Absorption for ABoxes with Local Universal Restrictions

Jiewen Wu, Taras Kinash, David Toman, and Grant Weddell

Cheriton School of Computer Science
University of Waterloo, Canada
{j55wu, tkinash, david, gweddell}@uwaterloo.ca

Abstract. We elaborate on earlier work in which we developed a novel method for evaluating instance queries over DL knowledge bases that derives from binary absorption. An important feature of this earlier method and its refinement in this paper is that they avoid the need to check explicitly for consistency, a property that is desirable, for example, in SPARQL query evaluation over RDF data sets that can dynamically include sophisticated ontologies.

In particular, we resolve a number of outstanding issues with the earlier method that limited its capabilities for knowledge bases that involve an extensive use of *typing constraints* expressed as axioms of the form $A \sqsubseteq \forall R.B$, or that require and use both role hierarchies and transitive roles. We also show how our more general method supports a safe use of nominals in instance queries, and how the method can therefore be used to evaluate basic graph patterns in the SPARQL query language.

Finally, we present the results of a preliminary experimental evaluation that validates the efficacy of our more refined method for instance checking.

1 Introduction

In earlier work, we developed a novel method for instance checking over an $\mathcal{ALCIQ}(\mathbf{D})$ knowledge base [13, 14]. The method operated by reducing such problems to concept subsumption problems over an $\mathcal{ALCIOQ}(\mathbf{D})$ knowledge base. Roughly, this was achieved by introducing additional inclusion dependencies with nominals, and then by relying on a refinement of binary absorption [6] to ensure such dependencies are absorbed and thereby avoiding the overhead of reasoning about *arbitrary* $\mathcal{ALCIOQ}(\mathbf{D})$ terminologies that would otherwise be required.

This earlier method operates with the assumption that a knowledge base is consistent, and at no time requires any internal check to ensure this, for example, at the start of a session when the knowledge base is first loaded. This enables very fast load times and is also a useful feature in cases where an agent must evaluate a SPARQL query over a knowledge base for which consistency has already been established by other agents or, indeed, for which such a test is simply infeasible.

In this paper, we present a more refined version of this earlier method that addresses a number of outstanding issues that limited its effectiveness. In particular, our new method can now accommodate arbitrary $\mathcal{SHIQ}(\mathbf{D})$ knowledge

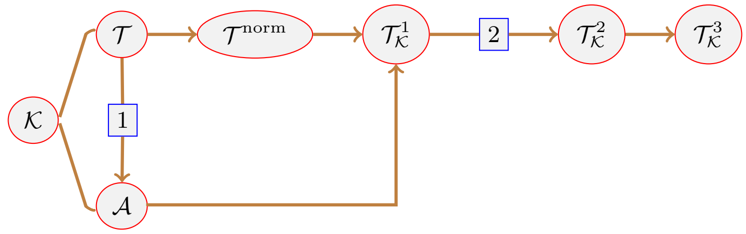

Fig. 1: Obtaining an ABox Absorption.

bases with role hierarchies and transitive roles, and is also more adept in cases that involve a more extensive use of *typing constraints*, in particular for local universal restrictions such as inclusion dependencies of the form $A \sqsubseteq \forall R.B$. We also show how this new method supports a safe use of nominals in instance queries, and how the method can therefore be used to help reduce the cost of evaluating basic graph patterns (BGPs) in the SPARQL query language.

Like the earlier method, our new method proceeds in a series of steps that ultimately obtains an absorbed $\mathcal{SHOIQ}(\mathbf{D})$ terminology $\mathcal{T}^3_{\mathcal{K}}$ from an input $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, as illustrated in Figure 1: a *normalized* TBox $\mathcal{T}^{\text{norm}}$ is first obtained from $\mathcal{T}$, essentially to *extract* embedded typing constraints, and then a series of three subsequent TBoxes $\mathcal{T}^i_{\mathcal{K}}$ are derived from $\mathcal{T}^{\text{norm}}$ and the ABox $\mathcal{A}$. Our main result is that an instance check of the form $\mathcal{K} \models a : C$ then maps to a subsumption check

$$\mathcal{T}^3_{\mathcal{K}} \models \{a\} \sqcap \mathcal{D} \sqsubseteq C, \tag{1}$$

where $\mathcal{D}$ is a concept that initializes an appropriate "firing" of binary absorptions in $\mathcal{T}^3_{\mathcal{K}}$.

The first and last steps in our new method are inherited unchanged from our earlier method [13], although we have included a more thorough presentation of $\mathcal{T}^{\text{norm}}$ in Section 2 in which we give our preliminary definitions. The remainder of the paper is organized as follows. Our primary contributions are in Section 3 in which we define the computation of $\mathcal{T}^1_{\mathcal{K}}$ and $\mathcal{T}^2_{\mathcal{K}}$ in Figure 1, and present our main result. We then show how nominals can *safely* occur in concept C in (1) above, that is, in a way that avoids any requirement for revising our method, and discuss how this can be useful in evaluating BGPs over $\mathcal{SHIQ}(\mathbf{D})$ knowledge bases. The results of a preliminary experimental evaluation of our new method are given in Section 4. These results are evidence that our new method is efficacious with respect to addressing the issues that were outstanding with the earlier version. Our review of related work and summary discussion then follow in Section 5. Part of this discussion relates to the two labelled arcs in Figure 1 in which we outline refinements to our method that can improve its performance or increase the scope of $\mathcal{SHIQ}(\mathbf{D})$ knowledge bases for which the method can be used.

2 Preliminaries

We consider instance checking problems over knowledge bases expressed in terms of the DL dialect $\mathcal{SHIQ}(\mathbf{D})$, where $\mathbf{D}$ is the simple concrete domain of finite length strings. However, such problems will be mapped to subsumption checking problems in the more general logic $\mathcal{SHOIQ}(\mathbf{D})$ in which nominals can occur in inclusion dependencies. Although not really necessary, our definition of $\mathcal{SHOIQ}(\mathbf{D})$ introduces a number of non-terminals in a concept grammar that helps to improve the clarity of the remainder of the paper.

