Assertion Absorption in Object Queries over Knowledge Bases

Jiewen Wu, Alexander Hudek, David Toman, and Grant Weddell
{j55wu, akhudek, david, gweddell}@uwaterloo.ca

Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario Canada

Technical Report CS-2012-04

Abstract. We present a novel optimization of evaluating object queries, especially instance queries, over description logic knowledge bases. The method is designed to perform well for large ABoxes and expressive TBoxes, e.g., where background knowledge requires the use of disjunction and negation. The method significantly improves the performance of answering object queries, and particularly so in cases where a large number of concrete feature values are included. We also report on the results of an experimental evaluation that validates the efficacy of the optimization.

1 Introduction

Description logics (DLs) knowledge bases, often formalized as ontologies, have received substantial attention as a means of knowledge representation and reasoning. In general, a DL knowledge base $\mathcal{K}$ describes an application domain as a set of axioms (known as a TBox $\mathcal{T}$) and a set of assertions (an ABox $\mathcal{A}$). These knowledge bases, authored by various domain experts, are prone to inconsistencies emerging in the phase of construction; therefore, a fundamental task over DL knowledge bases is to determine if they are *consistent*. Among the diverse uses of a knowledge base $\mathcal{K}$, answering queries over $\mathcal{K}$ is of paramount importance, typically when the data, i.e., $\mathcal{A}$, consists of a large volume of objects (called instances in DL terminology). In particular, *instance queries* for determining if an object a instantiates a concept C w.r.t a knowledge base $\mathcal{K}$, written $\mathcal{K} \models a : C$, are pivotal for more complex queries. Instance queries, considered to be the basic object queries, are the focus of this paper.

Usually, instance queries are assumed to include consistency checking of $\mathcal{K}$. However, typical workloads for a reasoning service will include far more instance checking tasks than knowledge base consistency tasks. In practice, most of the knowledge bases that queries are issued against are indeed consistent. Thus, the resulting "separation of concerns" can therefore enable technology that is considerably more efficient for such workloads.

We contribute to this development by introducing a novel adaptation of binary absorption for DL knowledge bases and demonstrate that the technique is efficacious for cases that contain thousands of instance checking tasks, particularly so in the case of non-Horn DL $\mathcal{T}$ that precludes the possibility of computing so-called canonical ABoxes $\mathcal{A}'$ from $\mathcal{A}$ (e.g., when disjunction is used in $\mathcal{T}$). Our novel technique can substantially improve performance for instance queries over consistent knowledge bases because it allows for a tableaux algorithm to only explore a (potentially much smaller) subset of $\mathcal{A}$, achieved by the so-called *guarded reasoning* to be elaborated in later sections. To consider performance issues for instance queries in the new optimization, we first consider how one can map instance checking problems to concept satisfaction problems in which consistency is assumed, and then revisit *absorption* in this new setting.

To date, work on absorption has focused on the concept satisfaction problem, a simple case of the instance checking problem for knowledge bases with an ABox consisting of a single assertion $a : \top$. Indeed, it has been known for some time in this case that *lazy unfolding* is an important optimization technique in model building algorithms for satisfiability [BFH+94]. It is also imperative for a large TBox to be manipulated by an *absorption generation* procedure to maximize the benefits of lazy unfolding in such algorithms, thereby reducing the combinatorial effects of disjunction in underlying tableaux procedures [Hor98].

In particular, we present an absorption generation procedure that is an adaptation of an earlier procedure reported in [HW06]. This earlier procedure was called *binary absorption* and was itself a generalization of the absorption theory and algorithms developed in [HT00a,HT00b]. The generalization makes it possible for lazy unfolding to be used for parts of terminologies not handled by earlier absorption algorithms and theory.

Binary absorption combines two key ideas. The first makes it possible to avoid *internalizing* (at least some of the) terminological axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$, where the A_i denote *primitive concepts* and C a general concept. The second is an idea relating to *role absorptions* developed by Tsarkov and Horrocks [TH04]. To illustrate, binary absorption makes it possible to completely absorb the inclusion dependency

$$A_1 \sqcap (\exists R_1^-.A_2) \sqcap (\exists R_2.(A_3 \sqcup A_4)) \sqsubseteq A_5.$$

In this case, the absorption would consist of a set of dependencies with a single primitive concept on the left-hand-side

$$\{A_2 \sqsubseteq \forall R_1.A_6, A_3 \sqsubseteq A_7, A_4 \sqsubseteq A_7, A_7 \sqsubseteq \forall R_2^-.A_8\}$$

and a second set of dependencies with a conjunction of two primitive concepts on the left-hand-side

$$\{(A_1 \sqcap A_6) \sqsubseteq A_9, (A_9 \sqcap A_8) \sqsubseteq A_5\},$$

in which A_6, A_7, A_8 and A_9 are fresh atomic concepts introduced by the binary absorption procedure. (Hereon, we refer to an instance of the latter set as a *binary axiom*.) A key insight and contribution of this paper is that it is not necessary for *both* concepts occurring in the left-hand-side of such a dependency to be atomic. In particular, we show that binary absorption raises the possibility of reducing assertion membership problems to concept satisfaction problems via the introduction of nominals in such dependencies, but without suffering the consequent overhead that doing so would almost certainly entail without binary absorption.

Note that there are other reasons that binary absorption is useful, beyond the well-documented advantages of reducing the need for internalization of general terminological axioms. In particular, it works very well for the parts of a terminology that are Horn-like, as illustrated by the above example.

Our contributions are as follows:

1. We introduce the notion of guards in the context of a knowledge base for the DL dialect $\mathcal{ALCIQ}(\mathbf{D})$. In particular, we show how instance retrieval in this dialect can map to concept satisfaction problems in the dialect $\mathcal{ALCIOQ}(\mathbf{D})$, but where binary absorption in combination with guards can usefully avoid reasoning about irrelevant ABox individuals and concrete facts with the assumption of knowledge base consistency.
2. We propose a generalization of binary absorption. In particular, we now allow nominals in place of one of the two left-hand-side concepts in an absorbed binary axiom.
3. We report on the results of an experimental evaluation that validates the efficacy of the proposed optimization.

This paper is organized as follows: Section 2 introduces some preliminary definitions. Then, Section 3 elaborates the mapping from instance queries to concept satisfaction problems and the generalized binary absorption together with its correctness proof. Section 4 details a procedure for general binary absorption that is capable of absorbing the mapped knowledge bases. An experimental evaluation follows in Section 5 and we present summary comments of this paper in Section 6.

1.1 Related Work

Query answering over DL knowledge bases has been studied extensively in the research community, ranging from investigating query answering complexity over knowledge bases with varying expressiveness to practical optimization techniques for large ontologies. In general, the size of the ABox, viewed as data from a relational database perspective, dominates that of the knowledge base and the query. It is therefore reasonable to consider data complexity as the complexity measure for query answering in knowledge bases, i.e., fixing the TBox and the query.

Instance retrieval over DL knowledge bases is in general more difficult to deal with than concept satisfaction, for example, for $\mathcal{ALE}$ the (data) complexity of instance queries is coNP-hard, while that of concept satisfaction is coNP-complete [DLNS94]. However, for many other DLs the complexity of instance retrieval is in the same complexity class as that of concept satisfaction, i.e., PSpace-complete for $\mathcal{ALC}$ [BCM$^+$03b]. Because of such intrinsic complexity, reasoners requires effective optimization techniques to answer instance queries.

Traditionally, an instance query of the form $\mathcal{K} \models a : C$ is answered by first computing a clash free *pre-completion*, via exhaustive application of non-generating expansion rules, that captures *some* relevant information regarding a w.r.t. $\mathcal{K}$. For DLs without disjunctions, only one pre-completion can be obtained, otherwise, a pre-completion has to be guessed non-deterministically for disjunctions. A similar idea, called *model merging* technique, was also presented in [Hor97] for more expressive DLs, which uses pre-completion obtained by conjunction and disjunction rules. The model merging strategy has been further refined as *pseudo model merging* in [HM08], which appeared to be indispensable for answering queries over large ABoxes.

Notably, the pseudo model merging technique separates consistency checking of $\mathcal{A}$ from instance checking by avoiding exploring other instances occurring in $\mathcal{A}$. Specifically, given a *consistent* $\mathcal{A}$, a pseudo model of an instance a captures the deterministic information relevant to a in one model of $\mathcal{A}$ computed by the tableaux algorithm, which exhibits the interaction between other instances and a. Pseudo models are then used to exclude obvious non-instances of a given concept, based on the observation that an instance usually belongs to a small number of concepts. The experimental evaluation of [HM08] demonstrated the usefulness of pseudo model merging, however, it is a sound but incomplete optimization for instance queries.

Recent advances in scalable query answering focus on "lightweight" DLs in terms of expressiveness. These approaches, instead of answering queries in knowledge bases, leverage relational technology to scale to large ABoxes, provided that the DLs satisfy certain conditions. In addition, the queries supported in such a setting, often referred to as *ontology-based data access* (OBDA) [CDGL$^+$09,KLT$^+$11], go beyond instance retrieval to conjunctive queries. The main idea of OBDA for conjunctive query answering is to rewrite the original query into another query, which can then be evaluated in relational databases that consist of the data (ABoxes) alone, instead of relying on DL reasoning over the knowledge bases. The advantage of OBDA is evident: conjunctive query answering in plain databases is AC^0 [AHV95], making it scalable with the data. However, several restrictions are imposed on the KBs, for example, only queries in certain DLs can be rewritten into FOL formulae (i.e., SQL) that can be evaluated efficiently by a relational DBMS, such as the logics DL-Lite. In reality, knowledge bases adopts expressive logics to model application domains, which consequently requires novel optimizations in place of relational technology to enable efficient query answering.

In this paper, absorption is used to map instance queries into concept satisfaction problems. Absorption aims at transforming *general concept inclusions* (GCIs) into axioms that can be exploited by lazy unfolding. The basic absorption [BCM$^+$03a] rewrites an axiom into the form $A \sqsubseteq C$ where A is atomic, with its counterpart transforming axioms into the form $\neg A \sqsubseteq C$. [HW06] extends the above absorption to axioms of the form $A_1 \sqcap A_2 \sqsubseteq C$, called binary absorption. Furthermore, axioms of the form $\exists R.\top \sqsubseteq C$ could be absorbed by role absorption [TH04]. To our knowledge, Hypertableaux [MSH09], a special reasoning calculus that reduces nondeterminism for expressive DLs, is the first to provide an absorption framework, instead of using a single absorption technique presented above.

2 Preliminaries

We consider instance checking problems in the context of knowledge bases expressed in terms of the DL dialect $\mathcal{ALCIQ}(\mathbf{D})$. However, such problems will be mapped to concept satisfaction problems in the more general dialect $\mathcal{ALCIOQ}(\mathbf{D})$.

