Absorption for ABoxes and TBoxes with General Value Restrictions

Jiewen Wu, Taras Kinash, David Toman(✉), and Grant Weddell

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

Abstract. We consider the instance checking problem for $\mathcal{SHIQ}(\mathbf{D})$ knowledge bases. In particular, we present a procedure that significantly reduces the number of ABox individuals that need to be examined for a given instance checking problem over a consistent $\mathcal{SHIQ}(\mathbf{D})$ knowledge base that contains arbitrary occurrences of value restrictions. The procedure extends earlier work that assumed value restrictions were predominantly used to establish global domain and range restrictions, and, consequently, in which other applications of value restrictions had a significant risk of requiring an infeasible number of individuals to be examined for a given problem. Finally, experimental results are given that validate the effectiveness of the procedure.

1 Introduction

We consider the instance checking problem for knowledge bases based on the *description logic* (DL) dialect $\mathcal{SHIQ}(\mathbf{D})$. A particular problem is written $\mathcal{K} \models C(a)$, where $\mathcal{K}$ and C are a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base and concept, respectively, and where a is an individual occurring in $\mathcal{K}$. The problem is to determine if membership of a in C must follow from $\mathcal{K}$.

An effective procedure for solving the instance checking problem has become indispensable in SPARQL query evaluation over RDF data sources under the OWL 2 *direct semantics* entailment regime. To see why, consider a simple SPARQL query of the form

```
select ?x
from {?x in C},
```

where C is a $\mathcal{SHIQ}(\mathbf{D})$ concept, i.e., an OWL 2 complex class, and where the data source is a knowledge base $\mathcal{K}$ with a $\mathcal{SHIQ}(\mathbf{D})$ TBox $\mathcal{T}$, i.e., an OWL 2 ontology. An evaluation of the query corresponds to a potentially large number of instance checking problems; the evaluation must produce *all* individuals a occurring as URIs in $\mathcal{K}$ for which $\mathcal{K} \models C(a)$. And since $\mathcal{T}$ can be non-Horn and C not necessarily conjunctive (e.g., mentions the OWL 2 class constructor `ObjectUnionOf`), an appeal to a procedure to decide if $\mathcal{K} \models C(a)$, for *some* individuals a, will almost certainly be necessary.

B. Pfahringer and J. Renz (Eds.): AI 2015, LNAI 9457, pp. 609–622, 2015.
DOI: 10.1007/978-3-319-26350-2_54

In this paper, we present a procedure for the instance checking problem over a *consistent* $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$, that is, the procedure operates with the assumption that the consistency of $\mathcal{K}$ has already been established and at no time requires any additional check to ensure this.[1] Our procedure is a significant enhancement of the procedure reported in [12] that assumes value restrictions are only used to establish global domain and range restrictions, e.g., that correspond to RDFS domain and range restrictions. In this earlier procedure, other applications of value restrictions are likely to lead to what we call the *large ripple problem*, in particular, a need to examine an infeasible number of individuals for a *particular* instance checking problem. Notably, this includes cases in which value restrictions are also used to capture so-called *unary foreign keys* in the relational model, a very common variety of constraint for RDF data sources that derive from legacy relational databases. Such constraints are given in an ontology by OWL 2 *inclusion dependencies* of the form $C_1 \sqsubseteq \forall R.C_2$, where the C_i are concepts and R is a role corresponding to some RDF property. Such "foreign key" constraints ensure that, in the particular case of C_1-objects, R-values must be C_2-objects.[2]

As with the procedure reported in [12], our procedure is designed to use off-the-shelf tableau-based *subsumption checking* technology for DL dialects that include the fragment $\mathcal{SHOIQ}(\mathbf{D})$. A subsumption checking problem is given by a knowledge base $\mathcal{K}$ and an *inclusion dependency* of the form $C_1 \sqsubseteq C_2$ in which C_1 and C_2 are arbitrary concepts. The problem is to determine if $\mathcal{K}$ implies that C_1-objects are also C_2-objects, written $\mathcal{K} \models C_1 \sqsubseteq C_2$.

Our procedure also shares an assumption that such technology incorporates *enhanced binary absorption* optimization, a generalization of binary absorption optimization [7] also presented in [12]. Note that absorption is an indispensable optimization in tableau methods for subsumption checking. Generally, absorption enables *lazy unfolding* of inclusion dependencies that comprise the so-called TBox of $\mathcal{K}$ [6]. Enhanced binary absorption generalizes absorption by also allowing absorbed inclusion dependencies to have the form $(A \sqcap B) \sqsubseteq C$, where $A \sqcap B$ is a *conjunction* of an atomic concept A and *either an atomic concept or a nominal* B. As with any absorbed inclusion dependencies, such constraints are "unfolded" for an individual in a tableau chase only when it becomes known that it must be both an A-object and a B-object. This considerably enhances the overall performance of lazy unfolding and of reasoning.

Both earlier work and our procedure operate by reducing an instance checking problem over a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ to a subsumption checking problem over a $\mathcal{SHOIQ}(\mathbf{D})$ TBox $\mathcal{T}_\mathcal{K}$ derived from $\mathcal{K}$. Roughly, this is achieved

[1] This can be an important feature in cases for which a consistency check when "loading $\mathcal{K}$" would constitute a significant overhead due to the size and complexity of an included ontology.