Definition 1 (Description Logic $\mathcal{SHOIQ}(\mathbf{D})$).
$\mathcal{SHOIQ}(\mathbf{D})$ is a DL dialect based on disjoint infinite sets of atomic concepts NC, *atomic roles* NR, *concrete features* NF *and nominals* NI. *Let $S \in \mathrm{NR} \cup \{R^- \mid R \in \mathrm{NR}\}$ denote a general role. To avoid considering S^{--}, we define $S^- = R$ if $S = R^-$ and $S^- = R^-$ otherwise. A role inclusion is in the form of $S_1 \sqsubseteq S_2$. Let $\overset{*}{\sqsubseteq}$ be the transitive-reflexive closure of $\sqsubseteq$ over the set $\{S_1 \sqsubseteq S_2\} \cup \{S_1^- \sqsubseteq S_2^- \mid S_1 \sqsubseteq S_2\}$, a role S is transitive, denoted $\mathtt{Trans}(S)$, iff $\mathtt{Trans}(R)$ or $\mathtt{Trans}(R^-)$ for some R where $R \overset{*}{\sqsubseteq} S$ and $S \overset{*}{\sqsubseteq} R$. A role S is called* complex *if $\mathtt{Trans}(S')$ for some $S' \overset{*}{\sqsubseteq} S$.*

Let $A \in \mathrm{NC}$, $a \in \mathrm{NI}$, $f, g \in \mathrm{NF}$, and n be a non-negative integer, a $\mathcal{SHOIQ}(\mathbf{D})$ concept C is defined as follows:

$$\begin{array}{lcl} C & ::= & C_d \mid C \sqcap C \mid C \sqcup C \mid \{a\} \mid \neg\{a\} \mid \exists^{\leq n} S.C_1 \mid \exists^{\geq n} S.C_1 \\ C_d & ::= & C_b \mid f < g \mid f = k \\ C_b & ::= & L \mid \top \\ L & ::= & A \mid \neg A \end{array}$$

where k is a finite string. To avoid undecidability [5]*, a complex role S may occur only in concept descriptions of the form $\exists^{\leq 0} S.C_1$ or of the form $\exists^{\geq 1} S.C_1$.*

An interpretation $\mathcal{I}$ is a pair $\mathcal{I} = (\triangle^{\mathcal{I}} \uplus \mathbf{D}^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$, where $\triangle^{\mathcal{I}}$ is a non-empty set, $\mathbf{D}^{\mathcal{I}}$ a disjoint concrete domain of finite strings, and $(\cdot)^{\mathcal{I}}$ is a function mapping each feature f to a total function $(f)^{\mathcal{I}} : \triangle \to \mathbf{D}$, the "$=$" symbol to the equality relation over $\mathbf{D}$, the "$<$" symbol to the binary relation for an alphabetic ordering of $\mathbf{D}$, a finite string k to itself, NC *to subsets of $\triangle^{\mathcal{I}}$,* NR *to subsets of $\triangle^{\mathcal{I}} \times \triangle^{\mathcal{I}}$, and* NI *to singleton subsets of $\triangle^{\mathcal{I}}$, with the interpretation of inverse roles being $(R^-)^{\mathcal{I}} = \{(o_2, o_1) \mid (o_1, o_2) \in R^{\mathcal{I}}\}$. The interpretation is extended to compound concepts in the standard way.*

A TBox *$\mathcal{T}$ is a finite set of constraints $\mathcal{C}$ of the form $C_1 \sqsubseteq C_2$, $S_1 \sqsubseteq S_2$ or $\mathtt{Trans}(S)$. An* ABox *$\mathcal{A}$ is a finite set of assertions of the form $a : A$, $a : (f \ \mathsf{op}\ k)$ and $S(a, b)$. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{SHOIQ}(\mathbf{D})$* knowledge base *(KB). An interpretation $\mathcal{I}$ is a* model *of $\mathcal{K}$, written $\mathcal{I} \models \mathcal{K}$, iff $(C_1)^{\mathcal{I}} \subseteq (C_2)^{\mathcal{I}}$ holds for each $C_1 \sqsubseteq C_2 \in \mathcal{T}$, $(S_1)^{\mathcal{I}} \subseteq (S_2)^{\mathcal{I}}$ holds for each $S_1 \sqsubseteq S_2 \in \mathcal{T}$, $\{(o_1, o_2), (o_2, o_3)\} \subseteq (S)^{\mathcal{I}}$ implying $(o_1, o_3) \in (S)^{\mathcal{I}}$ holds for $\mathtt{Trans}(S) \in \mathcal{T}$, $(a)^{\mathcal{I}} \in (A)^{\mathcal{I}}$ for $a : A \in \mathcal{A}$, $((a)^{\mathcal{I}}, (b)^{\mathcal{I}}) \in (S)^{\mathcal{I}}$ for $S(a, b) \in \mathcal{A}$, $(f)^{\mathcal{I}}((a)^{\mathcal{I}})\ \mathsf{op}\ k$ for $a : (f\ \mathsf{op}\ k) \in \mathcal{A}$. A concept C is* satisfiable *with respect to a knowledge base $\mathcal{K}$ iff there is an $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{K}$ and such that $(C)^{\mathcal{I}} \neq \emptyset$.*

By a slight abuse of grammar in the following, we allow simpler shorthand for more general concrete domain concepts C_d of the form $(t_1 \texttt{ op } t_2)$, where t_1 and t_2 refer to either a concrete feature or a finite string, and $\texttt{op} \in \{<, \leq, >, \geq, =\}$. For example, $f < k$ would be shorthand for $(f < g) \sqcap (g = k)$, where g is a fresh concrete feature. Also, we write $\forall S.C$ (resp. $\exists S.C$) as shorthand for the concept $\exists^{\leq 0} S.\neg C$ (resp. $\exists^{\geq 1} S.C$).

What we have called *typing constraints* in our introductory comments have the general form

$$C_b \sqsubseteq \exists^{\leq n} S.C_b.$$

As also discussed in our introductory comments, our initial mapping of a given $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ requires that $\mathcal{K}$ adheres to a *normalized* form in which such constraints are always explicit.

