsf{sig}(\mathcal{T}_v)} \subseteq \mathcal{A}^{\mathsf{a}}$.
3. If $D(x_C) \in \mathcal{A}_j$, then
   (a) $C \in \mathsf{sig}(\mathcal{T}'_h)$ implies $D \in \mathsf{BC}(\mathsf{sig}(\mathcal{T}'_h))$, and
   (b) $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ implies $D \in \mathsf{sig}(\mathcal{T}_v)$.
4. If $r(x_C, x_D) \in \mathcal{A}_j$, then
   (a) $r \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ implies $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$,
   (b) $r \in \Gamma$ and $D \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ imply $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$, and
   (c) $r \in \mathsf{sig}(\mathcal{T}_v)$ and $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ imply $D \in \mathsf{sig}(\mathcal{T}_v)$.

The proof is by induction on the rule applications. For the induction base ($j = 0$), we have $\mathcal{A}_0 = \{C(x_C) \mid C \in \mathsf{sig}(\mathcal{T}_v \cup \mathcal{T}'_h)\}$. Statements 1–3 hold straightforwardly, and statement 4 is vacuously true. Assume now that statements 1–4 hold for $\mathcal{A}_{j-1}$ and consider an application of a rule that derives $\mathcal{A}_j$.

Assume that R1 is applied to $\exists r.D(x_C) \in \mathcal{A}_{j-1}$ and that it derives $r(x_C, x_D)$. By the induction hypothesis, $\exists r.D(x_C) \in \mathcal{A}^{\mathsf{a}}$ or $\exists r.D(x_C) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$. If $\exists r.D(x_C) \in \mathcal{A}^{\mathsf{a}}$, then $r(x_C, x_D) \in \mathcal{A}^{\mathsf{a}}$ because R1 is not applicable to $\mathcal{A}^{\mathsf{a}}$, so statement 1 holds. If $\exists r.D(x_C) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$, then $r(x_C, x_D) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$ because R1 is not applicable to $\mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$, so statement 1 also holds. If C, D, and r are all in $\mathsf{sig}(\mathcal{T}_v)$, by the induction hypothesis $\exists r.D(x_C) \in \mathcal{A}^{\mathsf{a}}$; since R1 is not applicable to $\mathcal{A}^{\mathsf{a}}$, we have $r(x_C, x_D) \in \mathcal{A}^{\mathsf{a}}$, so statement 2 holds. Statement 3 holds vacuously. Note that the contrapositive of statement 3a is as follows (*): $r \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ or $D \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$ imply $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$. Now (*) immediately implies statements 4a and 4b. Finally, statement 3b implies statement 4c.

Assume now that R2 is applied to an axiom $\alpha \in \mathcal{T}_v \cup \mathcal{T}'_h$ of the form $A_1 \sqcap \ldots \sqcap A_k \sqcap \exists r_1.B_1 \sqcap \ldots \sqcap \exists r_m.B_m \sqsubseteq D$ and that it derives $D(x_C)$. Then, individuals $y_{E_1}, \ldots, y_{E_m}$ in $\mathcal{A}_{j-1}$ exist such that $A_i(x_C) \in \mathcal{A}_{j-1}$ for each $1 \leq i \leq k$, and $\{r_\ell(x_C, y_{E_\ell}), B_\ell(y_{E_\ell})\} \subseteq \mathcal{A}_{j-1}$ for each $1 \leq \ell \leq m$. Statement 4 holds vacuously, so we next prove 1–3, depending on whether α belongs to $\mathcal{T}'_h$ or $\mathcal{T}_v$.