Definition 1 (Description Logic $\mathcal{ALCIOQ}(\mathbf{D})$).
$\mathcal{ALCIOQ}(\mathbf{D})$ is a DL dialect based on disjoint infinite sets of atomic concepts NC*, atomic roles* NR*, concrete features* NF *and nominals* NI*. Also, if $A \in$ NC, $R \in$ NR, $a \in$ NI, $f, g \in$ NF, n is a non-negative integer and C_1 and C_2 are concept descriptions, then A, $\neg C_1$, $C_1 \sqcap C_2$, $C_1 \sqcup C_2$, $\top$, $\bot$, $\exists R.C_1$, $\forall R.C_1$, $\exists R^-.C_1$, $\forall R^-.C_1$, $\{a\}$, $\exists^{\leq n} R.C_1$, $\exists^{\geq n} R.C_1$, $\exists^{\leq n} R^-.C_1$, $\exists^{\geq n} R^-.C_1$, $f < g$ and $f = k$, where k is a finite string, are also concept descriptions.*

An interpretation $\mathcal{I}$ is a pair $\mathcal{I} = (\Delta^{\mathcal{I}} \uplus \mathbf{D}^{\mathcal{I}}, \cdot^{\mathcal{I}})$, where $\Delta^{\mathcal{I}}$ 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 $f^{\mathcal{I}} : \Delta \rightarrow \mathbf{D}$, 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 $\Delta^{\mathcal{I}}$,* NR *to subsets of $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and* NI *to elements of $\Delta^{\mathcal{I}}$. The interpretation is extended to inverse roles as follows: $(R^-)^{\mathcal{I}} = \{(o_2, o_1) \in \Delta \times \Delta \mid (o_1, o_2) \in R^{\mathcal{I}}\}$. The interpretation is further extended to compound concept descriptions in the following way:*

$$\top^{\mathcal{I}} = \Delta$$
$$\bot^{\mathcal{I}} = \emptyset$$
$$\neg C^{\mathcal{I}} = \Delta \setminus C^{\mathcal{I}}$$
$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$
$$(\exists S.C)^{\mathcal{I}} = \{o_1 \in \Delta \mid \exists o_2 \in C^{\mathcal{I}} : (o_1, o_2) \in S^{\mathcal{I}}\}$$
$$(\forall S.C)^{\mathcal{I}} = \{o_1 \in \Delta \mid \forall o_2 : (o_1, o_2) \in S^{\mathcal{I}} \rightarrow o_2 \in C^{\mathcal{I}}\}$$
$$(\exists^{\leq n} S.C_1)^{\mathcal{I}} = \{o_1 \in \Delta \mid |\{o_2 \in \Delta \mid (o_1, o_2) \in S^{\mathcal{I}} \wedge o_2 \in C^{\mathcal{I}}\}| \leq n\}$$
$$(\exists^{\geq n} S.C_1)^{\mathcal{I}} = \{o_1 \in \Delta \mid |\{o_2 \in \Delta \mid (o_1, o_2) \in S^{\mathcal{I}} \wedge o_2 \in C^{\mathcal{I}}\}| \geq n\},$$

where S is either an atomic role R or its inverse R^-.

Some abbreviations are used in this paper: $\forall S.C$ is considered to be a shorthand for $\exists^{\leq 0} S.\neg C$; concrete domain concepts such as $f < k$ are an abbreviation for $(f < g) \sqcap (g = k)$; $(t_1 \ \mathtt{op} \ t_2)$ generalizes concrete domain concepts by allowing t_1 and t_2 to be either a concrete feature or a finite string and $\mathtt{op} \in \{<, =\}$.

Definition 2 (TBox, ABox, and KB Satisfiability).
A TBox $\mathcal{T}$ *is a finite set of axioms of the form* $C_1 \sqsubseteq C_2$ *or* $C_1 \doteq C_2$*. A TBox* $\mathcal{T}$ *is called* primitive *iff it consists entirely of axioms of the form* $A \doteq C$ *with* $A \in \mathrm{NC}$*, each* $A \in \mathrm{NC}$ *appears in at most one left hand side of an axiom, and* $\mathcal{T}$ *is* acyclic*. Acyclicity is defined as follows:* $A_1 \in \mathrm{NC}$ directly uses $A_2 \in \mathrm{NC}$ *if* $A_1 \doteq C \in \mathcal{T}$ *and* A_2 *occurs in* C*; "uses" is the transitive closure of "directly uses". Then* $\mathcal{T}$ *is* acyclic *if there is no* $A \in \mathrm{NC}$ *that uses itself.* $A \in \mathrm{NC}$ is defined in $\mathcal{T}$ *if* $\mathcal{T}$ *contains* $A \sqsubseteq C$ *or* $A \doteq C$*. An* ABox $\mathcal{A}$ *is a finite set of assertions of the form* $a : A$*,* $a : (f \ \mathsf{op}\ k)$ *and* $R(a, b)$*.*

Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be an* $\mathcal{ALCIOQ}(\mathbf{D})$ knowledge base *(KB). An interpretation* $\mathcal{I}$ *is a* model *of* $\mathcal{K}$*, written* $\mathcal{I} \models \mathcal{T}$*, iff* $C_1^{\mathcal{I}} \subseteq C_2^{\mathcal{I}}$ *holds for each* $C_1 \sqsubseteq C_2 \in \mathcal{T}$*,* $C_1^{\mathcal{I}} = C_2^{\mathcal{I}}$ *holds for each* $C_1 \doteq C_2 \in \mathcal{T}$*,* $a^{\mathcal{I}} \in A^{\mathcal{I}}$ *for* $a : A \in \mathcal{A}$*,* $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$*, and* $f^{\mathcal{I}}(a^{\mathcal{I}})\ \mathsf{op}\ \ k$ *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* $C^{\mathcal{I}} \neq \emptyset$*.*

Observe that in Definition 2 ABox assertions of the form $a : A$ and $a : (f\ \mathsf{op}\ k)$ disallow the use of compound concepts. Any ABoxes that violate the stipulation can be rectified by introducing fresh atomic concepts in the corresponding assertions and appropriate axioms in the TBoxes. Once an ABox $\mathcal{A}$ is normalized as in the definition, it can then be converted using the technique presented in the next section.

3 On Absorbing an ABox

The absorption of an ABox $\mathcal{A}$ proceeds in two steps. First, *guards* that allow a DL reasoner to ease the exploration of $\mathcal{A}$ are added to the assertions (which are in turn converted into TBox axioms via the use of nominals). During the first step, $\mathcal{A}$ will be converted to a TBox $\mathcal{T}_{\mathcal{A}}$ and the original TBox $\mathcal{T}$ to a new TBox $\mathcal{T} \cup \mathcal{T}_{\mathcal{T}}$. Second, we adapt binary absorption to deal with the resulting TBox $\mathcal{T} \cup \mathcal{T}_{\mathcal{T}} \cup \mathcal{T}_{\mathcal{A}}$.

3.1 Mapping instance checking to concept satisfaction

In this section we convert an $\mathcal{ALCIQ}(\mathbf{D})$ knowledge base $\mathcal{K}$ to a TBox by representing individuals in $\mathcal{K}$'s ABox by nominals (i.e., in a *controlled* fragment of $\mathcal{ALCIOQ}(\mathbf{D})$):

Definition 3 (ABox Conversion). *Let* $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ *be a knowledge base. We define a TBox* $\mathcal{T}_{\mathcal{A}}$ *for the ABox of* $\mathcal{K}$*:*

$$\begin{aligned}
\mathcal{T}_{\mathcal{A}} = \{ & \{a\} \sqcap \mathrm{Def}_a \sqsubseteq \mathrm{A} \mid a : \mathrm{A} \in \mathcal{A}\} \\
\cup \{ & \{a\} \sqcap \mathrm{Def}_f \sqsubseteq (f\ \mathsf{op}\ k) \mid a : (f\ \mathsf{op}\ k) \in \mathcal{A}\} \\
\cup \{ & \{a\} \sqcap \mathrm{Def}_R \sqsubseteq \exists R.(\{b\} \sqcap \mathrm{Def}_b), \{a\} \sqcap \mathrm{Def}_a \sqsubseteq \exists R.\top, \\
& \{b\} \sqcap \mathrm{Def}_{R^-} \sqsubseteq \exists R^-.(\{a\} \sqcap \mathrm{Def}_a), \{b\} \sqcap \mathrm{Def}_b \sqsubseteq \exists R^-.\top \mid R(a, b) \in \mathcal{A}\}
\end{aligned}$$

Note that all the axioms resulting from ABox assertions are *guarded* by auxiliary primitive concepts of the form Def_a, Def_R, and Def_f. Intuitively, these concepts, when coupled with an appropriate absorption allow a reasoner to *ignore* parts of the original ABox: all the constants for which Def_a is not *set*. Similarly, for any instance, a reasoner only examines the relevant concrete domain concepts that have the guard Def_f set and only explores the relevant instances that have the guard Def_R or Def_{R^-} set. We show later that the guarding concepts with the extended binary absorption can yield considerable performance gains.

To make guarding fully functional, it is necessary (without loss of generality) that the TBox of $\mathcal{K}$ only use qualified *at-most* number restrictions of the form $\mathrm{A} \sqsubseteq \exists^{\leq n} R.\mathrm{B}$ where A and B are atomic concepts or their negations. Axioms of the form $\exists^{\geq n} R.\mathrm{A} \sqsubseteq \mathrm{B}$ are also considered to be at-most number restrictions and have to be rewritten in the form of $\neg B \sqsubseteq \exists^{\leq n} R.\mathrm{A}$ and that nested qualified number restrictions must be unnested. It is easy to see that every $\mathcal{ALCIQ}(\mathbf{D})$ TBox can be transformed to an equi-satisfiable TBox that satisfies this requirement by introducing new auxiliary atomic concepts. If the TBox $\mathcal{T}$ satisfies the aforementioned requirement, additional axioms have to be introduced to manipulate the guards:

Definition 4 (TBox Augmentation). *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a knowledge base. We define a TBox $\mathcal{T}_\mathcal{T}$ for the ABox of $\mathcal{K}$ as follows:*

$$\begin{aligned}\mathcal{T}_\mathcal{T} = &\{\mathrm{A} \sqsubseteq \mathrm{Def}_R, \mathrm{B} \sqsubseteq \mathrm{Def}_{R^-} \mid \mathrm{A} \sqsubseteq \exists^{\leq n} R.\mathrm{B} \in \mathcal{T}\}\\ &\cup \{(t_1 \ \mathsf{op}\ t_2) \sqsubseteq \mathrm{Def}_f \mid f \textit{ appears in } t_1 \textit{ or in } t_2, (t_1 \ \mathsf{op}\ t_2) \textit{ appears in } \mathcal{T}\}.\end{aligned}$$

Intuitively, an instance guard is generated on the fly only if that particular instance interacts with other guarded instances via guarded roles, as shown in Definition 3. The augmentation of the original TBox (Definition 4) further supplies guards for concrete domain concepts and roles to be used in guarded reasoning.

In the following, we present the main theoretical result that enables a DL reasoner to view instance checking tasks as concept satisfaction problems. We write $\mathcal{T}_\mathcal{K}$ for $\mathcal{T} \cup \mathcal{T}_\mathcal{T} \cup \mathcal{T}_\mathcal{A}$. To ease the presentation, we define a *derivative guards* $\mathfrak{D}_C$ w.r.t. C as follows: $\mathfrak{D}_A = \top$, $\mathfrak{D}_{(t_1 \ \mathsf{op}\ t_2)} = \mathrm{Def}_{f_1} \sqcap \cdots \sqcap \mathrm{Def}_{f_n}$ where f_i, $1 \leq i \leq n$, appears in t_1 or t_2, $\mathfrak{D}_{\exists R.C} = \mathfrak{D}_{\exists^{\leq n} R.C} = \mathrm{Def}_R \sqcap \forall R.\mathfrak{D}_C$, $\mathfrak{D}_{C_1 \sqcap C_2} = \mathfrak{D}_{C_1} \sqcap \mathfrak{D}_{C_2}$, and $\mathfrak{D}_{\neg C} = \mathfrak{D}_C$.

Theorem 1. *Let $\mathcal{K} = (\mathcal{T}, \mathcal{A})$ be a consistent knowledge base. Then*

$$\mathcal{K} \models a : \mathrm{C} \textit{ if and only if } \mathcal{T}_\mathcal{K} \models \{a\} \sqcap \mathrm{D} \sqsubseteq \mathrm{C},$$

where $\mathrm{D} = \mathrm{Def}_a \sqcap \mathfrak{D}_C$.