Definition 2 (Normalized $\mathcal{SHIQ}(\mathbf{D})$ Terminologies). *A $\mathcal{SHIQ}(\mathbf{D})$ constraint $\mathcal{C}$ is* normalized *if it has one of the forms $C_b \sqsubseteq \exists^{\leq n} S.C_b$, $C_L \sqsubseteq C_R$, $S_1 \sqsubseteq S_2$, or $\texttt{Trans}(S)$. where C_L and C_R are defined by the following grammar.*

$$\begin{array}{rcl} C_L & ::= & C_d \;\mid\; C_L \sqcap C_L \;\mid\; C_L \sqcup C_L \;\mid\; \exists^{\leq n} S.C_L \\ C_R & ::= & C_d \;\mid\; C_R \sqcap C_R \;\mid\; C_R \sqcup C_R \;\mid\; \exists^{\geq n} S.C_R \end{array}$$

A $\mathcal{SHIQ}(\mathbf{D})$ terminology $\mathcal{T}$ is normalized *if each constraint $\mathcal{C}$ occurring in $\mathcal{T}$ is normalized.*

It is a straightforward process to obtain an equisatisfiable normalized terminology from an arbitrary $\mathcal{SHIQ}(\mathbf{D})$ terminology $\mathcal{T}$. In particular, we write $\mathcal{T}^{\text{norm}}$ to denote such a terminology, $\bigcup_{\mathcal{C} \in \mathcal{T}} \mathcal{C}^{\text{norm}}$, where $\mathcal{C}^{\text{norm}}$ is obtained by an exhaustive top-to-bottom application of the following rules, where $\text{NNF}(C)$ denotes concept C in *negation normal form* and also that A' is always a fresh atomic concept.

$$\begin{array}{rcl} (C_b \sqsubseteq \exists^{\leq n} S.C_b)^{\text{norm}} & = & \{C_b \sqsubseteq \exists^{\leq n} S.C_b\} \\ (C_L \sqsubseteq C_R)^{\text{norm}} & = & \{C_L \sqsubseteq C_R\} \\ (S_1 \sqsubseteq S_2)^{\text{norm}} & = & \{S_1 \sqsubseteq S_2\} \\ (\texttt{Trans}(S))^{\text{norm}} & = & \{\texttt{Trans}(S)\} \\ (C_b \sqsubseteq C_1 \sqcap C_2)^{\text{norm}} & = & (C_b \sqsubseteq A' \sqcap C_1)^{\text{norm}} \cup (A' \sqsubseteq C_2)^{\text{norm}} \\ (C_b \sqsubseteq C_1 \sqcup C_2)^{\text{norm}} & = & (C_b \sqsubseteq A' \sqcup C_1)^{\text{norm}} \cup (A' \sqsubseteq C_2)^{\text{norm}} \\ (C_b \sqsubseteq \exists^{\leq n} S.C)^{\text{norm}} & = & \{C_b \sqsubseteq \exists^{\leq n} S.A'\} \cup (C \sqsubseteq A')^{\text{norm}} \\ (C_b \sqsubseteq \exists^{\geq n} S.C)^{\text{norm}} & = & \{C_b \sqsubseteq \exists^{\geq n} S.A'\} \cup (A' \sqsubseteq C)^{\text{norm}} \\ (C_1 \sqsubseteq C_2)^{\text{norm}} & = & (\neg A' \sqsubseteq \text{NNF}(\neg C_1))^{\text{norm}} \cup (A' \sqsubseteq \text{NNF}(C_2))^{\text{norm}} \end{array}$$

Lemma 1. *Let $\mathcal{T}$ be an arbitrary $\mathcal{SHIQ}(\mathbf{D})$ terminology. Then:* (1) *If $\mathcal{I} \models \mathcal{T}^{norm}$ for some $\mathcal{I}$, then $\mathcal{I} \models \mathcal{T}$; and* (2) *If $\mathcal{I} \models \mathcal{T}$ for some $\mathcal{I}$, then there is some interpretation $\mathcal{I}'$ over the same domain such that $\mathcal{I}$ and $\mathcal{I}'$ agree on the interpretation of all symbols in $\mathcal{T}$ and $\mathcal{I}' \models \mathcal{T}^{norm}$.*

Proof. The proof follows by an induction on the normalization rules.

3 ABox Absorption for Local Universal Restrictions

We now show how local universal restrictions of the form of $L_1 \sqsubseteq \forall S.L_2$ can be leveraged to further optimize ABox absorption. In our original ABox absorption framework [13, 14], the following pair of axioms are introduced for each $S(a,b)$ occurring in $\mathcal{A}$, and are then recognized as binary absorptions:

$$\{(\{a\} \sqcap G_S) \sqsubseteq \exists S.(\{b\} \sqcap G), (\{b\} \sqcap G_{S^-}) \sqsubseteq \exists S^-.(\{a\} \sqcap G)\}.$$

Intuitively, a tableau algorithm starts by generating the successor, the nominal on the right-hand side, after lazy unfolding. Other axioms are subsequently unfolded since the newly introduced nominal includes its guard, for example, the guard G for nominal $\{b\}$. We now show how, under some circumstances, one can exploit local universal restrictions to eliminate guards for nominals on the right-hand side of such axioms, possibly replacing the above axioms with the pair

$$\{(\{a\} \sqcap G_S) \sqsubseteq \exists S.\{b\}, (\{b\} \sqcap G_{S^-}) \sqsubseteq \exists S^-.\{a\}\},$$

and thereby avoiding subsequent unfolding. Again, Figure 1 illustrates this process. In particular: $\mathcal{T}^1_{\mathcal{K}}$ attempts such eliminations with simple syntactic checks in the original ABox, and $\mathcal{T}^2_{\mathcal{K}}$ uses $\mathcal{T}^1_{\mathcal{K}}$ for more general subsumption checks to do the same. Also observe that such eliminations are disallowed in the case that S is transitive. Details now follow.