[2] Note that unary foreign keys occur in the LUBM benchmark [2] that we appeal to in our experimental evaluation. And indeed, as confirmed by our experimental results, such keys lead to large ripple problems.

by rewriting the TBox of $\mathcal{K}$ and by introducing additional inclusion dependencies with nominals to encode the ABox assertions in $\mathcal{K}$. Note that, by relying on enhanced binary absorption to ensure the additional dependencies encoding the ABox are absorbed, the overhead of reasoning about $\mathcal{SHOIQ}(\mathbf{D})$ ontologies with *arbitrary* use of nominals that would otherwise be required is entirely avoided.

Our procedure is comprised of four steps that are largely inherited from [12]. An input $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ is first separated into its component TBox $\mathcal{T}$ and ABox $\mathcal{A}$. A *normalized* TBox $\mathcal{T}^{\mathrm{NR}}$ is then obtained from $\mathcal{T}$ in order to *isolate* inclusion dependencies that mention value restrictions. Two subsequent $\mathcal{SHOIQ}(\mathbf{D})$ TBoxes, $\mathcal{T}^1_{\mathcal{K}}$ and $\mathcal{T}^2_{\mathcal{K}}$ are then derived from $\mathcal{T}^{\mathrm{NR}}$ and $\mathcal{A}$. In particular, the above-mentioned subsumption checking technology uses TBox $\mathcal{T}^1_{\mathcal{K}}$ to compute $\mathcal{T}^2_{\mathcal{K}}$. The same technology is then also used to solve subsequent instance checking problems over $\mathcal{K}$ by encoding them as subsumption checking problems over $\mathcal{T}^2_{\mathcal{K}}$. Our main result relates to this encoding. In particular, an instance check of the form $\mathcal{K} \models C(a)$ is mapped to a subsumption check of the form

$$\mathcal{T}^2_{\mathcal{K}} \models \{a\} \sqcap \mathfrak{D}_C \sqsubseteq C, \tag{1}$$

where $\mathfrak{D}_C$ is a concept that initializes an appropriate "firing" of binary absorptions in $\mathcal{T}^2_{\mathcal{K}}$. This result refines the so-called *donut theorem* in [12]. The theorem is so-named since its proof relies on a way of combining the results of tableau expansion in reasoning about (1), called a *tinbit*[3], with a particular partial interpretation that must exist for any consistent $\mathcal{K}$, the *donut*, to obtain an interpretation of $\mathcal{K}$ for which a is not a C whenever $\mathcal{K} \models C(a)$ does not hold, that is, which interprets concept $\{a\} \sqcap \mathfrak{D}_C \sqcap \neg C$ as non-empty.

The remainder of the paper is organized as follows. Our primary contributions are in Sect. 3 in which we define the computation of $\mathcal{T}^1_{\mathcal{K}}$ and $\mathcal{T}^2_{\mathcal{K}}$ (see [12] for the computation of $\mathcal{T}^{\mathrm{NR}}$) and in which we present our main result. The results of an experimental evaluation of our procedure are given in Sect. 4. The results confirm that our procedure is effective in addressing performance issues that surface with TBoxes that rely on the general use of value restrictions. A review of related work and a summary discussion then follow in Sect. 5. Part of this discussion outlines refinements of our procedure that can improve its performance or increase the scope of $\mathcal{SHIQ}(\mathbf{D})$ knowledge bases for which the method can be used.

Finally, note that we do *not* address all problems relating to *instance queries* in this paper, an example of which is given as a SPARQL query above. An instance query is given by a knowledge base $\mathcal{K}$ and a concept C, and asks for the set of *all* individuals a mentioned in $\mathcal{K}$ for which $\mathcal{K} \models a : C$. For example, there is a growing body of work on exploiting precomputed or cached results by DL reasoning engines to address this problem [8,9], a topic that we briefly return to in our summary comments.[4] In this paper, we focus on the problem

[3] Also inspired by the Canadian word for a "donut hole" pastry called a *Timbit*.

[4] At a minimum, an interface to a cache of all individual names occurring in $\mathcal{K}$ would be required, that is, a cache of the result of evaluating the instance query given by $\mathcal{K}$ and the "top" concept $\top$.

of instance checking for cases where general reasoning is necessary, in particular where precomputed or cached results are unavailable, indeed, an unavoidable circumstance with (arbitrary) non-atomic concepts and non-Horn ontologies.