- $\alpha \in \mathcal{T}'_h$. Statement 3 holds trivially. Consider each assertion $A_i(x_C)$ with $1 \leq i \leq n$. By the induction hypothesis, we have $A_i(x_C) \in \mathcal{A}^{\mathsf{a}} \cup \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$. If $A_i(x_C) \in \mathcal{A}^{\mathsf{a}}$, since $\alpha \in \mathcal{T}'_h$, we have $A_i \in \Gamma$, so $A_i(x_C) \in \mathcal{A}^{\mathsf{a}}|_\Gamma$. Thus, we have $A_i(x_C) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$. By a completely analogous argument, we have $\{r_\ell(x_C, y_{E_\ell}), B_\ell(y_{E_\ell})\} \subseteq \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$ for each $1 \leq \ell \leq m$ as well. But then, since R2 is not applicable to $\mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$, we have $D(x_C) \in \mathsf{sat}(\mathcal{T}'_h, \mathcal{A}^{\mathsf{a}}|_\Gamma)$, so statement 1 holds. Statement 2 is vacuously true if $C \notin \Gamma$ or $D \notin \mathsf{BC}(\Gamma)$. Otherwise, by statement 1 and property **P1**, we have $\mathcal{T}'_h \cup \mathcal{A}^{\mathsf{a}}|_\Gamma \models D(x_C)$; since R3 is not applicable to $\mathcal{A}^{\mathsf{a}}$, we have $D(x_C) \in \mathcal{A}^{\mathsf{a}}$, so statement 2 holds.
- $\alpha \in \mathcal{T}_v$. Note that $\mathcal{T}_v$ is safe for Γ, so we have these possibilities:
  - $A_i \notin \Gamma$ for some $1 \leq i \leq k$. By the contrapositive of statement 3a, we have $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$.
  - $r_\ell \notin \Gamma$ for some $1 \leq \ell \leq m$. By statement 4a, we have $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$.
  - $B_\ell \notin \Gamma$ and $r_\ell \in \Gamma$ for some $1 \leq \ell \leq m$. By the contrapositive of statement 3a, $y_{E_\ell} \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$; however, by statement 4b, $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$.

  Since $C \in \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$, statement 4c implies $E_\ell \in \mathsf{sig}(\mathcal{T}_v)$. But then, by statement 2, all $A_i(x_C)$, $r_\ell(x_C, y_{E_\ell})$, and $B_\ell(y_{E_\ell})$ are in $\mathcal{A}^{\mathsf{a}}$. Since R2 is not applicable to $\mathcal{A}^{\mathsf{a}}$, $D(x_C) \in \mathcal{A}^{\mathsf{a}}$, which implies statements 1 and 2. Statements 3a and 3b respectively hold since $C \notin \mathsf{sig}(\mathcal{T}'_h)$ and $D \in \mathsf{BC}(\mathsf{sig}(\mathcal{T}_v))$. □

The algorithm $\mathsf{ibq}^{\mathsf{a}}(\mathcal{T}_v, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ returns $C \sqsubseteq D$ for each pair of concepts C and D from $\mathsf{sig}(\mathcal{T}) \cup \{\top, \bot\}$ such that $D(x_C) \in \mathsf{sat}^{\mathsf{a}}(\mathcal{T}, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$.

Theorem 5. *The algorithm $\mathsf{ibq}^{\mathsf{a}}(\mathcal{T}_v, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ is an import-by-query algorithm, and it can be implemented such that it runs in time polynomial in the size of $\mathcal{T}_v$ with a polynomial number of calls to $\Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma}$.*