Computing $\mathcal{T}^1_{\mathcal{K}}$: The computation of $\mathcal{T}^1_{\mathcal{K}}$ is based on our earlier method [13] from which we have extracted and refined the definitions of $\mathcal{T}_{\mathcal{T}}$ and $\mathcal{T}_{\mathcal{A}}$ to properly account for role hierarchies, for transitive roles and the above-mentioned syntactic guard elimination. In particular, $\mathcal{T}^1_{\mathcal{K}}$ is now given by

$$\mathcal{T}^{\text{norm}} \;\cup\; \mathcal{T}_{\mathcal{T}} \;\cup\; \mathcal{T}_{\mathcal{A}} \;\cup\; \mathcal{T}_F \;\cup\; \mathcal{T}_{Fd} \;\cup\; \mathcal{T}_B \;\cup\; \mathcal{T}_{Bd},$$

where each of the terminologies is defined as follows:

$$\begin{aligned}\mathcal{T}_{\mathcal{T}} = \{&L_1 \sqsubseteq G_S, L_2 \sqsubseteq G_{S^-} \mid L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}}\}\\ \cup\, \{&t_1 \;\texttt{op}\; t_2 \sqsubseteq G_f \mid f \text{ appears in } t_1 \text{ or in } t_2, t_1 \;\texttt{op}\; t_2 \text{ appears in } \mathcal{T}^{\text{norm}}\}\\ \cup\, \{&G_{S_2} \sqsubseteq G_{S_1}, G_{S_2^-} \sqsubseteq G_{S_1^-} \mid S_1 \sqsubseteq S_2\}\\ \cup\, \{&\top \sqsubseteq G_S \sqcap G_{S^-} \mid S \text{ appears in } \mathcal{T}^{\text{norm}} \text{ and } S \text{ is complex}\}\end{aligned}$$

$$\begin{aligned}\mathcal{T}_{\mathcal{A}} = \{&\{a\} \sqcap G \sqsubseteq A \mid a : A \in \mathcal{A}\}\\ \cup\, \{&\{a\} \sqcap G_f \sqsubseteq f \;\texttt{op}\; k \mid a : (f \;\texttt{op}\; k) \in \mathcal{A}\}\\ \cup\, \{&\{a\} \sqcap G \sqsubseteq \exists S.\top, \{b\} \sqcap G \sqsubseteq \exists S^-.\top \mid S(a,b) \in \mathcal{A}\}\end{aligned}$$

$$\begin{aligned}\mathcal{T}_F = \{&\{a\} \sqcap G_S \sqsubseteq \exists S.\{b\} \mid S(a,b) \in \mathcal{A}, \texttt{Trans}(S) \notin \mathcal{T}^{\text{norm}} \text{ and}\\ &\text{for all } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n = 0 \text{ and } \{a : L_1, b : \text{NNF}(\neg L_2)\} \cap \mathcal{A} \neq \emptyset\}\end{aligned}$$

$$\begin{aligned}\mathcal{T}_{Fd} = \{&\{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \mid S(a,b) \in \mathcal{A} \text{ and } (\texttt{Trans}(S) \in \mathcal{T}^{\text{norm}} \text{ or}\\ &\text{exists } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n > 0 \text{ or } \{a : L_1, b : \text{NNF}(\neg L_2)\} \cap \mathcal{A} = \emptyset)\}\end{aligned}$$

$$\begin{aligned}\mathcal{T}_B = \{&\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.\{a\} \mid S(a,b) \in \mathcal{A}, \texttt{Trans}(S) \notin \mathcal{T}^{\text{norm}} \text{ and}\\ &\text{for all } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n = 0 \text{ and } \{a : \text{NNF}(\neg L_1), b : L_2\} \cap \mathcal{A} \neq \emptyset\}\end{aligned}$$

$$\begin{aligned}\mathcal{T}_{Bd} = \{&\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G) \mid S(a,b) \in \mathcal{A} \text{ and } (\texttt{Trans}(S) \in \mathcal{T}^{\text{norm}} \text{ or}\\ &\text{exists } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n > 0 \text{ or } \{a : \text{NNF}(\neg L_1), b : L_2\} \cap \mathcal{A} = \emptyset)\}\end{aligned}$$

Computing $\mathcal{T}^2_{\mathcal{K}}$: Recall that no reasoning is required in computing $\mathcal{T}^1_{\mathcal{K}}$. Instead, syntactic checks are performed for concept assertions of the form of $a : L_1$ or $b : L_2$ over the ABox. If these concept assertions are found, then it is guaranteed that $S(a, b)$, together with the concept assertions, is consistent with any local universal restrictions of the form $L_1 \sqsubseteq \forall S.L_2$. Although such checks are far from complete, $\mathcal{T}^1_{\mathcal{K}}$ can now be used to perform subsumption checks to find additional cases where local universal restrictions are satisfied by role assertions, that is, to compute $\mathcal{T}^2_{\mathcal{K}}$. The subsumption checks require the notion of a *derivation concept* (cf. Theorem 1) given by the following.

Definition 3 (Derivative Concept). *The* derivative concept $\mathfrak{D}_C$ *for a general* $\mathcal{SHIQ}(\mathbf{D})$ *concept* C *is defined as follows:*

$$\mathfrak{D}_C = \begin{cases} \top & \textit{if } C = C_b; \\ \bigsqcap G_{f_i} & \textit{if } C = (t_1 \ \mathsf{op} \ t_2) \textit{ and } f_i \textit{ appears in } t_1 \textit{ or } t_2; \\ \mathfrak{D}_{C_1} \sqcap \mathfrak{D}_{C_2} & \textit{if } C = C_1 \sqcap C_2 \textit{ or } C = C_1 \sqcup C_2; \\ G_S \sqcap \forall S.(\mathfrak{D}_{C_1} \sqcap G) & \textit{if } C = \exists^{\geq n} S.C_1 \textit{ or } C = \exists^{\leq n} S.C_1. \end{cases}$$

$\mathcal{T}^2_{\mathcal{K}}$ is given by $(\mathcal{T}^1_{\mathcal{K}} \backslash \mathcal{T}^{sub}) \cup \mathcal{T}^{add}$, where $\mathcal{T}^{add}$ and $\mathcal{T}^{sub}$ are defined as follows:

$$\begin{aligned} \mathcal{T}^{sub} = \{\mathcal{C} \mid\ & \mathcal{C} \in \mathcal{T}_{Fd},\ \mathcal{C} = \text{“}\{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G)\text{”}, \mathtt{Trans}(S) \notin \mathcal{T}^{\text{norm}} \\ & \text{and for all } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n = 0 \text{ and} \\ & (\mathcal{T}^1_{\mathcal{K}} \models \{a\} \sqcap G \sqsubseteq L_1 \text{ or } \mathcal{T}^1_{\mathcal{K}} \models \{b\} \sqcap G \sqsubseteq \text{NNF}(\neg L_2))\} \\ \cup \{\mathcal{C} \mid\ & \mathcal{C} \in \mathcal{T}_{Bd},\ \mathcal{C} = \text{“}\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G)\text{”}, \mathtt{Trans}(S) \notin \mathcal{T}^{\text{norm}} \\ & \text{and for all } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{\text{norm}} : n = 0 \text{ and} \\ & (\mathcal{T}^1_{\mathcal{K}} \models \{a\} \sqcap G \sqsubseteq \text{NNF}(\neg L_1) \text{ or } \mathcal{T}^1_{\mathcal{K}} \models \{b\} \sqcap G \sqsubseteq L_2)\} \end{aligned}$$

$$\mathcal{T}^{add} = \{\{a\} \sqcap G_S \sqsubseteq \exists S.\{b\} \mid \{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \in \mathcal{T}^{sub}\}$$