Proof. The *only-if* direction is equivalent to the following claim to be proven: if $\mathcal{T}_\mathcal{K} \not\models \{a\} \sqcap \mathrm{D} \sqsubseteq \mathrm{C}$, then $\mathcal{K} \not\models a : \mathrm{C}$. Assume that there is an interpretation I_0 that satisfies $\mathcal{T}_\mathcal{K}$ such that $(\{a\})^{I_0} \subseteq (\mathrm{D})^{I_0}$ but $(\{a\})^{I_0} \cap (\mathrm{C})^{I_0} = \emptyset$ and an interpretation I_1 that satisfies $\mathcal{K}$ in which *all at-least restrictions are fulfilled by anonymous objects*. Hence, we do not need to consider at-least restrictions, no matter how expressed, in the construction below. Without loss of generality, we assume both I_0 and I_1 are tree-shaped outside of the ABox (converted ABox). Our proof proceeds by building an interpretation J such that J satisfies $\mathcal{K}$ and $(a)^J \notin (C)^J$.

The construction of the interpretation J for $\mathcal{K} \cup \{a : \neg \mathrm{C}\}$ follows. Let Γ^{I_0} be the set of objects $o \in \Delta^{I_0}$ such that either $o \in (\{a\})^{I_0}$ and $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a)^{I_0}$ or o is an anonymous object in Δ^{I_0} rooted by such an object. Similarly let Γ^{I_1} be the set of objects $o \in \Delta^{I_1}$ such that either $o \in (\{a\})^{I_1}$ and $(\{a\})^{I_0} \cap (\mathrm{Def}_a)^{I_0} = \emptyset$ or o is an anonymous object in Δ^{I_1} rooted by such an object. We stipulate the following protocols for constructing J:

1. $\Delta^J = \Gamma^{I_0} \cup \Gamma^{I_1}$;
2. $(a)^J \in (\{a\})^{I_0}$ for $(a)^J \in \Gamma^{I_0}$ and $(a)^J = (a)^{I_1}$ for $(a)^J \in \Gamma^{I_1}$;
3. $o \in A^J$ if $o \in A^{I_0}$ and $o \in \Gamma^{I_0}$ or if $o \in A^{I_1}$ and $o \in \Gamma^{I_1}$ for an atomic concept A (similarly for concrete domain concepts of the form $(t_1 \;\mathsf{op}\; t_2)$);
4. $(o_1, o_2) \in (R)^J$ if
   (a) $(o_1, o_2) \in R^{I_0}$ and $o_1, o_2 \in \Gamma^{I_0}$, or $(o_1, o_2) \in R^{I_1}$ and $o_1, o_2 \in \Gamma^{I_1}$; or
   (b) $o_1 \in (\{a\})^{I_0} \cap (\mathrm{Def}_a)^{I_0}$, $o_2 \in (\{b\})^{I_1}$ and $R(a, b) \in \mathcal{A}$ (or vice versa).

It should be noted that a is in the initial query, hence, $(a)^J \in \Gamma^{I_0}$ and $(a)^J \in (\mathrm{Def}_a)^{I_0}$. In addition, we have $(\{a\})^{I_0} \subseteq (\mathrm{D})^{I_0}$ and $(\{a\})^{I_0} \cap (\mathrm{C})^{I_0} = \emptyset$ as our assumption. We show that $(a)^J \notin (C)^J$ holds using structural induction, considering the query concept C to be in NNF. Observe that the induction hypothesis is that, for any a, if $(\{a\})^{I_0} \subseteq (\mathrm{D}')^{I_0}$ and $(\{a\})^{I_0} \cap (\mathrm{C}')^{I_0} = \emptyset$ then $(a)^J \notin (C')^J$.

- $C = A$ or $C = (t_1 \;\mathsf{op}\; t_2)$, then $(a)^J \notin (C)^J$ holds trivially by protocol 3: $(a)^J \in \Gamma^{I_0}$ and $(a)^J \notin C^{I_0}$ (due to our assumption that $(\{a\})^{I_0} \cap (\mathrm{C})^{I_0} = \emptyset$).
- $C = \exists R.C'$. By way of contradiction, we assume $(a)^J \in (\exists R.C')^J$, which means there is an individual b such that $(b)^J \in (C')^J$ and $((a)^J, (b)^J) \in (R)^J$.
  1. If $(b)^J \in \Gamma^{I_0}$, then $(b)^J \in (\{b\})^{I_0}, ((\{a\})^{I_0}, (\{b\})^J) \in (R)^{I_0}$. By our initial assumption, $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a \sqcap \mathfrak{D}_C)^{I_0}$, where $\mathfrak{D}_C = \mathrm{Def}_R \sqcap \forall R.\mathfrak{D}_{C'}$, hence, we have $(\{b\})^{I_0} \subseteq (\mathfrak{D}_{C'})^{I_0}$, which, together with $(\{b\})^{I_0} \subseteq (\mathrm{Def}_b)^{I_0}$, implies that $(\{b\})^{I_0} \subseteq (D')^{I_0}$. However, by the induction hypothesis and $(b)^J \in (C')^J$, either $(\{b\})^{I_0} \nsubseteq (D')^{I_0}$ or $(\{b\})^{I_0} \subseteq (\mathrm{C}')^{I_0}$, so the latter must hold. The latter, nevertheless, contradicts our original assumption that $(\{a\})^{I_0} \cap (\mathrm{C})^{I_0} = \emptyset$, with $C = \exists R.C'$.
  2. If $(b)^J \in \Gamma^{I_1}$, then $(\{b\})^{I_0} \cap (\mathrm{Def}_b)^{I_0} = \emptyset$. By our initial assumption, $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a \sqcap \mathfrak{D}_C)^{I_0}$, where $\mathfrak{D}_C = \mathrm{Def}_R \sqcap \forall R.\mathfrak{D}_{C'}$, hence, $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a \sqcap \mathrm{Def}_R)^{I_0}$. Since $R(a, b) \in \mathcal{A}$, by Definition 3 it is easy to see that $(\{b\})^{I_0} \subseteq (\mathrm{Def}_b)^{I_0}$: a contradiction.
- $C = \forall R.C'$. Observe that $\forall R.C'$ is a shorthand for $\exists^{\leq 0} R.\neg C'$, so $\mathfrak{D}_{\forall R.C'} = \mathfrak{D}_{\exists R.C'}$. Assume, by way of contradiction, that $(a)^J \in (\forall R.C')^J$. Because $(\{a\})^{I_0} \nsubseteq (\forall R.C')^{I_0}$, there is a nominal $\{b\}$ such that $((\{a\})^{I_0}, (\{b\})^{I_0}) \in (R)^{I_0}$ and $(\{b\})^{I_0} \nsubseteq (C')^{I_0}$. Thus, by definition $(\{b\}) \in \Gamma^{I_0}$ and by protocol 4a $((\{a\})^J, (\{b\})^J) \in (R)^J$. Since $(a)^J \in (\forall R.C')^J$, it must be that $(b)^J \in (C')^J$. However, $(\{a\})^{I_0} \subseteq (\mathrm{D})^{I_0}$ implies that $(\{a\})^{I_0} \subseteq (\forall R.\mathfrak{D}_{C'})^{I_0}$, which means $(\{b\})^{I_0} \subseteq (\mathfrak{D}_{C'})^{I_0}$, hence, by the induction hypothesis, $(b)^J \notin (C')^J$: a contradiction.

- $C = \exists^{\leq n} R.C'$. The proof follows immediately from the cases $C = \forall R.\neg C'$ if $n = 0$. Otherwise, by assuming $(a)^J \in (\exists^{\leq n} R.C')^J$, the case analyses are the same as that of the case $C = \exists R.C'$.
- $C = \neg A$. Because $(\{a\})^{I_0} \cap C^{I_0} = \emptyset$, we have $(\{a\})^{I_0} \subseteq (A)^{I_0}$, which by protocol 3, together with $(a)^J \in (\{a\})^{I_0}$ and $(a)^J \in \Gamma^{I_0}$, implies that $(a)^J \in A^J$, i.e., $(a)^J \notin (C)^J$.
- $C = C_1 \sqcap C_2$. Assume, by way of contradiction, that $(a)^J \in (C)^J$, then $(a)^J \in (C_1)^J$ and $(a)^J \in (C_2)^J$. By the induction hypothesis and $(a)^J \in (C_1)^J$, we have either $(\{a\})^{I_0} \not\subseteq (\mathrm{D}_1)^{I_0}$ or $(\{a\})^{I_0} \subseteq (\mathrm{C}_1)^{I_0}$. Because it holds that $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a)^{I_0}, (\{a\})^{I_0} \subseteq (\mathfrak{D}_C)^{I_0}$ and $\mathfrak{D}_C \sqsubseteq \mathfrak{D}_{C_1}$, it follows that $(\{a\})^{I_0} \subseteq (\mathrm{Def}_a \sqcap \mathfrak{D}_{C_1})^{I_0} = (\mathrm{D}_1)^{I_0}$, so it must be the case that $(\{a\})^{I_0} \subseteq (\mathrm{C}_1)^{I_0}$. For the same reason, $(\{a\})^{I_0} \subseteq (\mathrm{C}_2)^{I_0}$. Hence, $(\{a\})^{I_0} \subseteq (\mathrm{C}_1 \sqcap C_2)^{I_0}$, i.e., $(\{a\})^{I_0} \subseteq (C)^{I_0}$, which contradicts the initial assumption that $(\{a\})^{I_0} \cap (\mathrm{C})^{I_0} = \emptyset$.

To show $J \models \mathcal{K}$, it suffices to consider only the R edges crossing the two interpretations as defined in protocol 4b, i.e., when $o_1 \in (\{a\})^{I_0}$, $o_2 \in (\{b\})^{I_1}$ and $R(a, b) \in \mathcal{A}$. Note that none of these edges need to fulfill existential restrictions, which are already fulfilled (potentially redundantly) by anonymous objects whenever possible. Therefore, only at-most restrictions (including universal restrictions) need to be considered.

For any inclusion axiom expressing an *at-most* restriction $\mathrm{A} \sqsubseteq \exists^{\leq n} R.\mathrm{B} \in \mathcal{T}$, we can conclude that $o_1 \notin (\mathrm{A})^{I_0}$ as otherwise $o_1 \in (\mathrm{Def}_R)^{I_0}$ by Definition 4 and thus $o_2 \in (\mathrm{Def}_b)^{I_0}$ by Definition 3 which contradicts our assumption that $(\{b\})^{I_0} \cap (\mathrm{Def}_b)^{I_0} = \emptyset$. Hence the axiom is satisfied vacuously. The remaining edges in protocol 4a satisfy all axioms in $\mathcal{K}$ as the remainder of the interpretation J is copied from one of the two interpretations that satisfy $\mathcal{K}$. Hence all inclusion axioms in $\mathcal{K}$ are satisfied by J.

The *if* direction, equivalent to the claim that if $\mathcal{K} \not\models a : \mathrm{C}$ then $\mathcal{T}_{\mathcal{K}} \not\models \{a\} \sqcap \mathrm{D} \sqsubseteq \mathrm{C}$, holds by observing that if $\mathcal{K} \cup \{a : \neg\mathrm{C}\}$ is satisfiable then the satisfying interpretation I can be extended to $(\mathrm{Def}_a)^I = (\mathrm{Def}_f)^I = (\mathrm{Def}_R)^I = \Delta^I$ and $(\{a\})^I = \{a^I\}$ for all individuals a, concrete features f, and roles R. This extended interpretation then satisfies $\mathcal{T}_{\mathcal{K}}$ and $(\{a\})^I \subseteq (\mathrm{D})^I \cap (\neg\mathrm{C})^I$.

□

In $\mathcal{ALCIQ}(\mathbf{D})$, which on its own cannot equate constants, we do not need to rely explicitly on the unique name assumption (UNA). However, we could allow explicit equalities and inequalities in the ABox and then process them similarly to Definition 3, e.g., $a \approx b$ to $\{a\} \sqcap \mathrm{Def}_a \sqsubseteq \{b\} \sqcap \mathrm{Def}_b$ and vice versa and so on. This is sufficient for the construction of the interpretation J in the proof of Theorem 1 to go through. Note that the interpretations of constants (nominals) for which Def_a is not set in I_0 are irrelevant for constructing the interpretation J even though there could be axioms of the form $\top \sqsubseteq \mathrm{C}$ that are applicable to such constants (one could even augment all such axioms by adding guards to avoid this effect). Therefore those constants (nominals) can be ignored completely during reasoning and, thus, nodes corresponding to the constant symbols can be generated *lazily* on demand driven by the guarding concept Def_a.