Proof. The size of $\mathsf{BC}(\mathsf{sig}(\mathcal{T}_v))$ is quadratic in the size of $\mathcal{T}_v$. Each application of R1, R2, or R3 adds an assertion of the form $C(a)$ or $r(a, b)$ for $C \in \mathsf{BC}(\mathsf{sig}(\mathcal{T}_v))$. Since no rule removes assertions from $\mathcal{A}$, the total number of rule applications is polynomial. Thus, the algorithm can be implemented such that it runs in polynomial time with a polynomial number of calls to the oracle. Since $\mathcal{T}'_h$ is obtained from $\mathcal{T}_h$ through normalization, for all concepts C and D with $\{C, D\} \subseteq \mathsf{sig}(\mathcal{T}_h) \cup \mathsf{sig}(\mathcal{T}_v) \cup \{\top, \bot\}$ we have $\mathcal{T}_v \cup \mathcal{T}_h \models C \sqsubseteq D$ iff $\mathcal{T}_v \cup \mathcal{T}'_h \models C \sqsubseteq D$. Let $\mathcal{A}^{\mathsf{a}}$ and $\mathcal{A}^{\mathsf{el}}$ be as in Lemma 4. If $\mathsf{ibq}^{\mathsf{a}}(\mathcal{T}_v, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ returns $C \sqsubseteq D$, then $D(x_C) \in \mathcal{A}^{\mathsf{a}}$; the soundness property of Lemma 4 implies $D(x_C) \in \mathcal{A}^{\mathsf{el}}$; thus, $\mathsf{el}(\mathcal{T}_v \cup \mathcal{T}'_h)$ returns $C \sqsubseteq D$; hence, $\mathcal{T}_v \cup \mathcal{T}'_h \models C \sqsubseteq D$; finally, $\mathcal{T}_v \cup \mathcal{T}_h \models C \sqsubseteq D$. The converse direction holds analogously by the completeness property of Lemma 4. □

6 Weakly Acyclic Knowledge Bases

We next present an import-by-query algorithm based on subsumption oracles where $\mathcal{T}_v$ needs to satisfy weaker conditions than those in [5]. Our results rely on the notion of *weak acyclicity* borrowed from database dependency theory [9]. This condition prohibits cyclic existential quantification over the roles imported from $\mathcal{T}_h$, which invalidates the proof of Theorem 3. In our example, weak acyclicity allow us to represent all axioms δ_1–δ_9 and hence to entail $EA\_Pat \sqsubseteq CHD\_Pat$ (using δ_4–δ_6) and $EA\_Pat \sqsubseteq TVD\_Pat$ (using δ_4–δ_7 and δ_9).

6.1 Weak Acyclicity

Let $[A] = A$ for A an atomic concept, and let $[\exists r.D] = r$.

Definition 6. *Let Γ be a signature such that $\{\top, \bot\} \subseteq \Gamma$, and let $\mathcal{T}$ be an $\mathcal{EL}$ TBox in normal form. The* dependency graph *for $\mathcal{T}$ w.r.t. Γ is the smallest graph that contains a node w_X for each symbol $X \in \mathsf{sig}(\mathcal{T})$ and the following edges for each axiom $C_1 \sqcap \ldots \sqcap C_n \sqsubseteq D$ in $\mathcal{T}$ and each $1 \leq i \leq n$:*

- *an unlabeled edge $\langle w_{[C_i]}, w_{[D]} \rangle$, and*
- *another edge $\langle w_{[C_i]}, w_{[E]} \rangle$ labeled with $\star$ if D of the form $\exists r.E$ with $r \in \Gamma$.*

$\mathcal{T}$ is weakly acyclic *w.r.t. Γ if, in each cycle in the dependency graph for $\mathcal{T}$ w.r.t. Γ, all edges are unlabeled.*

It can be readily checked using Definition 6 that our example TBox $\mathcal{T}_v$ in Table 2 is weakly acyclic w.r.t. the imported signature.

Table 4. Additional $\mathcal{EL}$ Rule for oracle Ω^{s}

R3'	If for some individual x_C in $\mathcal{A}_i$ and concept $D \in \mathsf{BC}(\Gamma)$, $\Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma}(\mathsf{con}(x_C, \mathcal{A}_i, \Gamma), D) = \mathsf{t}$ and $D(x_C) \notin \mathcal{A}_i$ then $\mathcal{A}_{i+1} := \mathcal{A}_i \cup D(x_C)$.