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 will help with clarity in 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 $S \in \mathrm{NR} \cup \{R^- \mid R \in \mathrm{NR}\}$ S^{--}, we define $S^- = R$ if $S = R^-$ and $S^- = R^-$ otherwise. A role inclusion has the form $S_1 \sqsubseteq S_2$. Let $\circledast$ 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 $R \circledast S$ and $S \circledast R$. A role S is called* complex *if $\mathtt{Trans}(S')$ for some $S' \circledast S$.*

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

$$
\begin{array}{lcl}
C & ::= & C_d \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \{a\} \mid \neg\{a\} \mid \exists^{\leq n} S.C_1 \mid \exists^{\geq n} S.C_1 \\
C_d & ::= & C_b \mid f < g \mid f = k \\
C_b & ::= & L \mid \top \\
L & ::= & A \mid \neg A
\end{array}
$$

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, $\mathbf{D}^{\mathcal{I}}$ is a disjoint concrete domain of finite strings, and $(\cdot)^{\mathcal{I}}$ is a function that maps 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 $A(a)$, $(f = k)(a)$ and $S(a, b)$. Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be an $\mathcal{SHOIQ}(\mathbf{D})$* knowledge base*. 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* $S(a,b) \in \mathcal{A}$, $(f)^{\mathcal{I}}((a)^{\mathcal{I}}) < (g)^{\mathcal{I}}((a)^{\mathcal{I}})$ *for* $(f < g)(a) \in \mathcal{A}$, *and* $(f)^{\mathcal{I}}((a)^{\mathcal{I}}) = k$ *for* $(f = k)(a) \in \mathcal{A}$. *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* $(C)^{\mathcal{I}} \neq \emptyset$. □

Note that assertions of the form $S(a,b)$ and $(f = k)(a)$ correspond, respectively, to so-called *object property assertions* and *data property assertions* in RDF. By a slight abuse of grammar in the following, we allow simpler shorthand for more general concrete domain concepts C_d of the form $(t_1 \ \mathtt{op} \ t_2)$, where t_1 and t_2 refer to either a concrete feature or a finite string, and $\mathtt{op} \in \{<, =\}$. Also note that we write $\forall S.C$ and $\exists S.C$ as shorthand for the respective concepts $\exists^{\leq 0} S.\neg C$ and $\exists^{\geq 1} S.C$.

Regarding value restrictions, we focus on their use in inclusion dependencies of the form

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

In particular, we shall be concerned with cases where $n = 0$ and where each C_b is a literal L, and refer to such dependencies as *local universal restrictions*. Indeed, the cases where C_b is not a literal correspond to domain and range restrictions of the form $\top \sqsubseteq \forall S.C_b$, a variety of dependency that has already been addressed in earlier work [12].

Recall that the first step of our procedure for instance checking separates a given $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ into its component TBox $\mathcal{T}$ and ABox $\mathcal{A}$, and that the second step requires $\mathcal{T}$ to be mapped to a *normal form* in which local value restrictions have been extracted. The normal form is defined as follows:

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* $\mathtt{Trans}(S)$, *where* C_L *and* C_R *are defined by the following grammar:*

$$C_L ::= C_d \mid C_L \sqcap C_L \mid C_L \sqcup C_L \mid \exists^{\leq n} S.C_L$$
$$C_R ::= C_d \mid C_R \sqcap C_R \mid C_R \sqcup C_R \mid \exists^{\geq n} S.C_R$$

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 any $\mathcal{SHIQ}(\mathbf{D})$ terminology $\mathcal{T}$ [12].

3 Absorption for ABoxes and TBoxes with Local Universal Restrictions

We now show how local universal restrictions of the form $L_1 \sqsubseteq \forall S.L_2$ occurring in a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ with ABox $\mathcal{A}$ and normalized TBox $\mathcal{T}^{\mathrm{NR}}$ can be leveraged to further optimize ABox absorption.

As prescribed in [12], the following pair of axioms are introduced for each role assertion $S(a, b)$ occurring in $\mathcal{A}$, which then qualify for binary absorption:

$$\{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G), \text{ and} \tag{2}$$

$$\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G). \tag{3}$$

Note that G_S, G_{S^-} and G are atomic concepts functioning as *guards* that can control lazy unfolding. Also note that both (2) and (3) are needed to account for the bi-directional significance of $S(a, b)$.

A tableau algorithm might "fire" either (2) and (3) during lazy unfolding, and therefore create a new individual, either b or a, and labeled with the guard G. Unconditionally adding this label is problematic since this is likely to cause the firing of other absorptions derived from $\mathcal{A}$, thus leading to the large ripple problem mentioned in our introductory comments in which an infeasible number of individuals are "examined" for a particular instance checking problem. 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 with the respective pair

$$\{a\} \sqcap G_S \sqsubseteq \exists S.\{b\}, \text{ and} \tag{4}$$

$$\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.\{a\}, \tag{5}$$

and thereby avoiding subsequent unfolding that might otherwise ensue if the right-hand-side G guards were not removed.

With the assumption of knowledge base consistency, the conditions that enable guard elimination for a role assertion $S(a, b)$, e.g., enable replacing (2) with (4), can be informally stated as the following pair of conditions:

(C1) S is not used in at-most restrictions other than local universal restrictions, and
(C2) for each local universal restriction $L_1 \sqsubseteq \forall S.L_2$ mentioning S, it must hold that $\mathcal{K} \models L_2(b)$.