The guarding concepts is effective only if they are "observed" simultaneously with a constant. Binary absorption, when properly extended, ensures that guarded constants are reasoned about only if a guard is seen. Nevertheless, using other absorption algorithms is unable to retain the effects of guards. Consequently, Section 3.3 expands on an extension to binary absorption that functions on the converted knowledge base $\mathcal{T}_{\mathcal{K}}$. To define such an extension, the notion of witnesses needs to be introduced prior to absorption, as shown in the next section.

3.2 On witnesses

Model building algorithms for checking the satisfaction of a concept C operate by manipulating an internal data structure (e.g., in the form of a node and edge labeled rooted tree with "back edges"). The data structure "encodes" a *partial description* of (eventual) interpretations $\mathcal{I}$ for which $C^{\mathcal{I}}$ will be non-empty. Such a partial description will almost always abstract details on class membership for hypothetical elements of $\Delta^{\mathcal{I}}$ and on details relating to the interpretation of roles. To talk formally about absorption and lazy evaluation, it is necessary to codify the idea of a partial description. This has been done in [HT00b] by introducing the notion of a *witness*, of an interpretation that *stems* from a witness, and of what it means for a witness to be *admissible* with respect to a given terminology.

Definition 5. (Witness) *Let C be an $\mathcal{ALCIOQ}(\mathbf{D})$ concept.*[1] *A* witness $\mathcal{W} = (\Delta^{\mathcal{W}}, \cdot^{\mathcal{W}}, \mathcal{L}^{\mathcal{W}})$ *for C consists of a non-empty set $\Delta^{\mathcal{W}}$, a function $\cdot^{\mathcal{W}}$ that maps* NR *to subsets of $\Delta^{\mathcal{W}} \times \Delta^{\mathcal{W}}$, and a function $\mathcal{L}^{\mathcal{W}}$ that maps $\Delta^{\mathcal{W}}$ to sets of $\mathcal{ALCIOQ}(\mathbf{D})$ concepts such that:*

- *(W1) there is some $x \in \Delta^{\mathcal{W}}$ with $C \in \mathcal{L}^{\mathcal{W}}(x)$,*
- *(W2) there is an interpretation $\mathcal{I}$ that* stems *from $\mathcal{W}$, and*
- *(W3) for each $\mathcal{I}$ that* stems *from $\mathcal{W}$, $x \in C^{\mathcal{I}}$ if $C \in \mathcal{L}^{\mathcal{W}}(x)$.*

An interpretation $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ is said to stem *from $\mathcal{W}$ if $\Delta^{\mathcal{I}} = \Delta^{\mathcal{W}}$, $\cdot^{\mathcal{I}}|_{\mathrm{NR}} = \cdot^{\mathcal{W}}$, for each $A \in \mathrm{NC}$, $A \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \in A^{\mathcal{I}}$ and $\neg A \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \notin A^{\mathcal{I}}$, for each $a \in \mathrm{NI}$, $\{a\} \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \in \{a\}^{\mathcal{I}}$ and $\neg\{a\} \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \notin \{a\}^{\mathcal{I}}$, for each $(f \ \mathsf{op}\ k)$, $(f \ \mathsf{op}\ k) \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \in (f \ \mathsf{op}\ k)^{\mathcal{I}}$ and $\neg(f \ \mathsf{op}\ k) \in \mathcal{L}^{\mathcal{W}}(x)$ implies $x \notin (f \ \mathsf{op}\ k)^{\mathcal{I}}$.*

A witness $\mathcal{W}$ is called admissible *with respect to a TBox $\mathcal{T}$ if there is an interpretation $\mathcal{I}$ that stems from $\mathcal{W}$ with $\mathcal{I} \models \mathcal{T}$.*

The properties satisfied by a witness are presented in the following lemmas, originally shown in [HT00b].

Lemma 1. *Let L be a DL. A concept $C \in L$ is satisfiable w.r.t. a TBox $\mathcal{T}$ iff it has a witness that is admissible w.r.t. $\mathcal{T}$.*

[1] The definition of witness can be abstracted for any DLs that have $\mathcal{ALCIO}$ as a sublanguage and that satisfy some criteria on the interpretations stated in [HT00b].

$$
\begin{aligned}
\{a\} \in \mathcal{L}^{\mathcal{W}}(x) \ \text{ and } \ \{a\} \in \mathcal{L}^{\mathcal{W}}(y) &\text{ implies } x = y \\
\{\{a\}, A\} \subseteq \mathcal{L}^{\mathcal{W}}(x) \text{ , and } \ (\{a\} \sqcap A) \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(x) \\
(x, y) \in R^{\mathcal{I}} \ \text{ and } \ \exists R.\top \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(x) \\
(x, y) \in R^{\mathcal{I}} \ \text{ and } \ \exists R^{-}.\top \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(y) \\
\{A_1, A_2\} \subseteq \mathcal{L}^{\mathcal{W}}(x) \ \text{ and } \ (A_1 \sqcap A_2) \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(x) \\
A \in \mathcal{L}^{\mathcal{W}}(x) \ \text{ and } \ A \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(x) \\
\neg A \in \mathcal{L}^{\mathcal{W}}(x) \ \text{ and } \ \neg A \sqsubseteq C \in \mathcal{T}_u &\text{ implies } C \in \mathcal{L}^{\mathcal{W}}(x) \\
C_1 \sqsubseteq C_2 \in \mathcal{T}_g &\text{ implies } \neg C_1 \sqcup C_2 \in \mathcal{L}^{\mathcal{W}}(x) \\
C_1 \doteq C_2 \in \mathcal{T}_g &\text{ implies } \neg C_1 \sqcup C_2 \in \mathcal{L}^{\mathcal{W}}(x) \\
C_1 \doteq C_2 \in \mathcal{T}_g &\text{ implies } C_1 \sqcup \neg C_2 \in \mathcal{L}^{\mathcal{W}}(x)
\end{aligned}
$$

Fig. 1: Absorption witness conditions

Lemma 2. *Let L, C, $\mathcal{T}$ and $\mathcal{W}$ be a DL, a concept in L, a TBox for L and a witness for C, respectively. Then $\mathcal{W}$ is admissible w.r.t. $\mathcal{T}$ if, for each $x \in \Delta^{\mathcal{W}}$:*

$$
\begin{aligned}
&C_1 \sqsubseteq C_2 \in \mathcal{T} \textit{ implies } \neg C_1 \sqcup C_2 \in \mathcal{L}^{\mathcal{W}}(x), \\
&C_1 \doteq C_2 \in \mathcal{T} \textit{ implies } \neg C_1 \sqcup C_2 \in \mathcal{L}^{\mathcal{W}}(x) \textit{ and} \\
&C_1 \doteq C_2 \in \mathcal{T} \textit{ implies } C_1 \sqcup \neg C_2 \in \mathcal{L}^{\mathcal{W}}(x).
\end{aligned}
$$

A generalization of an *absorption* developed in [HT00a,HT00b] has been given in [HW06], dubbed *binary absorption*. We further extend binary absorption [HW06] to accommodate the absorbed ABoxes as shown in Section 3.1.

3.3 On binary absorption

Definition 6. (Binary Absorption) *Let $\mathcal{K}$=$\{\mathcal{T}, \mathcal{A}\}$ be a KB. A* binary absorption *of $\mathcal{T}$ is a pair of TBoxes $(\mathcal{T}_u, \mathcal{T}_g)$ such that $\mathcal{T} \equiv \mathcal{T}_u \cup \mathcal{T}_g$ and $\mathcal{T}_u$ contains axioms of the form $A_1 \sqsubseteq C$, $\neg A_1 \sqsubseteq C$, $\exists R.\top \sqsubseteq C$ (resp. $\exists R^{-}.\top \sqsubseteq C$), and the form $(A_1 \sqcap A_2) \sqsubseteq C$ and $(\{a\} \sqcap A) \sqsubseteq C$, where $\{A, A_1, A_2\} \subseteq \mathrm{NC}$ and $a \in \mathrm{NI}$.*

A binary absorption $(\mathcal{T}_u, \mathcal{T}_g)$ of $\mathcal{T}$ is called correct *if it satisfies the following condition: For each witness $\mathcal{W}$ and $x \in \Delta^{\mathcal{W}}$, if all conditions in Figure 1 are satisfied, then $\mathcal{W}$ is admissible w.r.t. $\mathcal{T}$. A witness that satisfies the above property will be called* unfolded.

The distinguishing feature of our extension of binary absorption is the addition of the first four implications in Figure 1. Binary absorption itself allows additional axioms in $\mathcal{T}_u$ to be dealt with in a deterministic manner, as illustrated in our introductory example. ABox absorption, treating assertions as axioms, extends binary absorption to handle nominals in binary axioms. In addition, domain and range constraints are also absorbed in a manner that resembles role absorption introduced in [TH04].

Lemmas 3, 4 and 5 originally presented in [HT00b] hold without modification. We show in Lemma 6 that the generalized binary absorption is also a correct absorption.

Lemma 3. *Let $(\mathcal{T}_u, \mathcal{T}_g)$ be a correct binary absorption of $\mathcal{T}$. For any $C \in L$, C has a witness that is admissible w.r.t. $\mathcal{T}$ iff C has an unfolded witness.*

Lemma 4. *Let $\mathcal{T}$ be a primitive TBox and $\mathcal{T}_u$ defined as*

$$\{A \sqsubseteq C, \neg A \sqsubseteq \neg C \mid A \doteq C \in \mathcal{T}\}.$$

Then $(\mathcal{T}_u, \emptyset)$ is a correct absorption of $\mathcal{T}$.

Lemma 5. *Let $(\mathcal{T}_u, \mathcal{T}_g)$ be a correct absorption of a TBox $\mathcal{T}$.*

1. *If $\mathcal{T}'$ is an arbitrary TBox, then $(\mathcal{T}_u, \mathcal{T}_g \cup \mathcal{T}')$ is a correct absorption of $\mathcal{T} \cup \mathcal{T}'$.*
2. *If $\mathcal{T}'$ is a TBox that consists entirely of axioms of the form $A \sqsubseteq C$, where $A \in \mathrm{NC}$ and A is not defined in $\mathcal{T}_u$, then $(\mathcal{T}_u \cup \mathcal{T}', \mathcal{T}_g)$ is a correct absorption of $\mathcal{T} \cup \mathcal{T}'$.*

Lemma 6. *Let $(\mathcal{T}_u, \mathcal{T}_g)$ be a correct absorption of a TBox $\mathcal{T}$. If $\mathcal{T}'$ is a TBox that consists entirely of axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$ and $(\{a\} \sqcap A_3) \sqsubseteq D$, where $\{A_1, A_2, A_3\} \subseteq \mathrm{NC}$ and where none of A_1, A_2, A_3 are defined in $\mathcal{T}_u$, $a \in \mathrm{NI}$, then $(\mathcal{T}_u \cup \mathcal{T}', \mathcal{T}_g)$ is a correct absorption of $\mathcal{T} \cup \mathcal{T}'$.*

Proof. We only show the proof for axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$, and the proof of axioms of the form $(\{a\} \sqcap A_3) \sqsubseteq D$ follows, viewing $\{a\}$ as a primitive concept.