Instance checking as subsumption checking: Once $\mathcal{T}^2_{\mathcal{K}}$ is generated, it can be supplied to the absorption procedure (cf. [13]), which will produce the final TBox $\mathcal{T}^3_{\mathcal{K}}$. Also, there is an important special case in which $\mathcal{T}^3_{\mathcal{K}}$ can be incrementally updated to accommodate new concept assertions of the form a : A in which A is not mentioned in $\mathcal{T}^3_{\mathcal{K}}$ (and therefore in the original $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$):

Lemma 2. *Let* $\mathcal{K}' = \mathcal{K} \cup \bigcup_i \{a_i : A_i\}$,*where each* A_i *is an atomic concept not occurring in* $\mathcal{T}^3_{\mathcal{K}}$. *Then,*

$$\mathcal{T}^3_{\mathcal{K}'} = \mathcal{T}^3_{\mathcal{K}} \cup \bigcup_i \{\{a_i\} \sqcap G_{a_i} \sqsubseteq A_i\}.$$

Proof. The proof follows from the computation of $\mathcal{T}^1_{\mathcal{K}}$ and $\mathcal{T}^2_{\mathcal{K}}$ and the absorption procedure in [13].

Our main results now follow in which we show that an instance checking problem over a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ can be mapped to a subsumption checking problem over the $\mathcal{SHOIQ}(\mathbf{D})$ TBox $\mathcal{T}_{\mathcal{K}}^{i}$, for $1 \leq i \leq 3$.

Definition 4. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base and $\mathcal{T}_{\mathcal{K}} = \mathcal{T}_{\mathcal{K}}^{i}$ for any $1 \leq i \leq 3$. Let $a : C$ be an instance checking over $\mathcal{K}$, and $\{a\} \sqcap D \sqsubseteq C$, be a subsumption check over $\mathcal{T}_{\mathcal{K}}$, where $D = G \sqcap \mathfrak{D}_C$. Let $\mathcal{I}_0$ be an interpretation that satisfies $\mathcal{T}_{\mathcal{K}}$ such that $(\{a\})^{\mathcal{I}_0} \subseteq (D)^{\mathcal{I}_0}$ but $(\{a\})^{\mathcal{I}_0} \cap (C)^{\mathcal{I}_0} = \emptyset$; also, let $\mathcal{I}_1$ be an interpretation that satisfies $\mathcal{K}$ in which* all at-least restrictions are fulfilled by ABox individuals and, if necessary, anonymous objects*. Without loss of generality, we assume both $\mathcal{I}_0$ and $\mathcal{I}_1$ are tree-shaped outside of the ABox (converted ABox). Define an interpretation $\mathcal{J}$ as follows: let a_0 be any ABox individual and $\Gamma^{\mathcal{I}_0}$ be the set of objects $o \in \triangle^{\mathcal{I}_0}$ such that either $o \in (\{a_0\})^{\mathcal{I}_0}$ and $(\{a_0\})^{\mathcal{I}_0} \subseteq (G)^{\mathcal{I}_0}$ or o is an anonymous object in $\triangle^{\mathcal{I}_0}$ rooted by such an object. Similarly let $\Gamma^{\mathcal{I}_1}$ be the set of objects $o \in \triangle^{\mathcal{I}_1}$ such that either $o \in (\{a_0\})^{\mathcal{I}_1}$ and $(\{a_0\})^{\mathcal{I}_0} \cap (G)^{\mathcal{I}_0} = \emptyset$ or o is an anonymous object in $\triangle^{\mathcal{I}_1}$ rooted by such an object. We set*

1. *$\triangle^{\mathcal{J}} = \Gamma^{\mathcal{I}_0} \cup \Gamma^{\mathcal{I}_1}$;*
2. *$(a_0)^{\mathcal{J}} \in (\{a_0\})^{\mathcal{I}_0}$ for $(a_0)^{\mathcal{J}} \in \Gamma^{\mathcal{I}_0}$ and $(a_0)^{\mathcal{J}} = (a_0)^{\mathcal{I}_1}$ for $(a_0)^{\mathcal{J}} \in \Gamma^{\mathcal{I}_1}$;*
3. *$o \in A^{\mathcal{J}}$ if $o \in A^{\mathcal{I}_0}$ and $o \in \Gamma^{\mathcal{I}_0}$ or if $o \in A^{\mathcal{I}_1}$ and $o \in \Gamma^{\mathcal{I}_1}$ for an atomic concept A (similarly for concrete domain concepts of the form $(t_1 \mathtt{op}\ t_2)$);*
4. *$(o_1, o_2) \in (S)^{\mathcal{J}}$ if*
   - *(a) $(o_1, o_2) \in S^{\mathcal{I}_0}$ and $o_1, o_2 \in \Gamma^{\mathcal{I}_0}$, or $(o_1, o_2) \in S^{\mathcal{I}_1}$ and $o_1, o_2 \in \Gamma^{\mathcal{I}_1}$; or*
   - *(b) $o_1 \in (\{a_0\})^{\mathcal{I}_0} \cap (G)^{\mathcal{I}_0}$, $o_2 \in (\{b_0\})^{\mathcal{I}_1}$ and $S(a_0, b_0) \in \mathcal{A}$ (or vice versa); or*
   - *(c) $(o_1, o_2) \in (S_1)^{\mathcal{J}}$ and $S_1 \underline{\circledast} S$; or*
   - *(d) $(o_1, o') \in (S)^{\mathcal{J}}$, $(o', o_2) \in (S)^{\mathcal{J}}$ and $\mathtt{Trans}(S) \in \mathcal{T}$.*