Recall that our instance checking procedure has four steps. The first two steps are outlined in the preceding section and result in mapping a given $\mathcal{SHIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ to an ABox $\mathcal{A}$ and a normalized TBox $\mathcal{T}^{\text{NR}}$. The remaining two steps of this procedure require computing $\mathcal{T}^1_{\mathcal{K}}$ and then using off-the-shelf subsumption checking technology (with enhanced binary absorption optimization) to compute $\mathcal{T}^2_{\mathcal{K}}$. Details now follow in which the definition of $\mathcal{T}^1_{\mathcal{K}}$ is derived from the corresponding computation in [12] from which we have extracted and refined the definitions of $\mathcal{T}^{\mathcal{T}}$ and $\mathcal{T}^{\mathcal{A}}$ to properly account for the syntactic guard elimination as outlined above.

Definition 3 (Computing $\mathcal{T}^1_{\mathcal{K}}$). *$\mathcal{T}^1_{\mathcal{K}}$ is given by*

$$\mathcal{T}^{NR} \cup \mathcal{T}^{\mathcal{T}} \cup \mathcal{T}^{\mathcal{A}} \cup \mathcal{T}^{\forall}_{\rightarrow} \cup \mathcal{T}^{\forall}_{\rightarrow d} \cup \mathcal{T}^{\forall}_{\leftarrow} \cup \mathcal{T}^{\forall}_{\leftarrow d},$$

with component terminologies other than $\mathcal{T}^{NR}$ 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}^{NR}\} \\
&\quad \cup \{(t_1 \ \mathit{op}\ t_2) \sqsubseteq G_f \mid f \ \mathit{in}\ t_1 \ \mathit{or\ in}\ t_2, (t_1 \ \mathit{op}\ t_2) \ \mathit{in}\ \mathcal{T}^{NR}\} \\
&\quad \cup \{G_{S_2} \sqsubseteq G_{S_1}, G_{S_2^-} \sqsubseteq G_{S_1^-} \mid S_1 \sqsubseteq S_2\} \\
&\quad \cup \{\top \sqsubseteq G_S \sqcap G_{S^-} \mid S \ \mathit{occurs\ in}\ \mathcal{T}^{NR} \ \mathit{and}\ S \ \mathit{is\ complex}\} \\
\mathcal{T}^{\mathcal{A}} &= \{\{a\} \sqcap G \sqsubseteq A \mid A(a) \in \mathcal{A}\} \\
&\quad \cup \{\{a\} \sqcap G_f \sqsubseteq (f \ \mathit{op}\ k) \mid (f \ \mathit{op}\ k)(a) \in \mathcal{A}\} \\
&\quad \cup \{\{a\} \sqcap G \sqsubseteq \exists S.\top, \{b\} \sqcap G \sqsubseteq \exists S^-.\top \mid S(a,b) \in \mathcal{A}\} \\
\mathcal{T}^{\forall}_{\rightarrow} &= \{\{a\} \sqcap G_S \sqsubseteq \exists S.\{b\} \mid S(a,b) \in \mathcal{A}, \mathit{Trans}(S) \notin \mathcal{T}, \\
&\qquad \mathit{and\ for\ each}\ L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR}, \\
&\qquad\quad n = 0 \ \mathit{and}\ \{L_1(a), \mathrm{NNF}(\neg L_2)(b)\} \cap \mathcal{A} \neq \emptyset\} \\
\mathcal{T}^{\forall}_{\rightarrow d} &= \{\{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \mid S(a,b) \in \mathcal{A}, \\
&\qquad \mathit{and\ for\ some}\ L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR}, \\
&\qquad\quad n > 0 \ \mathit{or}\ \{L_1(a), \mathit{NNF}(\neg L_2)(b)\} \cap \mathcal{A} = \emptyset\} \\
\mathcal{T}^{\forall}_{\leftarrow} &= \{\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.\{a\} \mid S(a,b) \in \mathcal{A}, \mathit{Trans}(S) \notin \mathcal{T}, \\
&\qquad \mathit{and\ for\ each}\ L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR}, \\
&\qquad\quad n = 0 \ \mathit{and}\ \{\mathit{NNF}(\neg L_1)(a), L_2(b)\} \cap \mathcal{A} \neq \emptyset\} \\
\mathcal{T}^{\forall}_{\leftarrow d} &= \{\{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G) \mid S(a,b) \in \mathcal{A}, \\
&\qquad \mathit{and\ for\ some}\ L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR}, \\
&\qquad\quad n > 0 \ \mathit{or}\ \{\mathit{NNF}(\neg L_1)(a), L_2(b)\} \cap \mathcal{A} = \emptyset\}
\end{aligned}
$$

□

Intuitively, for any role assertion $S(a,b)$, the corresponding axioms in the form of (2) are checked *syntactically* for guard elimination, which are placed into $\mathcal{T}^{\forall}_{\rightarrow}$ if guards cannot be eliminated and into $\mathcal{T}^{\forall}_{\rightarrow d}$ otherwise. Similarly, "backward" axioms in the form of (3) are either in $\mathcal{T}^{\forall}_{\leftarrow}$ or $\mathcal{T}^{\forall}_{\leftarrow d}$, depending on if the right-hand side guards can be eliminated.