Observe that $\mathcal{T}_u \cup \mathcal{T}_g \cup \mathcal{T}' \equiv \mathcal{T} \cup \mathcal{T}'$ holds trivially. Let $C \in L$ be a concept and $\mathcal{W}$ be an unfolded witness for C w.r.t. the absorption $(\mathcal{T}_u \cup \mathcal{T}', \mathcal{T}_g)$. From $\mathcal{W}$, define a new witness $\mathcal{W}'$ for C by setting $\Delta^{\mathcal{W}'} = \Delta^{\mathcal{W}}$, $\cdot^{\mathcal{W}'} = \cdot^{\mathcal{W}}$, and defining $\mathcal{L}^{\mathcal{W}'}$ to be the function that, for every $x \in \Delta^{\mathcal{W}'}$, maps x to the set

$$\begin{aligned}\mathcal{L}^{\mathcal{W}}(x) \cup \{&\neg A_1, \neg A_2 \mid (A_1 \sqcap A_2) \sqsubseteq C' \in \mathcal{T}', \{A_1, A_2\} \cap \mathcal{L}^{\mathcal{W}}(x) = \emptyset\} \\ \cup \{&\neg A_1 \mid (A_1 \sqcap A_2) \sqsubseteq C' \in \mathcal{T}', A_1 \notin \mathcal{L}^{\mathcal{W}}(x), A_2 \in \mathcal{L}^{\mathcal{W}}(x)\} \\ \cup \{&\neg A_2 \mid (A_1 \sqcap A_2) \sqsubseteq C' \in \mathcal{T}', A_1 \in \mathcal{L}^{\mathcal{W}}(x), A_2 \notin \mathcal{L}^{\mathcal{W}}(x)\}.\end{aligned}$$

It is easy to see that $\mathcal{W}'$ is also unfolded w.r.t. the absorption $(\mathcal{T}_u \cup \mathcal{T}', \mathcal{T}_g)$. This implies that $\mathcal{W}'$ is also unfolded w.r.t. the (smaller) absorption $(\mathcal{T}_u, \mathcal{T}_g)$. Since $(\mathcal{T}_u, \mathcal{T}_g)$ is a correct absorption of $\mathcal{T}$, there exists an interpretation $\mathcal{I}$ stemming from $\mathcal{W}'$ such that $\mathcal{I} \models \mathcal{T}$. We show that $\mathcal{I} \models \mathcal{T}'$ also holds. Assume $\mathcal{I} \not\models \mathcal{T}'$. Then there is an axiom $(A_1 \sqcap A_2) \sqsubseteq C_1 \in \mathcal{T}'$ and an $x \in \Delta^{\mathcal{I}}$ such that $x \in (A_1 \sqcap A_2)^{\mathcal{I}}$ but $x \notin C_1^{\mathcal{I}}$. By construction of $\mathcal{W}'$, $x \in (A_1 \sqcap A_2)^{\mathcal{I}}$ implies $\{A_1, A_2\} \subseteq \mathcal{L}^{\mathcal{W}'}(x)$ because otherwise $\{\neg A_1, \neg A_2\} \cap \mathcal{L}^{\mathcal{W}'}(x) \neq \emptyset$ would hold in contradiction to (W3). Then, since $\mathcal{W}'$ is unfolded, $C_1 \in \mathcal{L}^{\mathcal{W}'}(x)$, which, again, by (W3), implies $x \in C_1^{\mathcal{I}}$, a contradiction.

Hence, we have shown that there exists an interpretation $\mathcal{I}$ stemming from $\mathcal{W}'$ such that $\mathcal{I} \models \mathcal{T}_u \cup \mathcal{T}' \cup \mathcal{T}_g$. By construction of $\mathcal{W}'$, any interpretation stemming from $\mathcal{W}'$ also stems from $\mathcal{W}$, hence $\mathcal{W}$ is admissible w.r.t. $\mathcal{T} \cup \mathcal{T}'$.

4 A Procedure for ABox Absorption

In this section, we present a procedure for ABox absorption that works on arbitrary axioms obtained from an initial knowledge base as shown in Section 3.1, extending binary absorptions [HW06] with the following notable features, which

- maximally absorbs a TBox that results from a knowledge base as shown in Section 3, and
- retains the guarding constraints as much as possible by prioritizing binary absorptions, and
- allows a DL reasoner to reason with restricted uses of nominals without introducing extra computational overhead, and
- makes it possible to absorb domain and range constraints in such a way that guards for domain and range axioms become unnecessary.

The procedure is given in Section 4.1, which also serves as a general framework for absorption. Its correctness proof follows in Section 4.2.

4.1 The procedure

The algorithm is given a $\mathcal{T}_{\mathcal{K}}$ that consists of arbitrary axioms. It proceeds by constructing five TBoxes $\mathcal{T}_g, \mathcal{T}_{prim}, \mathcal{T}_{uinc},$, $\mathcal{T}_{binc}$, and $\mathcal{T}_{rinc}$ such that: $\mathcal{T} \equiv \mathcal{T}_g \cup \mathcal{T}_{prim} \cup \mathcal{T}_{uinc} \cup \mathcal{T}_{binc} \cup \mathcal{T}_{rinc}$, $\mathcal{T}_{prim}$ is primitive, $\mathcal{T}_{uinc}$ consists of axioms of the form $A_1 \sqsubseteq C$, $\mathcal{T}_{binc}$ consists of axioms of the form $(A_1 \sqcap A_2) \sqsubseteq C$ and $(\{a\} \sqcap A) \sqsubseteq C$ and none of the above primitive concept are defined in $\mathcal{T}_{prim}$, and $\mathcal{T}_{rinc}$ consists of axioms of the form $\exists R.\top \sqsubseteq C$ (or $\exists R^-.\top \sqsubseteq C$). Here, $\mathcal{T}_{uinc}$ contains unary inclusion dependencies, $\mathcal{T}_{binc}$ contains binary inclusion dependencies and $\mathcal{T}_{rinc}$ contains domain and range inclusion dependencies.

In the first phase, we move as many axioms as possible from $\mathcal{T}$ into $\mathcal{T}_{prim}$. We initialize $\mathcal{T}_{prim} = \emptyset$ and process each axiom $X \in \mathcal{T}$ as follows.

1. If X is of the form $A \doteq C$, A is not defined in $\mathcal{T}_{prim}$, and $\mathcal{T}_{prim} \cup \{X\}$ is primitive, then move X to $\mathcal{T}_{prim}$.
2. If X is of the form $A \doteq C$, then remove X from $\mathcal{T}$ and replace it with axioms $A \sqsubseteq C$ and $\neg A \sqsubseteq \neg C$.
3. Otherwise, leave X in $\mathcal{T}$.

In the second phase, we process axioms in $\mathcal{T}$, either by simplifying them or by placing absorbed components in $\mathcal{T}_{uinc}$, $\mathcal{T}_{binc}$ or $\mathcal{T}_{rinc}$. We place components that cannot be absorbed in $\mathcal{T}_g$. We let $\mathbf{G} = \{C_1, \ldots, C_n\}$ represent the axiom $\top \sqsubseteq (C_1 \sqcup \ldots \sqcup C_n)$. Axioms are automatically converted to (out of) set notation. In addition, $\forall R.C$ (resp. $\forall R^-.C$) is considered a shorthand for $\exists^{\leq 0} R.\neg C$ (resp. $\exists^{\leq 0} R^-.\neg C$).

1. If $\mathcal{T}$ is empty, then return the binary absorption

$$(\{A \sqsubseteq C, \neg A \sqsubseteq \neg C \mid A \doteq C \in \mathcal{T}_{prim}\} \cup \mathcal{T}_{uinc} \cup \mathcal{T}_{binc} \cup \mathcal{T}_{rinc}, \mathcal{T}_g).$$

   Otherwise, remove an axiom $\mathbf{G}$ from $\mathcal{T}$.

2. Simplify $\mathbf{G}$.
   (a) If there is some $\neg C \in \mathbf{G}$ such that C is not a primitive concept, then add $(\mathbf{G} \cup \mathrm{NNF}(\neg C) \setminus \{\neg C\}$ to $\mathcal{T}$, where the function $\mathrm{NNF}(\cdot)$ converts concepts to negation normal form. Return to Step 1.
   (b) If there is some $C \in \mathbf{G}$ such that C is of the form $(C_1 \sqcap C_2)$, then add both $(\mathbf{G} \cup \{C_1\}) \setminus \{C\}$ and $(\mathbf{G} \cup \{C_2\}) \setminus \{C\}$ to $\mathcal{T}$. Return to Step 1.
   (c) If there is some $C \in \mathbf{G}$ such that C is of the form $C_1 \sqcup C_2$, then apply associativity by adding $(\mathbf{G} \cup \{C_1, C_2\}) \setminus \{C_1 \sqcup C_2\}$ to $\mathcal{T}$. Return to Step 1.
3. Partially absorb $\mathbf{G}$.
   (a) If $\{\neg\{a\}, \neg A\} \subset \mathbf{G}$, and A is a guard, then do the following. If an axiom of the form $(\{a\} \sqcap A) \sqsubseteq A'$ is in $\mathcal{T}_{binc}$, add $\mathbf{G} \cup \{\neg A'\} \setminus \{\neg\{a\}, \neg A\}$ to $\mathcal{T}$. Otherwise, introduce a new concept $A' \in \mathrm{NC}$, add $(\mathbf{G} \cup \{\neg A'\}) \setminus \{\neg\{a\}, \neg A\}$ to $\mathcal{T}$, and $(\{a\} \sqcap A) \sqsubseteq A'$ to $\mathcal{T}_{binc}$. Return to Step 1.
   (b) If $\{\neg A_1, \neg A_2\} \subset \mathbf{G}$, $(A_1 \sqcap A_2) \sqsubseteq A' \in \mathcal{T}_{binc}$, then add $\mathbf{G} \cup \{\neg A'\} \setminus \{\neg A_1, \neg A_2\}$ to $\mathcal{T}$. Return to Step 1.
   (c) If $\{\neg A_1, \neg A_2\} \subset \mathbf{G}$, and neither A_1 nor A_2 are defined in $\mathcal{T}_{prim}$, then do the following. If an axiom of the form $(A_1 \sqcap A_2) \sqsubseteq A'$ is in $\mathcal{T}_{binc}$, add $\mathbf{G} \cup \{\neg A'\} \setminus \{\neg A_1, \neg A_2\}$ to $\mathcal{T}$. Otherwise, introduce a new concept $A' \in \mathrm{NC}$, add $(\mathbf{G} \cup \{\neg A'\}) \setminus \{\neg A_1, \neg A_2\}$ to $\mathcal{T}$, and $(A_1 \sqcap A_2) \sqsubseteq A'$ to $\mathcal{T}_{binc}$. Return to Step 1.
   (d) If $\{\forall R.C\} = \mathbf{G}$ (resp. $\{\forall R^-.C\} = \mathbf{G}$), then do the following. Add $\exists R^-.\top \sqsubseteq C$ (resp. $\exists R.\top \sqsubseteq C$) to $\mathcal{T}_{rinc}$. Return to Step 1.
   (e) If $\forall R.\neg A$ (resp. $\forall R^-.\neg A) \in \mathbf{G}$, then do the following. Introduce a new internal primitive concept A' and add both $A \sqsubseteq \forall R^-.A'$ (resp. $A \sqsubseteq \forall R.A'$) and $(\mathbf{G} \cup \{\neg A'\}) \setminus \{\forall R.\neg A\}$ (resp. $\setminus \{\forall R^-.\neg A\}$) to $\mathcal{T}$. Return to Step 1.
4. Unfold $\mathbf{G}$. If, for some $A \in \mathbf{G}$ (resp. $\neg A \in \mathbf{G}$), there is an axiom $A \doteq C$ in $\mathcal{T}_{prim}$, then substitute $A \in \mathbf{G}$ (resp. $\neg A \in \mathbf{G}$) with C (resp. $\neg C$), and add $\mathbf{G}$ to $\mathcal{T}$. Return to Step 1.
5. Absorb $\mathbf{G}$. If $\neg A \in \mathbf{G}$ and A is not defined in $\mathcal{T}_{prim}$, add $A \sqsubseteq C$ to $\mathcal{T}_{uinc}$ where C is the disjunction of $\mathbf{G} \setminus \{\neg A\}$. Return to Step 1.
6. If none of the above are possible ($\mathbf{G}$ cannot be absorbed), add $\mathbf{G}$ to $\mathcal{T}_g$. Return to Step 1. □

In the above procedure, Step 3a is prioritized to ensure the pairing of a nominal concept and a guarding concept for the purpose of guarded reasoning, which in addition guarantees that nominals never occur on the right hand side of an axiom. Step 3b is performed before Step 3c to reduce nondeterminism of binary absorption and to minimize the number of fresh concepts to be introduced. In practice other heuristics may be applied for such purposes, for instance, a strict ordering can be imposed on all concept names such that binary absorption absorbs axioms in specific ways.