Lemma 3. *For $\{o_1, o_2\} \subseteq \triangle^{\mathcal{J}}$, if $(o_1, o_2) \in (S)^{\mathcal{J}}$ and $\mathtt{Trans}(S) \in \mathcal{T}$, then either $\{o_1, o_2\} \subseteq \Gamma^{\mathcal{I}_0}$ or $\{o_1, o_2\} \subseteq \Gamma^{\mathcal{I}_1}$, where $\mathcal{I}_0$, $\Gamma^{\mathcal{I}_0}$, $\mathcal{I}_1$, $\Gamma^{\mathcal{I}_1}$ and $\mathcal{J}$ are given in Definition 4.*

Proof. The proof proceeds by induction on all cases for interpretation of roles (i.e. 4th point) in Definition 4. Case (4a) is trivial; case (4b) is not applicable when $\mathtt{Trans}(S) \in \mathcal{T}$ as otherwise by the definition of $\mathcal{T}_{\mathcal{T}}$ it holds that $o_1 \in (G_S)^{\mathcal{I}_0}$ and thus the contradiction $o_2 \in (G)^{\mathcal{I}_0}$. Case (4c) is trivial by the induction hypothesis if $\mathtt{Trans}(S_1) \in \mathcal{T}$. We show the case $\mathtt{Trans}(S_1) \notin \mathcal{T}$ is not applicable. Suppose $o_1 \in (\{a_0\})^{\mathcal{I}_0} \cap (G)^{\mathcal{I}_0}$ and $o_2 \in (\{b_0\})^{\mathcal{I}_1}$ (or vice versa), then this is only possible through case (4b). While a similar contradiction can be drawn as in case (4b) because of $G_S \sqsubseteq G_{S_1}$, i.e., $o_1 \in (G_{S_1})^{\mathcal{I}_0}$ and thus the contradiction $o_2 \in (G)^{\mathcal{I}_0}$. Case (4d) follows from the induction hypotheses because either $\{o_1, o_2, o'\} \subseteq \Gamma^{\mathcal{I}_0}$ or $\{o_1, o_2, o'\} \subseteq \Gamma^{\mathcal{I}_1}$ hold. □

Theorem 1. *For any consistent $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$, concept C, and individual a:*

$$\mathcal{K} \models a : C \quad \textit{iff} \quad \mathcal{T}_{\mathcal{K}}^{3} \models \{a\} \sqcap G \sqcap \mathfrak{D}_C \sqsubseteq C.$$

Proof (Outline). Since $\mathcal{T}_{\mathcal{K}}^3$ is obtained by an absorption of $\mathcal{T}_{\mathcal{K}}^2$, of which the correctness follows immediately from the proof of the absorption procedure in [13], it suffices to prove the case $\mathcal{T}_{\mathcal{K}}^2$, i.e., $\mathcal{K} \models a : C$ iff $\mathcal{T}_{\mathcal{K}}^2 \models \{a\} \sqcap G \sqcap \mathfrak{D}_C \sqsubseteq C$. Note that the case $\mathcal{T}_{\mathcal{K}}^1$ also follows from the case $\mathcal{T}_{\mathcal{K}}^2$. Consider Definition 4. We claim that $(\{a\})^{\mathcal{J}} \cap (C)^{\mathcal{J}} = \emptyset$ (trivially) and $\mathcal{J} \models \mathcal{K}$. To show $\mathcal{J} \models \mathcal{K}$, note that the edges from case (4a) satisfy all dependencies in $\mathcal{K}$ as the remainder of the interpretation $\mathcal{J}$ is copied from $\mathcal{I}_0$ or $\mathcal{I}_1$. Thus, we only need to consider those S edges of the form covered by case (4b) (and the extended cases (4c) and (4d)): the edges that *cross* between the two interpretations, i.e., when $o_1 \in (\{a_0\})^{\mathcal{I}_0}$, $o_2 \in (\{b_0\})^{\mathcal{I}_1}$ and $S(a_0, b_0) \in \mathcal{A}$. Now consider an inclusion dependency expressing an *at-most* restriction $L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}$. There are two possibilities: in one case, we can conclude $o_1 \notin (L_1)^{\mathcal{I}_0}$ as otherwise $o_1 \in (G_S)^{\mathcal{I}_0}$ by the definition of $\mathcal{T}_{\mathcal{T}}$ and thus $o_2 \in (G)^{\mathcal{I}_0}$ by the rules for construction of $\mathcal{T}_{\mathcal{K}}^2$, which contradicts our assumption that $(\{b_0\})^{\mathcal{I}_0} \cap (G)^{\mathcal{I}_0} = \emptyset$, hence the inclusion dependency is satisfied vacuously; in the other case, we cannot derive a contradiction because G_{b_0} was removed by our optimization shown in Sect. 3, then it must be the case that the axiom $L_1 \sqsubseteq \exists^{\leq 0} S.L_2 \in \mathcal{T}$, i.e., $L_1 \sqsubseteq \forall S.\neg L_2$, has been satisfied by the role assertion $S(a, b)$. Lemma 3 stipulates that in case (4d) either $\{o_1, o_2, o'\} \subseteq \Gamma^{\mathcal{I}_0}$ or $\{o_1, o_2, o'\} \subseteq \Gamma^{\mathcal{I}_1}$ hold; hence any universal restriction of the form $L_1 \sqsubseteq \forall S.L_2$ (recall that concepts of the form $\exists^{\leq n} S.L_2$ are disallowed for complex S) must be satisfied by (o_1, o_2) because it is already satisfied by (o_1, o_2) in $\mathcal{I}_0$ ($\mathcal{I}_1$, respectively). Edges from case (4c) are follows from all of the above. Hence all inclusion dependencies in $\mathcal{K}$ are satisfied by $\mathcal{J}$.