The syntactic check conditions presented above for ensuring (C1) and (C2) are subtle. For example, $\mathcal{T}^{\forall}_{\rightarrow}$ requires the condition $\{L_1(a), \mathrm{NNF}(\neg L_2)(b)\} \cap \mathcal{A} \neq \emptyset$, where the presence of $L_1(a)$ in $\mathcal{A}$ guarantees $\mathcal{K} \models \neg L_2(b)$ because of the axiom $L_1 \sqsubseteq \exists^{\leq 0} S.L_2$ and the consistency assumption of $\mathcal{K}$. Hence, adding $L_1(a)$ to the conditions of $\mathcal{T}^{\forall}_{\rightarrow}$ renders syntactic checking over $\mathcal{A}$ more efficient.

Observe that computing $\mathcal{T}^1_{\mathcal{K}}$ entails simple syntactic lookups in $\mathcal{A}$ for concept assertions of the form $L_1(a)$ or $L_2(b)$. If the lookups succeed, $S(a,b)$ is consistent with all local universal restrictions of the form $L_1 \sqsubseteq \forall S.L_2$. Although such lookups are not complete, an absorbed $\mathcal{T}^1_{\mathcal{K}}$ can be useful in conducting further subsumption checks to find additional cases where role assertions are consistent with the local universal restrictions, indeed, to decide conditions (C1) and (C2) above.

Definition 4 (Computing $\mathcal{T}^2_{\mathcal{K}}$). *$\mathcal{T}^2_{\mathcal{K}}$ is given by $(\mathcal{T}^1_{\mathcal{K}} \backslash \mathcal{T}^{sub}) \cup \mathcal{T}^{add}$, where $\mathcal{T}^{sub}$ and $\mathcal{T}^{add}$ are defined as follows:*

$$
\begin{aligned}
\mathcal{T}^{sub} = \{ & \{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \mid \mathit{Trans}(S) \notin \mathcal{T}^{NR}, \\
& \{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \in \mathcal{T}^{\forall}_{\rightarrow d}, \ \textit{and} \\
& \quad \textit{for each } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR} : n = 0 \ \textit{and} \\
& \quad\quad (\mathcal{T}^1_{\mathcal{K}} \models \{a\} \sqcap G \sqsubseteq L_1 \ \textit{or} \ \mathcal{T}^1_{\mathcal{K}} \models \{b\} \sqcap G \sqsubseteq \neg L_2) \ \} \\
& \cup \{ \{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G) \mid \mathit{Trans}(S^-) \notin \mathcal{T}^{NR}, \\
& \{b\} \sqcap G_{S^-} \sqsubseteq \exists S^-.(\{a\} \sqcap G) \in \mathcal{T}^{\forall}_{\leftarrow d}, \ \textit{and} \\
& \quad \textit{for each } L_1 \sqsubseteq \exists^{\leq n} S.L_2 \in \mathcal{T}^{NR} : n = 0 \ \textit{and} \\
& \quad\quad (\mathcal{T}^1_{\mathcal{K}} \models \{a\} \sqcap G \sqcap \sqsubseteq \neg L_1 \ \textit{or} \ \mathcal{T}^1_{\mathcal{K}} \models \{b\} \sqcap G \sqsubseteq L_2) \ \} \\
\mathcal{T}^{add} = \{ & \{a\} \sqcap G_S \sqsubseteq \exists S.\{b\} \mid \{a\} \sqcap G_S \sqsubseteq \exists S.(\{b\} \sqcap G) \in \mathcal{T}^{sub}\}
\end{aligned}
$$

□

To illustrate the overall process of computing $\mathcal{T}^2_{\mathcal{K}}$, consider a knowledge base U with component TBox Ut and ABox Ua respectively defined as follows:

$$\{\mathit{Dept} \sqsubseteq \forall \mathit{headOf}^-.\mathit{Prof}, \mathit{Chair} \sqsubseteq \mathit{Prof}\}, \text{ and } \{p : \mathit{Chair}, \mathit{headOf}^-(d, p)\}.$$

From Definition 2, $(\mathrm{Ut})^{\mathrm{NR}} = \mathrm{Ut}$, and Sect. 3 then defines the following.

$$
\begin{aligned}
\mathcal{T}^{\mathrm{Ut}} &= \{\mathit{Dept} \sqsubseteq G_{\mathit{headOf}^-}, \neg \mathit{Prof} \sqsubseteq G_{\mathit{headOf}}\} \\
\mathcal{T}^{\mathrm{Ua}} &= \{\{p\} \sqcap G \sqsubseteq \mathit{Chair}, \{d\} \sqcap G \sqsubseteq \exists \mathit{headOf}^-.\top, \ \{p\} \sqcap G \sqsubseteq \exists \mathit{headOf}.\top\} \\
\mathcal{T}^{\forall}_{\rightarrow} &= \mathcal{T}^{\forall}_{\leftarrow} = \emptyset \\
\mathcal{T}^{\forall}_{\rightarrow d} &= \{\{d\} \sqcap G_{\mathit{headOf}^-} \sqsubseteq \exists \mathit{headOf}^-.(\{p\} \sqcap G)\} \\
\mathcal{T}^{\forall}_{\leftarrow d} &= \{\{p\} \sqcap G_{\mathit{headOf}} \sqsubseteq \exists \mathit{headOf}.(\{d\} \sqcap G)\} \\
\mathcal{T}^1_{\mathrm{U}} &= \mathrm{Ut} \cup \mathcal{T}^{\mathrm{Ut}} \cup \mathcal{T}^{\mathrm{Ua}} \cup \mathcal{T}^{\forall}_{\rightarrow d} \cup \mathcal{T}^{\forall}_{\leftarrow d}
\end{aligned}
$$