4.2 Correctness of the procedure

Termination of our procedure can be established by a counting argument. We now prove the correctness of our algorithm using induction. Lemmas 7 and 8 prove, in combination, that Steps 3a, 3b and 3c of our algorithm are correct. Lemmas 9 and 10 prove Step 3d correct and Lemmas 11 and 12 prove Step 3e correct, respectively.

Lemma 7. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, and A a primitive concept not used in C or $\mathcal{T}$. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{(C_1 \sqcap C_2 \sqcap C_3) \sqsubseteq C_4\}, \tag{1}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to

$$\mathcal{T}_2 = \mathcal{T} \cup \{(C_1 \sqcap C_2) \sqsubseteq A, (A \sqcap C_3) \sqsubseteq C_4\}. \tag{2}$$

Proof. First we prove the only-if direction. Assume C is satisfiable with respect to $\mathcal{T}_1$. For each interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}_1$ and $C^{\mathcal{I}} \neq \emptyset$, we extend to an interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$. First, set $\mathcal{I}' = \mathcal{I}$. For each $x \in \Delta^{\mathcal{I}}$ such that $x \in C_1^{\mathcal{I}}$ and $x \in C_2^{\mathcal{I}}$, add x to $A^{\mathcal{I}'}$. Then, $\mathcal{I}' \models \mathcal{T}_2$.

For the if direction, assume C is satisfiable with respect to $\mathcal{T}_2$. For each interpretation $\mathcal{I} \in \mathrm{Int}(L)$ such that $\mathcal{I} \models \mathcal{T}_2$ and $C^{\mathcal{I}} \neq \emptyset$, we show that $\mathcal{I} \models \mathcal{T}_1$. The proof is by contradiction. Assume $\mathcal{I} \not\models \mathcal{T}_1$. It must be the case that $(C_1 \sqcap C_2 \sqcap C_3) \sqsubseteq C_4 \in \mathcal{T}_1$ does not hold, since the rest of $\mathcal{T}_1$ is a subset of $\mathcal{T}_2$. Therefore, there exists $x \in \Delta^{\mathcal{I}}$ such that $x \in C_1^{\mathcal{I}} \cap C_2^{\mathcal{I}} \cap C_3^{\mathcal{I}}$, and $x \notin C_4^{\mathcal{I}}$. But, in this case either $(C_1 \sqcap C_2) \sqsubseteq A \in \mathcal{T}_2$ or $(A \sqcap C_3) \sqsubseteq C_4 \in \mathcal{T}_2$ must not hold. A contradiction.

Lemma 8 proves that instead of introducing a new primitive concept every time we execute Steps 3a, 3b and 3c of our algorithm, we may instead reuse a previously introduced primitive concept. We use $\mathbf{H}$ to denote an arbitrary axiom.

Lemma 8. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, A a primitive concept not used in C or $\mathcal{T}$, and A_1, A_2, primitive concepts introduced by Steps 3a, 3b and 3c of our algorithm modified such that a new primitive is always introduced. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{(C_1 \sqcap C_2) \sqsubseteq A_1, (C_1 \sqcap C_2) \sqsubseteq A_2\}, \tag{3}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to

$$\begin{aligned}\mathcal{T}_2 = &\{\mathbf{H} \ \ \textit{where } A \textit{ is substituted for } A_1 \textit{ and } A_2 \ \mid \mathbf{H} \in \mathcal{T}\}\\ &\cup \{(C_1 \sqcap C_2) \sqsubseteq A\}\end{aligned} \tag{4}$$

Proof. First we prove the only-if direction. We have two cases.

- Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \mathcal{T}_1$, $C^{\mathcal{I}} \neq \emptyset$, and $A_1^{\mathcal{I}} = A_2^{\mathcal{I}}$. We construct an interpretation $\mathcal{I}'$ from $\mathcal{I}$ such that $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$. First, set $\mathcal{I}' = \mathcal{I}$. Then, set $A^{\mathcal{I}'} = A_1^{\mathcal{I}}$ and remove any references to A_1 and A_2 in $\mathcal{I}'$.

– Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \mathcal{T}_1$, $C^{\mathcal{I}} \neq \emptyset$, and $A_1^{\mathcal{I}} \neq A_2^{\mathcal{I}}$. We construct an interpretation $\mathcal{I}'$ from $\mathcal{I}$ such that $\mathcal{I}' \models \mathcal{T}_1$, $C^{\mathcal{I}'} \neq \emptyset$, and $A_1^{\mathcal{I}} = A_2^{\mathcal{I}}$. For $x \in \Delta^{\mathcal{I}}$ such that $x \in A_1^{\mathcal{I}} \cup A_2^{\mathcal{I}}$ and $x \notin A_1^{\mathcal{I}} \cap A_2^{\mathcal{I}}$, we show that we can remove x from either $A_1^{\mathcal{I}}$ or $A_2^{\mathcal{I}}$ so that $x \notin A_1^{\mathcal{I}} \cup A_2^{\mathcal{I}}$ without causing any axiom in $\mathcal{T}_1$ to fail to hold. Without loss of generality, assume $x \in A_1^{\mathcal{I}}$ and $x \notin A_2^{\mathcal{I}}$. If $x \in C_1$ and $x \in C_2$, then we have a contradiction. Otherwise, we remove x from $A_1^{\mathcal{I}}$. Since either $x \notin C_1^{\mathcal{I}}$ or $x \notin C_2^{\mathcal{I}}$, the axiom $(C_1 \sqcap C_2) \sqsubseteq A_1$ holds. No other axiom in $\mathcal{T}_1$ has A_1 on the right hand side, therefore removing x from $A_1^{\mathcal{I}}$ does not cause any other axiom to fail to hold. Since the above is true for all x such that x is in only one of $A_1^{\mathcal{I}}$ and $A_2^{\mathcal{I}}$, we may remove individuals from $A_1^{\mathcal{I}}$ and $A_2^{\mathcal{I}}$ until $A_1^{\mathcal{I}} = A_2^{\mathcal{I}}$. Then the first case applies.

Now we prove the if direction. Let $\mathcal{I}$ be an interpretation such that $\mathcal{I} \models \mathcal{T}_2$ and $C^{\mathcal{I}} \neq \emptyset$. We construct an interpretation $\mathcal{I}'$ from $\mathcal{I}$ such that $\mathcal{I}' \models \mathcal{T}_1$ and $C^{\mathcal{I}'} \neq \emptyset$. First set $\mathcal{I}' = \mathcal{I}$. Then, set $A_1^{\mathcal{I}'} = A_2^{\mathcal{I}'} = A^{\mathcal{I}}$. Due to the construction of $\mathcal{T}_2$, $\mathcal{I}' \models \mathcal{T}_1$ and $C^{\mathcal{I}'} \neq \emptyset$.

Lemma 9. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, and R a role. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{\top \sqsubseteq \forall R.C_1\}, \tag{5}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to TBox

$$\mathcal{T}_2 = \mathcal{T} \cup \{\exists R^-.\top \sqsubseteq C_1\}. \tag{6}$$

Proof. First we prove the only-if direction. Assume C is satisfiable with respect to $\mathcal{T}_1$. For an interpretation $\mathcal{I} \in \mathrm{Int}(L)$ such that $\mathcal{I} \models \mathcal{T}_1$ and $C^{\mathcal{I}} \neq \emptyset$, we extend $\mathcal{I}$ to an interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$. First set $\mathcal{I}' = \mathcal{I}$. For each $x \in \Delta^{\mathcal{I}}$, we add x to $(\forall R.C_1)^{\mathcal{I}'}$. Then, $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$.

Now we prove the if direction. Assume C is satisfiable with respect to $\mathcal{T}_2$. For each interpretation $\mathcal{I} \in \mathrm{Int}(L)$ such that $\mathcal{I} \models \mathcal{T}_2$ and $C^{\mathcal{I}} \neq \emptyset$, it is also the case that $\mathcal{I} \models \mathcal{T}_1$. The proof is by contradiction. Assume $\mathcal{I} \not\models \mathcal{T}_1$. It must be the case that axiom $\top \sqsubseteq \forall R.C_1$ does not hold. Then for some $x \in \Delta^{\mathcal{I}}$, there is $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in R^{\mathcal{I}}$ and $y \notin C_1^{\mathcal{I}}$. However, from axiom $\exists R^-.\top \sqsubseteq C_1$ and the fact that $(y, x) \in (R^-)^{\mathcal{I}}$, it follows that $y \in (\exists R^-.\top)^{\mathcal{I}}$, thus, $y \in C_1^{\mathcal{I}}$: a contradiction.

Lemma 10. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, R a role and R^- an inverse role of R. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{\top \sqsubseteq \forall R^-.C_1\}, \tag{7}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to TBox

$$\mathcal{T}_2 = \mathcal{T} \cup \{\exists R.\top \sqsubseteq C_1\}. \tag{8}$$

The proof of this lemma is similar to that of Lemma 9.

Lemma 11. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, A a primitive concept not used in C or $\mathcal{T}$, and R a role. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{\exists R.C_1 \sqsubseteq C_2\}, \tag{9}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to TBox

$$\mathcal{T}_2 = \mathcal{T} \cup \{C_1 \sqsubseteq \forall R^-.A, A \sqsubseteq C_2\}. \tag{10}$$

Proof. First we prove the only-if direction. Assume C is satisfiable with respect to $\mathcal{T}_1$. For an interpretation $\mathcal{I} \in \mathrm{Int}(L)$ such that $\mathcal{I} \models \mathcal{T}_1$ and $C^{\mathcal{I}} \neq \emptyset$, we extend $\mathcal{I}$ to an interpretation $\mathcal{I}'$ such that $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$. First set $\mathcal{I}' = \mathcal{I}$. For each $x \in \Delta^{\mathcal{I}}$ such that $x \in (\exists R.C_1)^{\mathcal{I}} \cap C_2^{\mathcal{I}}$, we add x to $A^{\mathcal{I}'}$. Then, $\mathcal{I}' \models \mathcal{T}_2$ and $C^{\mathcal{I}'} \neq \emptyset$.

Now we prove the if direction. Assume C is satisfiable with respect to $\mathcal{T}_2$. For each interpretation $\mathcal{I} \in \mathrm{Int}(L)$ such that $\mathcal{I} \models \mathcal{T}_2$ and $C^{\mathcal{I}} \neq \emptyset$, it is also the case that $\mathcal{I} \models \mathcal{T}_1$. The proof is by contradiction. Assume $\mathcal{I} \not\models \mathcal{T}_1$. It must be the case that axiom $\exists R.C_1 \sqsubseteq C_2$ does not hold as all other axioms in $\mathcal{T}_1$ are also in $\mathcal{T}_2$. Then there exists $x \in \Delta^{\mathcal{I}}$ such that $x \in (\exists R.C_1)^{\mathcal{I}}$ and $x \notin C_2^{\mathcal{I}}$. However, this implies that there exists $y \in \Delta^{\mathcal{I}}$ such that $(x, y) \in R^{\mathcal{I}}$ and $y \in C_1$. From axiom $C_1 \sqsubseteq \forall R^-.A$, it must be the case that $x \in A^{\mathcal{I}}$. From axiom $A \sqsubseteq C_2$, it must be the case that $x \in C_2^{\mathcal{I}}$. A contradiction.