Let Γ be a signature and $\mathcal{A}$ an ABox occurring in a run of $\mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$. Then $\mathcal{A}$ contains a *harmful cycle* for Γ if assertions $r_i(x_{C_{i-1}}, x_{C_i}) \in \mathcal{A}$ with $1 \leq i \leq n$ exist such that $x_{C_n} = x_{C_0}$, $C_i \in \mathsf{sig}(\mathcal{T}_v)$ for each $0 \leq i \leq n$, and $r_i \in \Gamma$ for each $1 \leq i \leq n$. As we show next, if $\mathcal{T}_v$ is weakly acyclic w.r.t. Γ, then no ABox occurring in a run of $\mathsf{sat}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ contains a cycle harmful for Γ.

Lemma 7. *Let $\mathcal{T}_v$ be weakly acyclic w.r.t. Γ, and let $\mathcal{A}^{\mathsf{a}} = \mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$. Then, $\mathcal{A}^{\mathsf{a}}$ does not contain a cycle that is harmful for Γ.*

Proof (Sketch). Let $\mathcal{G}$ be the dependency graph for $\mathcal{T}_v$ w.r.t. Γ, and $\mathcal{A}_0, \ldots, \mathcal{A}_n$ a run of $\mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ such that $\mathcal{A}_n = \mathcal{A}^{\mathsf{a}}$. The following claim can be straightforwardly shown by induction on the number of rule applications:

CLAIM: The following properties hold for each $\mathcal{A}_i$ with $1 \leq i \leq n$:

1. if $D(x_C) \in \mathcal{A}_i$ and $\{C, D\} \subseteq \mathsf{BC}(\mathsf{sig}(\mathcal{T}_v) \setminus \Gamma)$, then $w_{[D]}$ is reachable in $\mathcal{G}$ from $w_{[C]}$; in addition, if D is of the form $\exists r.E$, then $w_{[E]}$ is reachable in $\mathcal{G}$ from $w_{[C]}$ via a path containing a labeled edge;
2. if $r(x_C, x_D) \in \mathcal{A}_i$, $\{C, D\} \subseteq \mathsf{sig}(\mathcal{T}_v) \setminus \Gamma$, and $r \in \Gamma$, then $w_{[D]}$ is reachable from $w_{[C]}$ via a path containing a labeled edge.

The lemma follows from statement 2 and the fact that $\mathcal{T}_v$ is weakly acyclic. □

6.2 An Import-by-Query Algorithm

The algorithm $\mathsf{ibq}^{\mathsf{s}}(\mathcal{T}_v, \Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma})$ accepts a normalized $\mathcal{EL}$ TBox $\mathcal{T}_v$ that is safe and weakly acyclic for Γ, and an oracle $\Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma}$. Let $\mathsf{sat}^{\mathsf{s}}(\mathcal{T}_v, \mathcal{A}, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$ be the same as $\mathsf{sat}(\mathcal{T}_v, \mathcal{A})$ with the difference that rule R3' in Table 4 is applied alongside rules R1 and R2. For Σ a signature and $\mathcal{A}$ an ABox not containing cycles harmful for Σ, the concept $\mathsf{con}(x_C, \mathcal{A}, \Sigma)$ used in rule R3' is inductively defined as follows:

$$\mathsf{con}(x_C, \mathcal{A}, \Sigma) = \begin{cases} \bigsqcap_{D(x_C) \in \mathcal{A}|_\Sigma} D & \text{if } C \in \Sigma \\ \bigsqcap_{D(x_C) \in \mathcal{A}|_\Sigma} D \;\sqcap \bigsqcap_{r(x_C, y_E) \in \mathcal{A}|_\Sigma} \exists r.\mathsf{con}(y_E, \mathcal{A}, \Sigma) & \text{otherwise} \end{cases}$$

Since $\mathcal{A}$ does not contain cycles harmful for Σ, induction is well-founded. Intuitively, $\mathsf{con}(x_C, \mathcal{A}, \Sigma)$ represents the "rolled-up" part of $\mathcal{A}|_\Sigma$ reachable from x_C. The algorithm $\mathsf{ibq}^{\mathsf{s}}(\mathcal{T}_v, \Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma})$ then returns $C \sqsubseteq D$ for each pair of concepts C and D such that $\{C, D\} \subseteq \mathsf{sig}(\mathcal{T}) \cup \{\top, \bot\}$ and $D(x_C) \in \mathsf{sat}^{\mathsf{s}}(\mathcal{T}, \emptyset, \Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma})$.