Computing $\mathcal{T}^2_{\mathrm{U}}$ requires a subsumption check over $\mathcal{T}^1_{\mathrm{U}}$ to determine if $\mathit{Dept} \sqsubseteq \forall \mathit{headOf}^-.\mathit{Prof}$ is consistent with the role assertion $\mathit{headOf}^-(d, p)$. In particular, since

$$\mathcal{T}^1_{\mathrm{U}} \models \{p\} \sqcap G \sqsubseteq \mathit{Prof},$$

we obtain the following:

$$
\begin{aligned}
\mathcal{T}^{sub} &= \{\{d\} \sqcap G_{\mathit{headOf}^-} \sqsubseteq \exists \mathit{headOf}^-.(\{p\} \sqcap G)\} \\
\mathcal{T}^{add} &= \{\{d\} \sqcap G_{\mathit{headOf}^-} \sqsubseteq \exists \mathit{headOf}^-.\{p\}\} \\
\mathcal{T}^2_{\mathrm{U}} &= (\mathcal{T}^2_{\mathrm{U}} \backslash \mathcal{T}^{sub}) \ \cup \ \mathcal{T}^{add} = \mathrm{Ut} \cup \mathcal{T}^{\mathrm{Ut}} \cup \mathcal{T}^{\mathrm{Ua}} \cup \mathcal{T}^{\forall}_{\leftarrow d} \cup \mathcal{T}^{add}
\end{aligned}
$$

Thus, the final terminology $\mathcal{T}^2_{\mathrm{U}}$ consists of the following axioms, all of which will be absorbed with extended binary absorption:

(unary absorptions)	(binary absorptions)
$Dept \sqsubseteq \forall headOf^{-}.Prof$	$\{p\} \sqcap G \sqsubseteq Chair$
$Chair \sqsubseteq Prof$	$\{d\} \sqcap G \sqsubseteq \exists headOf^{-}.\top$
$Dept \sqsubseteq G_{headOf^{-}}$	$\{p\} \sqcap G \sqsubseteq \exists headOf.\top$
$\neg Prof \sqsubseteq G_{headOf}$ †	$\{p\} \sqcap G_{headOf} \sqsubseteq \exists headOf.(\{d\} \sqcap G)$
	$\{d\} \sqcap G_{headOf^{-}} \sqsubseteq \exists headOf^{-}.\{p\}$

Our main result now follows in which we show that an instance checking problem $\mathcal{K} \models C(a)$, where $\mathcal{K}$ is a $\mathcal{SHIQ}(\mathbf{D})$ knowledge base, reduces to a subsumption checking problem over either the $\mathcal{SHOIQ}(\mathbf{D})$ TBox $\mathcal{T}^1_{\mathcal{K}}$ or $\mathcal{T}^2_{\mathcal{K}}$. In preparation, we define a *derivation concept* for C:

Definition 5 (Derivative Concept). *The* derivative concept $\mathfrak{D}_C$ *for a* $\mathcal{SHIQ}(\mathbf{D})$ *concept* C *is defined as follows:*

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

Definition 6 and Lemma 1 that follow are reproduced from [12].

Definition 6 *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be a* $\mathcal{SHIQ}(\mathbf{D})$ *knowledge base and* $\mathcal{T}_{\mathcal{K}} = \mathcal{T}^i_{\mathcal{K}}$ *for any* $i \in \{2, 3\}$. *Let* $a : C$ *be an instance check 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 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
4. $(f)^{\mathcal{J}}(o)$ op $(g)^{\mathcal{J}}(o)$ *if* $(f)^{\mathcal{I}_0}(o)$ op $(g)^{\mathcal{I}_0}(o)$ *and* $o \in \Gamma^{\mathcal{I}_0}$ *or if* $(f)^{\mathcal{I}_1}(o)$ op $(g)^{\mathcal{I}_1}(o)$ *and* $o \in \Gamma^{\mathcal{I}_1}$ *(and similarly for* f op k*);*
5. $(o_1, o_2) \in (S)^{\mathcal{J}}$ *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* $S_1 \sqsubseteq^{*} S$; *or*

(d) $(o_1, o') \in (S)^{\mathcal{J}}$, $(o', o_2) \in (S)^{\mathcal{J}}$ *and* $\mathtt{Trans}(S)$. □

Lemma 1. *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 6.* □

Theorem 1. *For any consistent* $\mathcal{SHIQ}(\mathbf{D})$ *knowledge base* $\mathcal{K}$, *concept* C, *individual* a, *and* $1 \leq i \leq 2$*:*