Lemma 12. *Let $\mathcal{T}_1, \mathcal{T}_2$, and $\mathcal{T}$ denote TBoxes, $C \in L$ an arbitrary concept, A a primitive concept not used in C or $\mathcal{T}$, and R a role. If $\mathcal{T}_1$ is of the form*

$$\mathcal{T}_1 = \mathcal{T} \cup \{\exists R^-.C_1 \sqsubseteq C_2\}, \tag{11}$$

then C is satisfiable with respect to $\mathcal{T}_1$ iff C is satisfiable with respect to TBox

$$\mathcal{T}_2 = \mathcal{T} \cup \{C_1 \sqsubseteq \forall R.A, A \sqsubseteq C_2\}. \tag{12}$$

The proof of this lemma is similar to that of Lemma 11.

Theorem 2. *For any TBox $\mathcal{T}$, the ABox absorption algorithm computes a correct absorption of $\mathcal{T}$.*

Proof. The proof is by induction on iterations of our algorithm. We abbreviate the pair $(\{A \sqsubseteq C, \neg A \sqsubseteq \neg C \mid A \doteq C \in \mathcal{T}_{prim}\} \cup \mathcal{T}_{uinc} \cup \mathcal{T}_{binc} \cup \mathcal{T}_{rinc}, \mathcal{T}_g \cup \mathcal{T})$ as $\mathcal{T}$ and claim that this pair is always a correct binary absorption. Initially, $\mathcal{T}_{uinc}$, $\mathcal{T}_{binc}$, $\mathcal{T}_{rinc}$ and $\mathcal{T}_g$ are empty, primitive axioms are in $\mathcal{T}_{prim}$, and the remaining axioms are in $\mathcal{T}$. By Lemma 3, Lemma 4, Lemma 5, and Lemma 6, $\mathcal{T}$ is a correct absorption at the start of our algorithm. Assume we just finish iteration i and now perform iteration $i+1$. By our induction hypothesis, $\mathcal{T}$ is a correct binary absorption after iteration i. We have a number of possible cases below.

- If we perform Step 3a, Step 3b or Step 3c then iteration $i+1$ is finished. Therefore, a newly introduced primitive concept only appears on the right hand side of an axiom once and Lemma 7 and Lemma 8 apply. We conclude that $\mathcal{T}$ is a correct binary absorption.

– If we perform Step 3d, then iteration $i + 1$ is finished and by Lemma 9 and Lemma 10, $\mathcal{T}$ is a correct binary absorption.
– If we perform Step 3e, then iteration $i + 1$ is finished and by Lemma 11 and Lemma 12, $\mathcal{T}$ is a correct binary absorption.
– If we perform any of Steps 1, 2, 4, 5, or 6, then $\mathcal{T}$ is a correct binary absorption at the end of iteration $i + 1$. This is because these steps use only equivalence preserving operations.

After the final iteration of our algorithm, $\mathcal{T}$ is a correct binary absorption by induction.

5 Empirical Evaluation

The proposed absorption technique has been implemented in our CARE Assertion Retrieval Engine (CARE)[2] based on a DL reasoner. We elaborate the implementation details of CARE and the experimental setting below.

5.1 Implementation and Experimental Setup

The DL reasoner implements a standard tableaux algorithm [HS02] for $\mathcal{ALCI}(\mathcal{D})$ knowledge bases. The reasoner also features a limited number of optimizations, including absorption presented in Section 4.1, optimized double blocking in [HS02], and dependency-directed backtracking in [BCM$^+$03b]. Moreover, the reasoner is also capable of reasoning with restricted ("safe" use of) nominals, where nominals appear only in binary axioms introduced in conversion steps. At present, string is the only supported data type in the reasoner and a transitive closure algorithm [Nuu95] is used to find clashes among concrete concepts.

All times, given in seconds, were averaged out over five independent runs on a Ubuntu 12.04 Linux server. A time out of 2000 seconds was set for state-of-the-art DL reasoners, which were run on a single core of a 2.6GHz AMD Opteron 6282 SE processor, with the virtual memory for Java programs set to 4GB. Queries were evaluated in the latest releases of these DL reasoners. For instance, queries were posed via OWL API 3 for FaCT++ 1.6.0[3], Pellet 2.3.0[4] and HermiT 1.3.6[5], and via JRacer API for RacerPro 2.0[6] using the nRQL query language. Observe that these highly optimized reasoners often pre-compute information w.r.t. the loaded KBs for later query answering; therefore, the times given in this section are the sums of the query response time and the amortized, i.e., divided by the number of queries, preprocessing time (including KB loading time).

A suite of datasets (KBs) about digital cameras has been built and used in the experiments. The KBs consist of digital camera model specifications extracted from `DPreview.com` and pricing information from *Google Product Search*

[2] http://db-tom.cs.uwaterloo.ca
[3] http://code.google.com/p/factplusplus/
[4] http://clarkparsia.com/pellet/
[5] http://www.hermit-reasoner.com/
[6] http://www.racer-systems.com/

as of July 2012. The ABox (data) of each KB contains a set of camera models described by a substantial number (around 70) of concrete feature concepts, in addition to other concepts. Every camera model has n ($0 \leq n \leq 10$) products for sale through various sellers. The TBox (schema) has 34 axioms, which remains the same for all KBs. Table 1 expands on the size of the KBs, i.e., the number of camera models, individuals, concept assertions and role assertions, respectively. Observe that DPC1 has all the available digital cameras on `DPreview.com`, assuming there is only one product available for sale through each seller for a particular model. DPC2 correspondingly assumes there are two products for sale through each seller.

	CMs	Inds	CAs	RAs	Q1	Q2	Q3	Q4	Q5	Q6	Q7
HP	577	8774	4926	12101	0	530	2	37	3737	8	0
Oly	1198	18354	10161	25705	1	1067	4	110	7906	183	3
DPC1	1895	28017	15445	39453	11	1673	7	181	12095	183	3
DPC2	1895	40595	15445	64510	11	1673	7	181	24190	183	6

Table 1: Description of KBs

Queries are shown in Table 2, which were designed to vary in query forms and selectivity. The result size of each query over every KB is also given in Table 1. These instance queries abbreviate assertion retrieval queries of the form (C, Pd), where C is a query concept in each row of Table 1 and the *projection decription* Pd is always $\top$ (cf. [PTWW11] for details about assertion retrieval). Q1 is a selective query that retrieves a specific subclass of SLR cameras, which may be answered without assuming any hierarchy information. Q2, contrary to Q1, is less selective and must be answered using class hierarchy. Q3 is a selective query involving concrete facts. Q4 negates the concrete fact in Q3 to make it a range query. Q5, involving roles and negated concrete facts, is subsumed by a primitive concept ($Available\_Digital\_Camera$), which was used as the query concept instead. Q6 has a disjunction occurring in the scope of an existential restriction. Q7 consists of a conjunction, of which the first (second) conjunct is a one-level (two-level) existential restriction involving concrete facts.

5.2 Comparing Guarding Strategies

Definition 3 effectively develops two types of guarding strategies, one being *partial guarding* (PG) and the other *full guarding* (FG). Intuitively, the PG strategy guards all individuals so that only individuals *relevant* to the individuals appearing in a query will be explored by reasoners. The FG strategy, in addition to PG, further guards concrete features so that only query-relevant feature concepts participate in reasoning. A naïve tableau-based implementation, however, tends

Q1	$Digital\_SLR\_mirrorless$
Q2	$Compact\_Camera$
Q3	$Digital\_SLR \sqcap (user\_review = \text{“5.00”})$
Q4	$Digital\_SLR \sqcap (\neg(user\_review = \text{“5.00”}))$
Q5	$\exists hasSale.(\neg(inventory\_status = \text{“}outOfStock\text{”}))$
Q6	$\exists hasManu.((manu\_name = \text{“}Kodak\text{”}) \sqcup (\exists locatedIn.Europe\_Country)))$
Q7	$(\exists hasInstance^{-}.(Lens\_mount = \text{“}Nikon\_F\_mount\text{”})) \sqcap$ $(\exists hasSale.\exists hasSeller.(seller\_name = \text{“}Walmart\text{”}))$

Table 2: Sample queries

to explore all ABox individuals for each reasoning request, which is dubbed *no guarding* in our context.

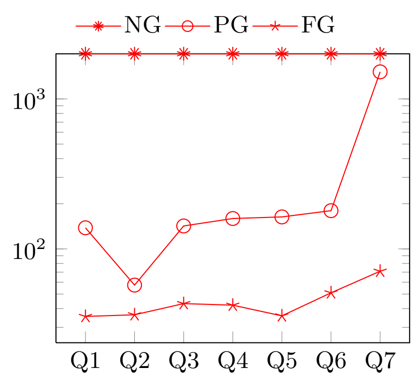
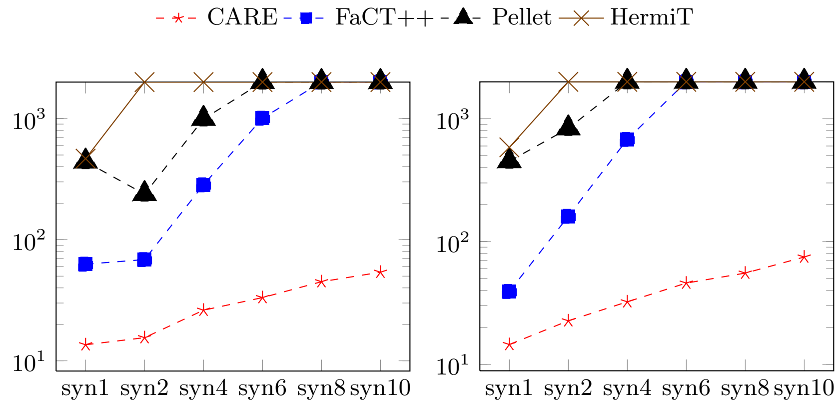


Fig. 2: Different Guarding Strategies (logarithmic scale)

Our first experiments evaluated all queries over the HP KB under different guarding strategies. Table 2 shows that CARE timed out under the NG strategy for all queries, because it has to explore the whole ABox for each $\mathcal{K} \models a : C$ (note that CARE implements a “linear” instance retrieval for a given query concept). By adopting the PG method, CARE managed to answer all queries. The query response times for almost all queries (except Q2) have been improved by one order of magnitude under the FG strategy. Thus, the comparison suggests the efficacy and the potential of the proposed guarding strategies. It should be noted that full guarding is extremely helpful in our experiment data sets due to the sheer number of features employed to describe camera models. The general performance gains of FG may not be as significant as in this example, nevertheless, we can expect the tendency of modelling feature-rich object in many applications.



(a) Query: B

(b) Query: $\exists R.B$

Fig. 3: Effectiveness of Partial Guarding (logarithmic scale)

It might be argued that the effectiveness of guarding may be limited for highly optimized reasoners. We, thus, performed another series of experiments, which, as plotted in Figure 3, demonstrated that partial guarding (PG) alone can result in significant performance gains for state-of-the-art reasoners. We synthesized a seed KB that has two axioms and a number of ABox assertions replicated from the following assertions:

$$
\begin{array}{lll}
(\exists R^{-}.\top \sqcap \exists S_1.\top) \sqsubseteq B & & E \sqsubseteq (B \sqcup C) \\
R(a_m, b_i) & S_1(a_m, c_i) & \\
S_1(b_m, e_i) & S_2(c_m, e_j) & \\
R^{-}(d_m, c_k) & S_1(e_m, d_i) & \\
C(c_m) & E(e_m) &
\end{array}
$$

. Initially, $m = 1000$ in the seed KB (syn1), which generates $5k$ instances, $6k$ role assertions and $2k$ concept assertions, as the second individual in a role assertion is randomly selected, e.g., $1 \leq i \leq m$. All other synthetical KBs were generated by repeat the seed KB k times, thus, named syn$k$. Because these KBs contain no concrete facts, partial guarding (PG) was adopted during reasoning. The results in Figure 3 are remarkable in that these highly optimized DL reasoners were, in most cases, at least one order of magnitude slower in answering simple, positive instance queries, compared to CARE. It, therefore, implies that guarding is an essential optimization technique for query answering over DL knowledge bases, complementary to existing ones.