The other direction of the proof follows by observing that if $\mathcal{K} \cup \{a : \neg C\}$ is satisfiable then the satisfying interpretation I can be extended to $(G_{a_0})^I = (G_f)^I = (G_S)^I = \triangle^I$ for all individuals a_0, concrete features f, and roles S, and $(\{a_0\})^I = \{a_0^I\}$. This extended interpretation then satisfies $\mathcal{T}_{\mathcal{K}}^2$ and $(\{a\})^I \subseteq (D)^I \cap (\neg C)^I$. □

On a safe use of nominals: We briefly consider how nominals can participate in instance queries in such a way that answering the queries will not lead to complex reasoning for $\mathcal{O}$. Such uses of nominals are considered safe because they do not require any modification to the underlying tableau procedure implemented for dialects without $\mathcal{O}$. Note that a similar treatment of nominals has been given for $\mathcal{EL}$ dialects [7]. However, Lemma 4 shows that safe nominals can also be mentioned in instance queries for more expressive DL dialects.

Lemma 4. *For any $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$, concept C and individual a,*

$$\mathcal{K} \models a : C \sqcap \bigsqcap_i \exists S_i.\{a_i\} \quad \textit{iff} \quad \mathcal{K} \cup \bigcup_i \{a_i : A_i\} \models a : C \sqcap \bigsqcap_i \exists S_i.A_i, \qquad (2)$$

where each A_i is a fresh atomic concept for each $\{a_i\}$. We call any occurrence of a nominal in the left-hand-side instance query safe*.*

Proof. The proof is again tedious but straightforward.

Recall that Lemma 2 allows $\mathcal{T}_{\mathcal{K}}^3$ to be incrementally augmented if concept assertions of the form a : A need to be added to $\mathcal{K}$ (where A does not occur in $\mathcal{T}_{\mathcal{K}}^3$). Thus, in combination with Lemma 4, safe uses of nominals in instance queries $a : C$ are easily supported by our method: one simply proceeds by temporarily adding binary absorptions of the form $\{a_i\} \sqcap G_{a_i} \sqsubseteq A_i$ to $\mathcal{T}_{\mathcal{K}}^3$, and invoking the corresponding subsumption check for the right-hand-side of (2) on the result.

The utility of this capability for evaluating BGPs over $\mathcal{SHIQ}(\mathbf{D})$ knowledge bases is a simple consequence of the sometimes unavoidable need to do expensive instance checking [8], that is, when precomputed results by reasoning engines are insufficient for checking query results. Query classes for expensive instance checking can now include "links" to individuals in the range of partial bindings for query variables that are captured as safe uses of nominals in instance queries.

4 Experimental Results

The absorption technique for local universal restrictions has been implemented in the CARE Assertion Retrieval Engine (CARE)[1], which has an underlying $\mathcal{SHI}(\mathbf{D})$ DL reasoner. The DL reasoner features a limited number of optimizations, including the ABox absorption technique described in this paper, optimized double blocking [4], and dependency-directed backtracking [3]. Note that the set of finite strings is the only concrete domain supported by the reasoner.

All times are an average of five independent runs on a single core of the 2.6GHz AMD Opteron 6282 SE processor of a Ubuntu 12.04 Linux server, with up to 4GB of memory. One of the two applications used in the experiments relates to digital cameras (DPC1). This $\mathcal{ALCI}$ KB consists of about 35 axioms, 18k individuals, and 25k role assertions. DPC1 has a considerable number of concrete feature concepts (around 70 features) for each model. Seven queries, varying in selectivity and complexity, were posed over DPC1, as listed in Table 1. The second application is based on the LUBM benchmark [1] using one university (LUBM0), which has about 17k individuals and 49k role assertions. Twelve queries out of the LUBM test queries[2] were used (q_2 and q_9 were excluded since they are not expressible as instance queries). Since the experiments focus on instance retrieval, the selection conditions of the twelve queries were reified, e.g., q_4 was rewritten as the instance query $\mathit{Professor} \sqcap \exists \mathit{worksFor}.A'$, for some fresh atomic concept A', and the concept assertion (`http://www.Department0.University0.edu` : A') added to the original ABox (cf. Lemma 4).

The experimental results are shown in Figure 2 in which the execution time of CARE with/without our optimization has been compared. For DPC1, the preprocessing times (for ABox absorption) are about 7 seconds and 17 seconds when the optimization was off and on, respectively. In the latter case, about 16% of the role assertions were optimized for four universal restrictions (three of which are about the same role). Observe that q_1 and q_3 were improved by 45% and 14%, respecptively, while, for other queries, the runtime improvement was

[1] `http://code.google.com/p/care-engine/`

[2] `http://swat.cse.lehigh.edu/projects/lubm/queries-sparql.txt`

q_1	*Digital_SLR_mirrorless*
q_2	*Compact_Camera*
q_3	$Digital\_SLR \sqcap (user\_review = \text{“5.00”})$
q_4	$Digital\_SLR \sqcap (\neg(user\_review = \text{“5.00”}))$
q_5	$\exists hasSale.(\neg(inventory\_status = \text{“outOfStock”}))$
q_6	$\exists hasManu.((manu\_name = \text{“Kodak”}) \sqcup (\exists locatedIn.Europe\_Country)))$
q_7	$(\exists hasInstance^{-}.(Lens\_mount = \text{“Nikon\_F\_mount”})) \sqcap (\exists hasSale.\exists hasSeller.(seller\_name = \text{“Walmart”}))$

Table 1: Sample Queries for DPC1.

under 5%. The limited gains are not surprising in view of the proportion of role assertions optimized for local universal restrictions and of the characteristics of these queries (which were originally designed to deal with concrete features).

For LUBM0, the preprocessing times are 5 and 16 seconds when the optimization was off and on, respectively. With the optimization on, about 23% of the role assertions were optimized for six local universal restrictions. We have witnessed dramatic improvement with optimization on: all queries were improved by over 40%. In particular, q_1 and q_{10} were improved by 90%. Most of the queries use the role (implicitly or explicitly) *takesCourse* that participates in the local universal restriction $\forall takesCourse.Course$. Hence, the improvement is apparent. The experimental evaluation suggests that our optimization is most useful if there are many local universal restrictions, especially when different roles are involved, and that guard elimination strongly correlates with reduced query evaluation time.