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

Proof (sketch). Case $i = 1$ is implicitly included in case $i = 2$, so, it is sufficient to prove the latter. For $i = 2$, consider Definition 6. We claim that $(\{a\})^{\mathcal{J}} \cap (C)^{\mathcal{J}} = \emptyset$ (trivially) and $J \models \mathcal{K}$. To show $J \models \mathcal{K}$, 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 $\mathcal{T}^2_{\mathcal{K}}$, which contradicts our assumption that $(\{b_0\})^{\mathcal{I}_0} \cap (G)^{\mathcal{I}_0} = \emptyset$, hence the inclusion dependency is consistent with the role assertion vacuously; in the other case, we cannot derive a contradiction because G was removed by our optimization shown in Sect. 3, then it must be the case that the role assertion $S(a, b)$ is consistent with the axiom $L_1 \sqsubseteq \exists^{\leq 0} S.L_2 \in \mathcal{T}$, i.e., $L_1 \sqsubseteq \forall S.\neg L_2$. Lemma 1 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 trivial extension to all of the above. Hence all axioms in $\mathcal{K}$ are satisfied by $\mathcal{J}$.

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

Returning to our example above to illustrate, consider the instance checking problem

$$\mathrm{U} \models p : \mathit{Prof}.$$

By Theorem 1, this can be decided by the subsumption checking problem

$$\mathcal{T}^2_{\mathrm{U}} \models \{p\} \sqcap G \sqcap \mathfrak{D}_{\mathit{Prof}} \sqsubseteq \mathit{Prof}$$

with the assumption that U is consistent.

4 Experimental Evaluation

Earlier work has already established the efficacy of ABox absorption for instance checking problems, particularly so in cases where there are a large number of concept assertions of the form $(f = k)(a)$[5]. Indeed, [12] presents an extensive comparison of the utility of ABox absorption implemented in a tableau based DL reasoner called *CARE Assertion Retrieval Engine* (https://code.google.com/p/care-engine/) with other state-of-the-art reasoners. CARE has an underlying $\mathcal{SHI}(\mathbf{D})$ DL reasoner that implements ABox absorption, optimized double blocking [5], and dependency-directed backtracking [4]. Note that the set of finite strings is the only concrete domain supported by the reasoner.

The comparison was based on a benchmark consisting of a selection of *instance queries* over a digital camera knowledge base. Indeed, ABox absorption enabled CARE to outperform other reasoners on a number of these queries, despite provisioning ensuring that CARE itself had no access to any precomputed or cached results.

Our objectives with the experiments presented below are focused on evaluating the performance of the our new procedure with the performance of the earlier procedure in [12] used as the baseline. Thus, the empirical evaluation also uses the above-mentioned CARE system. Note that all reported times are the average of five independent runs on the 2.6 GHz AMD Opteron 6282 SE processor of a Ubuntu 12.04 Linux server, with up to 4 GB of heap.

We conducted several experiments on the LUBM benchmark [2] using one university (LUBM0) which has about 17 k individuals and 49 k role assertions.[6] Twelve queries out of the LUBM test queries[7] were used (Q_2 and Q_9 were excluded since they are not expressible as instance queries). Since the experiments focus on instance checking, the selection conditions for each of the twelve queries were reified, e.g., Q_4 was rewritten as the instance query $Professor \sqcap \exists worksFor.A'$, for some fresh atomic concept A', and a new concept assertion, http://www.Department0.University0.edu: A', was added to the original ABox.

The run-time results are listed in Fig. 1 in which the execution time in seconds of CARE with the new method, called OPT (the right striped bars), is compared with the execution time of CARE with the earlier version lacking the optimizations outlined in previous sections, called BASE (the left blue bars). The figure also reports the relative improvement of OPT with respect to BASE as a percentage. For LUBM0, preprocessing costs 5 and 16 s for the BASE and OPT methods, respectively.

[5] Recall that these are called *data property assertions* in RDF.

[6] We chose to report on experiments using the LUBM benchmark for this study because of its wider appeal, e.g., its adoption of a set of predefined queries, and because its TBox includes foreign key constraints that are not global domain and range restrictions, a property missing with the digital camera case studied in [12]. Additional experiments can be found in [11].

[7] http://swat.cse.lehigh.edu/projects/lubm/.

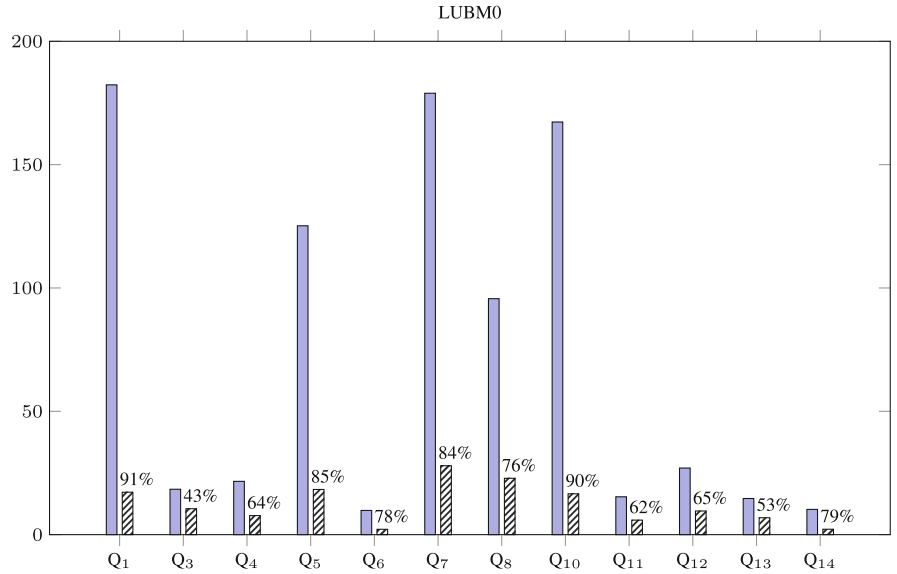

Fig. 1. Performance gains.