5.3 Comparing DL Reasoners

As numerous optimization techniques have been developed in state-of-the-art DL reasoners, we juxtaposed them with CARE for answering queries over realistic

KBs about digital cameras. The purpose here is not to compare reasoners, but to validate the efficacy of our proposed guarding optimization for instance retrieval, because CARE is a research prototype with limited optimizations. Figure 4 depicts the performance of the five reasoners over four KBs with increasing complication.

There are several interesting observations regarding the results. All reasoners computes certain useful information prior to the actual query answering, either during the KB loading phase or after receiving explicit request. For instance, Racer spent a significant amount of time in building indices when instructed, which made answering the first two queries extremely fast, e.g., the actual query response time is far less than the illustrated time in Figure 4. Racer, however, was even unable to finish preprocessing for KB DPC2 within the specified timeout interval. The preprocessing phase in CARE is devoted to converting KBs, and the runtime cost is, as we have observed, one fourth of that in other reasoners.

When the queries are positive, i.e., neither negation nor disjunction occur in them, almost all reasoners demonstrated satisfactory performance, e.g., for Q1-3 and Q7. In answering these queries, CARE was much faster than all others except Pellet. However, when the queries turned to be non positive, e.g., Q4-6, the runtime performance of Pellet, HermiT and Racer has been adversely affected. Under these circumstances, CARE outperformed all other reasoners in large KBs. FaCT seemed to perform consistently well over all queries, yet CARE was at least two-fold faster in most cases, except that FaCT had slightly better runtime performance than CARE for Q1. It can be observed from the experiment results that CARE, while adopting the guarding optimization, enjoys a consistent, superior runtime performance over different knowledge bases and object queries. Given that CARE is not as optimized as any of the other reasoners, the results are indeed significant.

6 Conclusions and Future Work

We have shown how, given a consistent knowledge base, one can avoid reasoning with irrelevant ABox individuals for an instance query, while preserving correctness of answers. This goal is achieved by *instrumenting* the original ABox with additional *guards* that are represented by auxiliary primitive concepts, and then by developing an extension to absorption theory and algorithms in [HT00a,HT00b]. This extension, called *binary absorption*, originally designed for TBox reasoning alone [HW06], allows terminological axioms of the form $(\{a\} \sqcap A) \sqsubseteq C$ to qualify for lazy unfolding in model building satisfaction procedures for description logics, such as those based on tableaux technology. Such lazily unfolded axioms with *binary left-hand sides* are *essential* when (translations of) ABox assertions are to be processed by such algorithms since they prevent exploring concepts and roles associated with irrelevant ABox individuals (indeed, a simple modification to said tableaux algorithms will avoid creating instances of such individuals altogether.) Such an optimization cannot be achieved when only unary absorption is available.

The nature of guarded reasoning resembles the idea of model merging (and its predecessor pre-completion) in that both use certain mechanism to reflect the interaction among individuals. The difference is that the latter exploits pseudo models for determining answers to queries $\mathcal{K} \not\models a : C$, i.e., showing "obvious no-insatnces"[HM08]. If it fails to exclude a from the answers to C, then it resorts to standard ABox consistency checking. Note that due to its relying on pre-completion, the technique is sound but incomplete. Nevertheless, guarded reasoning *fully* reflects the possible interaction of instances and is both sound and complete for instance retrieval.

Our experiments show that in realistic situations arising, e.g., in implementations of *assertion retrieval* [PTWW11] in which a number of instance checking queries are needed to answer a single user query, or in the case of *ontology-based query answering* [LTW09,KLT+10,RA10,KLT+11], when non-Horn DLs are used (and thus the above techniques cannot be applied), our technique makes querying often feasible. The experiments show, on relatively simple examples, that, while using the proposed technique allows answers to be computed in a few seconds, attempting the same tasks without the optimization is infeasible. To be effective, the technique relies on absorption procedures that have at least the capabilities of binary absorption. An interesting avenue of further work would be to explore how highly optimized DL reasoning procedures with more powerful capabilities for absorption such as procedures based on hypertableau [MSH09] could further improve performance.

References

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[BCM+03a] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *Implementation and Optimisation Techniques*, chapter 9, pages 306–346. Cambridge Univserity Press, 2003.

[BCM+03b] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The description logic handbook: theory, implementation, and applications.* Cambridge University Press, New York, NY, USA, 2003.

[BFH+94] F. Baader, E. Franconi, B. Hollunder, B. Nebel, and H.-J. Profitlich. An empirical analysis of optimization techniques for terminological representation systems, or: Making KRIS get a move on. *Applied Artificial Intelligence*, 4:109–132, 1994.

[CDGL+09] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Antonella Poggi, Mariano Rodriguez-Muro, and Riccardo Rosati. Ontologies and databases: The DL-Lite approach. In Sergio Tessaris, Enrico Franconi, Thomas Eiter, Claudio Gutierrez, Siegfried Handschuh, Marie-Christine Rousset, and Renate Schmidt, editors, *Reasoning Web. Semantic Technologies for Information Systems*, volume 5689 of *Lecture Notes in Computer Science*, pages 255–356. Springer Berlin / Heidelberg, 2009.

[DLNS94] Francesco M. Donini, Maurizio Lenzerini, Daniele Nardi, and Andrea Schaerf. Deduction in concept languages: From subsumption to instance checking. *Journal of Logic and Computation*, 4(4):423–452, 1994.

[HM08] Volker Haarslev and Ralf Möller. On the scalability of description logic instance retrieval. *J. Autom. Reason.*, 41(2):99–142, August 2008.

[Hor97] Ian Horrocks. *Optimising Tableaux Decision Procedures For Description Logics*. PhD thesis, the University of Manchester, 1997.

[Hor98] I. Horrocks. Using an Expressive Description Logic: FaCT or Fiction? In *KR'98*, pages 636–647, 1998.

[HS02] Ian Horrocks and Ulrike Sattler. Optimised reasoning for shiq. In *ECAI'02*, pages 277–281, 2002.

[HT00a] I. Horrocks and S. Tobies. Optimisation of terminological reasoning. In *Description Logics'00*, pages 183–192, 2000.

[HT00b] I. Horrocks and S. Tobies. Reasoning with axioms: Theory and practice. In *KR'00*, pages 285–296, 2000.

[HW06] Alexander K. Hudek and Grant E. Weddell. Binary absorption in tableaux-based reasoning for description logics. In *Description Logics'06*, 2006.

[KLT+10] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to query answering in DL-Lite. In *KR'10*, 2010.

[KLT+11] Roman Kontchakov, Carsten Lutz, David Toman, Frank Wolter, and Michael Zakharyaschev. The combined approach to ontology-based data access. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 2656–2661, 2011.

[LTW09] Carsten Lutz, David Toman, and Frank Wolter. Conjunctive query answering in the description logic EL using a relational database system. In *IJCAI'09*, pages 2070–2075, 2009.

[MSH09] Boris Motik, Rob Shearer, and Ian Horrocks. Hypertableau reasoning for description logics. *J. Artif. Intell. Res. (JAIR)*, 36:165–228, 2009.

[Nuu95] Esko Nuutila. Efficient transitive closure computation in large digraphs. *Acta Polytechnica Scandinavia: Math. Comput. Eng.*, 74:1–124, July 1995.

[PTWW11] Jeffrey Pound, David Toman, Grant E. Weddell, and Jiewen Wu. An assertion retrieval algebra for object queries over knowledge bases. In *IJCAI'11*, pages 1051–1056, 2011.

[RA10] Riccardo Rosati and Alessandro Almatelli. Improving query answering over DL-Lite ontologies. In *KR'10*, 2010.

[TH04] D. Tsarkov and I. Horrocks. Efficient reasoning with range and domain constraints. In *Description Logics'04*, 2004.

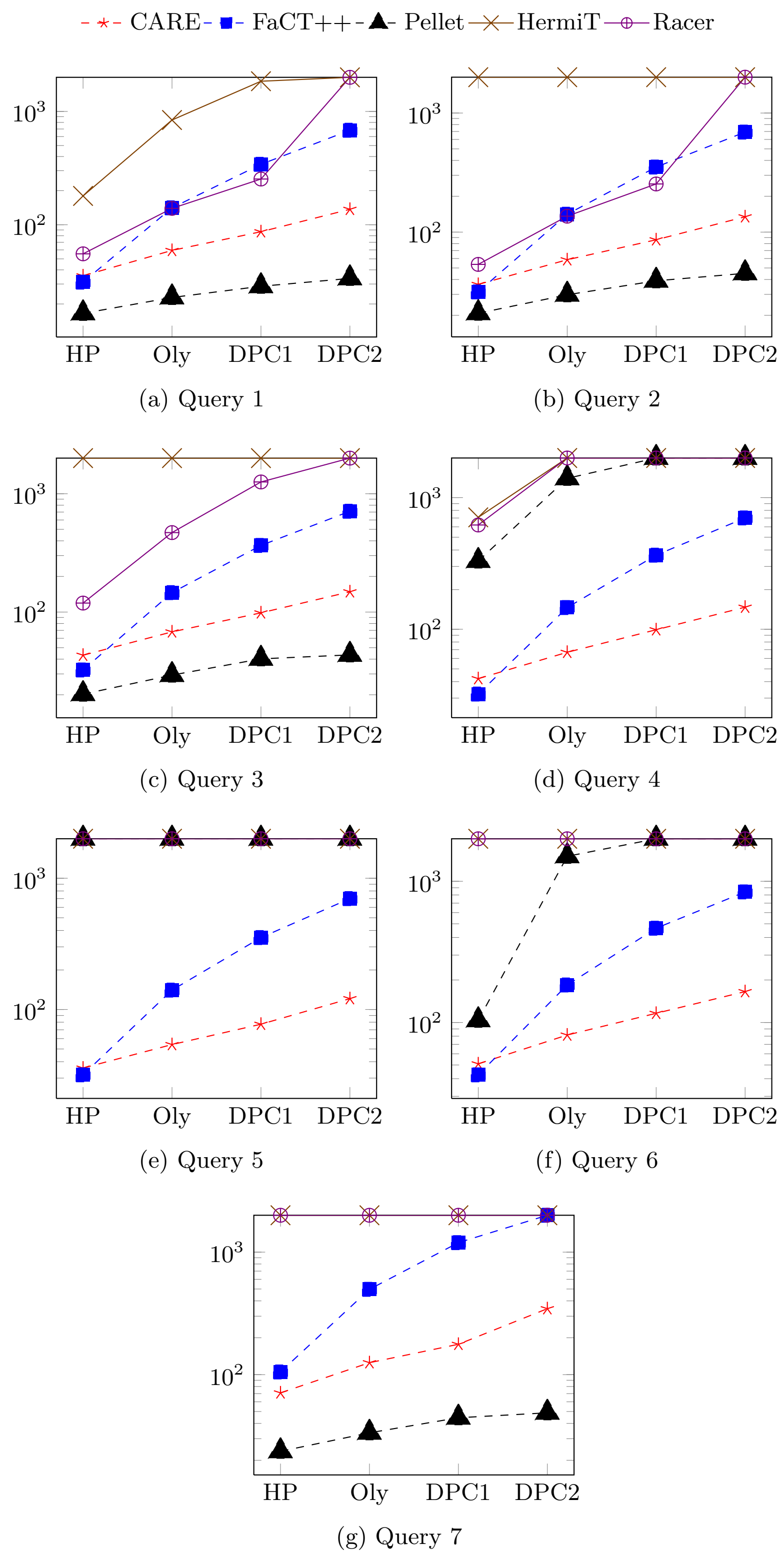


Fig. 4: Runtime Performance Comparison(logarithmic scale)