Lemma 8. *For each signature Γ, each $\mathcal{EL}$ TBox $\mathcal{T}_v$ that is safe and weakly acyclic for Γ, and each $\mathcal{EL}$ TBox $\mathcal{T}_h$ such that $\mathsf{sig}(\mathcal{T}_v) \cap \mathsf{sig}(\mathcal{T}_h) \subseteq \Gamma$, we have $\mathsf{sat}^{\mathsf{s}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma}) = \mathsf{sat}^{\mathsf{a}}(\mathcal{T}_v, \emptyset, \Omega^{\mathsf{a}}_{\mathcal{T}_h,\Gamma})$.*

Proof (Sketch). The lemma is a consequence of the following claim together with the fact that both algorithms are identical except for the rule R3'.

CLAIM: For each signature Γ, ABox $\mathcal{A}$ not containing cycles harmful for Γ, individual x in $\mathcal{A}$, and concept $D \in \mathsf{BC}(\Gamma)$, we have

$$\mathcal{T}_h \cup \mathcal{A}|_\Gamma \models D(x) \text{ if and only if } \mathcal{T}_h \models \mathsf{con}(x, \mathcal{A}, \Gamma) \sqsubseteq D.$$

To prove this claim, let Q_x be a fresh concept uniquely associated with each individual x in $\mathcal{A}|_\Gamma$. Let $\mathcal{T}'$ be a TBox containing the axiom $Q_{x_C} \sqsubseteq D$ for each $D(x_C) \in \mathcal{A}|_\Gamma$, and $Q_{x_D} \sqsubseteq \exists r.Q_{y_E}$ for each $r(x_D, y_E) \in \mathcal{A}|_\Gamma$ with $D \notin \Gamma$. It is easy to show that $\mathcal{T}_h \cup \mathcal{A}|_\Gamma \models D(x)$ iff $D(x) \in \mathsf{el}(\mathcal{T}_h' \cup \mathcal{T}')$, for $\mathcal{T}_h'$ the normalized version of $\mathcal{T}_h$. Since $\mathcal{A}|_\Gamma$ does not contain cycles harmful for Γ, by induction on the structure of $\mathsf{con}(x, \mathcal{A}, \Gamma)$, we have that $\mathsf{con}(x, \mathcal{A}, \Gamma)$ is obtained from Q_x by unfolding the relevant concepts Q_y with their definitions in $\mathcal{T}'$. □

Theorem 9. *The algorithm* $\mathsf{ibq}^{\mathsf{s}}(\mathcal{T}_v, \Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma})$ *is an import-by-query algorithm and it can be implemented such that it runs in time exponential in the size of* $\mathcal{T}_v$ *with a polynomial number of calls to* $\Omega^{\mathsf{s}}_{\mathcal{T}_h,\Gamma}$.

Proof. The algorithm is an import-by-query algorithm by Lemma 8 and Theorem 5. Finally, just like in the proof of Theorem 5, rules R1, R2, and R3' can only be applied a polynomial number of times. The inductive definition of $\mathsf{con}(x, \mathcal{A}, \Gamma)$, however, involves constructing a tree of polynomial depth and branching factor, the size of which can be at most exponential in the size of $\mathcal{A}|_\Gamma$. □

7 Related and Future Work

In a P2P setting, [4] consider the problem of answering a query q over KBs $\mathcal{K}_v$ and $\mathcal{K}_h$ and mappings $\mathcal{M}$ 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 using an oracle for $\mathcal{K}_h$. In this setting, however, a satisfiable $\mathcal{K}_h$ cannot affect the subsumption of concepts in $\mathcal{K}_v$ (see [5] for an example); thus, the results in [4] are not applicable to schema reasoning.