With OPT, about 23 % of the role assertions were optimized for LUBM's six typing constraints. Observe that, for LUBM0 instance queries, there is a dramatic improvement ranging from 43 % to 91 % by OPT. The results suggest that LUBM0 is quite sensitive to typing constraints in comparison to knowledge bases with global domain and range restrictions only.

We have also observed that, for these knowledge bases, $\mathcal{T}^1_{\mathcal{K}}$ constructed using *syntactic* lookups in the ABox to approximate concept membership of individuals yields empty $\mathcal{T}^{\forall}_{\rightarrow}$ and $\mathcal{T}^{\forall}_{\leftarrow}$, resulting in inferior performance. Only $\mathcal{T}^2_{\mathcal{K}}$, constructed using *semantic* concept membership tests, yields the desired improvements. Note, however, that $\mathcal{T}^1_{\mathcal{K}}$ is essential *to compute* $\mathcal{T}^2_{\mathcal{K}}$, i.e., allows more efficient class membership checks. Indeed, additional (auxiliary) experiments have shown that sidestepping either of the steps dramatically decreases the overall performance of the system.

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* [3], which is used to determine non-answers via a single (possibly large) satisfiability check.

Another approach to avoid checking individuals sequentially is through summarization and refinement [1], in which query evaluation is performed over a summary of the original ABox that is significantly smaller in size, and iterative refinement based on inconsistency justification is used to purge spurious answers.

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 [9]. Examples include the *pseudo-model merging* technique [3], presented earlier in [4] 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 [8,9]. Observe that the aforementioned optimizations concern how to reduce the number of instance checking tasks, not how to improve the instance checking problem itself.

An approach to instance checking that has much in common with our own method was introduced in [10]. 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. In contrast, our method simply reduces the problem of efficient instance checking to absorption.

Earlier work shows 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 further lazy unfolding.

References

1. Dolby, J., Fokoue, A., Kalyanpur, A., Kershenbaum, A., Schonberg, E., Srinivas, K., Ma, L.: Scalable semantic retrieval through summarization and refinement. In: AAAI 2007, pp. 299–304 (2007)
2. Guo, Y., Pan, Z., Heflin, J.: LUBM: a benchmark for OWL knowledge base systems. Web Semant. **3**(2–3), 158–182 (2005)
3. Haarslev, V., Möller, R.: On the scalability of description logic instance retrieval. J. Autom. Reasoning (JAR) **41**(2), 99–142 (2008)
4. Horrocks, I.: Optimising tableaux decision procedures for description logics. Ph.D. thesis, the University of Manchester (1997)
5. Horrocks, I., Sattler, U.: Optimised reasoning for $\mathcal{SHIQ}$. In: ECAI 2002, pp. 277–281 (2002)
6. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for expressive description logics. In: Ganzinger, H., McAllester, D., Voronkov, A. (eds.) LPAR 1999. LNCS, vol. 1705, pp. 161–180. Springer, Heidelberg (1999)
7. Hudek, A.K., Weddell, G.E.: Binary absorption in tableaux-based reasoning for description logics. In: Description Logics 2006 (2006)

8. Kollia, I., Glimm, B.: Cost based query ordering over OWL ontologies. In: Cudré-Mauroux, P., Heflin, J., Sirin, E., Tudorache, T., Euzenat, J., Hauswirth, M., Parreira, J.X., Hendler, J., Schreiber, G., Bernstein, A., Blomqvist, E. (eds.) ISWC 2012, Part I. LNCS, vol. 7649, pp. 231–246. Springer, Heidelberg (2012)
9. Pound, J., Toman, D., Weddell, G.E., Wu, J.: An assertion retrieval algebra for object queries over knowledge bases. In: Walsh, T. (ed.) IJCAI 2011, pp. 1051–1056 (2011)
10. Wandelt, S., Möller, R.: Towards ABox modularization of semi-expressive description logics. Appl. Ontology **7**(2), 133–167 (2012)
11. Wu, J.: Answering object queries over knowledge bases with expressive underlying description logics. Ph.D. thesis, University of Waterloo (2013)
12. Jiewen, W., Hudek, A., Toman, D., Weddell, G.: Absorption for ABoxes. J. Autom. Reasoning **53**, 215–243 (2014)
13. Wu, J., Hudek, A.K., Toman, D., Weddell, G.E.: Absorption for ABoxes. In: DL 2012 (2012)
14. Wu, J., Hudek, A.K., Toman, D., Weddell, G.E.: Assertion absorption in object queries over knowledge bases. In: KR 2012 (2012)