We also observed that, for these KBs, syntactic checks in computing $\mathcal{T}^1_\mathcal{K}$ were not efficacious, that is, both $\mathcal{T}_F$ and $\mathcal{T}_B$ were empty. This is why computing $\mathcal{T}^2_\mathcal{K}$ required more time than the case when this optimization was off: a large number of subsumption checks were performed during the computation of $\mathcal{T}^2_\mathcal{K}$. In our summary comments, we outline how intermediate steps can be introduced to obtain $\mathcal{T}^2_\mathcal{K}$ in a way that, we believe, will greatly reduce the number of such subsumption checks at load time.

5 Related Work and Summary

Instance queries are an important reasoning service over DL knowledge bases, and have been the subject of substantial work in the DL community. Although it is always possible to evaluate an instance query $C(x)$ by performing a sequence of instance checks $\mathcal{K} \models a : C$ for each individual a occurring in $\mathcal{K}$, reasoning engines usually try to reduce the number of such checks by using precomputed results or by "bulk processing" of a range of instance checks. An example of the latter is so-called *binary retrieval* [2], which is used to determine non-answers via

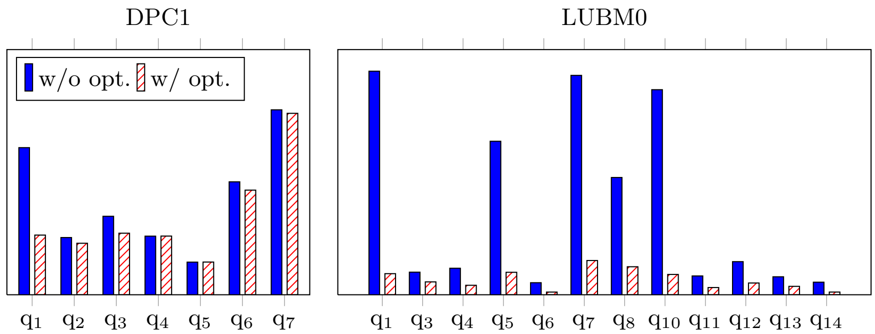

Fig. 2: Reasoning with and without the Optimization.

a single (possibly large) satisfiability check. There have been several approaches to exploiting precomputed results obtained at an earlier time: when a knowledge base is "loaded", or as a consequence of an explicit request [10]. Examples include the *pseudo-model merging* technique [2], presented earlier in [3] as a way to quickly falsify a subsumption check. In particular, a pseudo-model captures the deterministic consequences of concept membership for individuals. Note that model merging techniques are generally sound but incomplete. Methods on how precomputed information can be used to improve the efficiency of evaluating instance queries have also been developed [8, 10]. An approach to instance checking that has much in common with our own method was introduced in [12]. In this case, an ABox is partitioned into small islands such that an instance checking problem is routed to the island "owned" by an individual. Finally, although binary absorption is sufficient to ensure any occurrences of our guard concepts are absorbed, more powerful absorption algorithms could also be used [9].

We have shown in earlier work how instance checking can be improved by introducing guards that in turn prune any unnecessary consideration of individuals and the (possibly large) number of facts about individuals [13, 14]. To recap, the method introduced in this earlier work assumes that knowledge bases are consistent and relies on a refinement of binary absorption to achieve efficiency. Our main result shows how the method can be refined by an additional process that effectively disables the introduction of "trigger" guards in binary absorptions, which in turn reduces the need for lazy unfolding.

There are two labelled arcs in Figure 1 that indicate where additional processing might be useful. In particular, the arc labelled "1" is where a process called *nominal absorption* can be applied that would allow our method to be used for $\mathcal{SHOIQ}(\mathbf{D})$ knowledge bases that admit a limited use of nominals [11]. The arc labelled "2" is where an intermediate process might be included to eliminate guards by, say, reasoning about deterministic consequences using $\mathcal{T}_{\mathcal{K}}^{1}$, which might considerably reduce the number of subsumption checks required in computing $\mathcal{T}_{\mathcal{K}}^{2}$.

Bibliography

[1] Yuanbo Guo, Zhengxiang Pan, and Jeff Heflin. LUBM: A benchmark for OWL knowledge base systems. *Web Semant.*, 3(2-3):158–182, October 2005.

[2] Volker Haarslev and Ralf Möller. On the scalability of description logic instance retrieval. *J. Autom. Reason.*, 41(2):99–142, August 2008.

[3] Ian Horrocks. *Optimising Tableaux Decision Procedures For Description Logics*. PhD thesis, the University of Manchester, 1997.

[4] Ian Horrocks and Ulrike Sattler. Optimised reasoning for $\mathcal{SHIQ}$. In *ECAI'02*, pages 277–281, 2002.

[5] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. In *Proceedings of the 6th International Conference on Logic Programming and Automated Reasoning*, LPAR '99, pages 161–180, London, UK, UK, 1999. Springer-Verlag.

[6] Alexander K. Hudek and Grant E. Weddell. Binary absorption in tableaux-based reasoning for description logics. In *Description Logics'06*, 2006.

[7] Yevgeny Kazakov, Markus Kroetzsch, and Frantisek Simancik. Practical reasoning with nominals in the $\mathcal{EL}$ family of description logics. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, 2012.

[8] Ilianna Kollia and Birte Glimm. Cost based query ordering over OWL ontologies. In *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, pages 231–246, 2012.

[9] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *J. Artif. Intell. Res. (JAIR)*, 36:165–228, 2009.

[10] Jeffrey Pound, David Toman, Grant E. Weddell, and Jiewen Wu. An assertion retrieval algebra for object queries over knowledge bases. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 1051–1056, 2011.

[11] Evren Sirin, Bernardo Cuenca Grau, and Bijan Parsia. From wine to water: Optimizing description logic reasoning for nominals. In *Proceedings, Tenth International Conference on Principles of Knowledge Representation and Reasoning*, pages 90–99, 2006.

[12] Sebastian Wandelt and Ralf Möller. Towards abox modularization of semi-expressive description logics. *Applied Ontology*, 7(2):133–167, 2012.

[13] Jiewen Wu, Alexander K. Hudek, David Toman, and Grant E. Weddell. Absorption for aboxes. In *Proceedings of the 2012 International Workshop on Description Logics, DL-2012*, 2012.

[14] Jiewen Wu, Alexander K. Hudek, David Toman, and Grant E. Weddell. Assertion absorption in object queries over knowledge bases. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012*, 2012.