Another possibility for the owner of $\mathcal{T}_h$ to restrict access only to symbols in Γ is to publish a *uniform interpolant* $\mathcal{T}_h^\Gamma$ of $\mathcal{T}_h$ w.r.t. Γ [11]—an ontology $\mathcal{T}_h^\Gamma$ such that $\mathsf{sig}(\mathcal{T}_h^\Gamma) \subseteq \Gamma$, $\mathcal{T}_h \models \mathcal{T}_h^\Gamma$ and $\mathcal{T}_h^\Gamma \models C \sqsubseteq D$ iff $\mathcal{T}_h \models C \sqsubseteq D$ for all concepts C and D with $\mathsf{sig}(C) \cup \mathsf{sig}(D) \subseteq \Gamma$. Publishing $\mathcal{T}_h^\Gamma$, however, may reveal more information about $\mathcal{T}_h$ than what is strictly needed. For example, for $\Gamma = \{r, B\}$, $\mathcal{T}_v = \{A \sqsubseteq \exists r.B\}$ and $\mathcal{T}_h = \{\exists r.\exists r.B \sqsubseteq B\}$, the interpolant $\mathcal{T}_h^\Gamma$ is equal to $\mathcal{T}_h$; thus, publishing $\mathcal{T}_h^\Gamma$ reveals entire contents of the hidden ontology $\mathcal{T}_h$. In contrast, an import-by-query algorithm reveals only that $\mathcal{T}_h \not\models \exists r.B \sqsubseteq \bot$ and $\mathcal{T}_h \not\models \exists r.B \sqsubseteq B$; hence it does not reveal the actual syntactic structure of $\mathcal{T}_h$. Thus, enabling the reuse of $\mathcal{T}_h$ through import-by-query preserves the hidden content of $\mathcal{T}_h$ to a higher degree than if reuse were enabled by publishing $\mathcal{T}_h^\Gamma$.

We are currently working on extending the results presented in this paper to DLs more expressive than $\mathcal{EL}$.

References

[1] Lutz, C., Walther, D., Wolter, F.: Conservative Extensions in Expressive Description Logics. In: Proc. IJCAI. (2007) 453–458

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

[3] Lenzerini, M.: Data Integration: A Theoretical Perspective. In: Proc. PODS. (2002) 233–246

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

[5] Cuenca Grau, B., Motik, B., Kazakov, Y.: Import-by-Query: Ontology Reasoning under Access Limitations. In: Proc. IJCAI, AAAI Press (2009) To Appear.

[6] Baader, F., Brandt, S., Lutz, C.: Pushing the $\mathcal{EL}$ Envelope. In: Proc. IJCAI,. (2005) 364–369

[7] Kutz, O., Horrocks, I., Sattler, U.: The Even More Irresistible $\mathcal{SROIQ}$. In: Proc. KR. (2006) 68–78

[8] Horrocks, I., Kutz, O., Sattler, U.: The irresistible $\mathcal{SRIQ}$. In: Proc. of OWLED. (2005)

[9] Kolaitis, P.G., Panttaja, J., Tan, W.C.: The complexity of data exchange. In: PODS. (2006) 30–39

[10] Motik, B., Horrocks, I.: Individual Reuse in Description Logic Reasoning. In: Proc. of IJCAR 2008. Volume 5195 of LNAI., Springer (2008) 242–258

[11] Konev, B., Walter, D., Wolter, F.: Forgetting and uniform interpolation in large-scale description logic terminologies. In: Proc. IJCAI, AAAI Press (2009) To Appear.