

# Absorption for ABoxes

Jiewen Wu, Alexander Hudek, David Toman, and Grant Weddell

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

**Abstract.** We present a novel method of evaluating instance queries over description logic knowledge bases that derives from binary absorption. The method is designed to work well for large ABoxes and where the TBox is not necessarily Horn, e.g., where background knowledge requires the use of disjunction and negation. The method significantly improves the performance of instance checking, and particularly so in cases where a large number of concrete feature values are included. We also report on the results of a preliminary experimental evaluation that validates the efficacy of the method.

## 1 Introduction

Two of the basic reasoning tasks over a DL knowledge base  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  are to determine if  $\mathcal{K}$  is consistent, and so-called *instance checking*: to determine if a given concept assertion  $a : C$  (stating that individual  $a$  occurring in  $\mathcal{A}$  belongs to concept  $C$ ) is a logical consequence of a *consistent*  $\mathcal{K}$ , written  $\mathcal{K} \models a : C$ .

Usually, these tasks are combined in the sense that the latter is assumed to include the former. However, we believe that typical workloads for a reasoning service will include far more instance checking tasks than knowledge base consistency tasks. The resulting “separation of concerns” can therefore enable technology that is far more efficient for such workloads, particularly so in the case of non-Horn DL TBoxes  $\mathcal{T}$  that preclude the possibility of computing so-called canonical ABoxes  $\mathcal{A}'$  from  $\mathcal{A}$  (e.g., when disjunction is used in  $\mathcal{T}$ ). 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 workloads that contain many thousands of instance checking tasks.

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 [1]. 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 [3].

To consider performance issues for instance checking in the context of absorption, 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. In particular, we present an absorption generation procedure that is an adaptation of an earlier procedure reported at the description

logics workshop [6]. This earlier procedure was called *binary absorption* and was itself a generalization of the absorption theory and algorithms developed by Horrocks and Tobies [4, 5]. 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 [13]. 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 inclusion dependency*.) 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 role and concrete feature guards in the context of a knowledge base for the DL dialect  $\mathcal{ALCIQ}(\mathbf{D})$ . In particular, we show how instance checking tasks 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 inclusion dependency.
3. We report on the results of an experimental evaluation that validates the efficacy of the proposed optimization.

After some preliminary definitions, these contributions are the subject of successive sections, and are followed in turn by our summary comments. Finally, note that a short earlier version of this paper is being presented simultaneously in a poster session of KR 2012 [14].

## 2 Preliminaries

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

**Definition 1 (Description Logic  $\mathcal{ALC}\mathcal{I}\mathcal{O}\mathcal{Q}(\mathbf{D})$ ).**

$\mathcal{ALC}\mathcal{I}\mathcal{O}\mathcal{Q}(\mathbf{D})$  is a DL dialect based on disjoint infinite sets of atomic concepts NC, atomic roles NR, concrete features NF and nominals NI. Also, if  $A \in \text{NC}$ ,  $R \in \text{NR}$ ,  $a \in \text{NI}$ ,  $f, g \in \text{NF}$ ,  $n$  is a non-negative integer and  $C_1$  and  $C_2$  are concept descriptions, then  $A$ ,  $\neg C_1$ ,  $C_1 \sqcap C_2$ ,  $C_1 \sqcup C_2$ ,  $\top$ ,  $\perp$ ,  $\exists R.C_1$ ,  $\forall R.C_1$ ,  $\exists R^-.C_1$ ,  $\forall R^-.C_1$ ,  $\{a\}$ ,  $\exists^{\leq n} R.C_1$ ,  $\exists^{\geq n} R.C_1$ ,  $\exists^{\leq n} R^-.C_1$ ,  $\exists^{\geq n} R^-.C_1$ ,  $f < g$  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 compound descriptions in the standard way.

Concrete domain concepts such as  $f < k$  are considered to be a shorthand for  $(f < g) \sqcap (g = k)$ , which, together with  $f = k$ , may be generalized as  $(t_1 \text{ op } t_2)$ , where  $t_1$  and  $t_2$  refer to either a concrete feature or a finite string, and  $\text{op} \in \{<, =\}$ .

**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  $A \in \text{NC}$ , each  $A \in \text{NC}$  appears in at most one left hand side of an axiom, and  $\mathcal{T}$  is acyclic.  $A \in \text{NC}$  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  $a : A$ ,  $a : (f \text{ op } k)$  and  $R(a, b)$ .

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

## 3 On Absorbing an ABox

The absorption proceeds in two steps: first *guards* that allow us to prune the exploration of the ABox during reasoning are added to the ABox assertions (in turn converted into TBox axioms about nominals) and then the resulting TBox is processed by an extended *binary absorption* algorithm.

### 3.1 Mapping Instance Checking to Subsumption Testing

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  $\mathcal{ALCIQ}(\mathbf{D})$ ):

**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  $\mathcal{K}$ :*

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

Note that all the axioms resulting from ABox assertions are *guarded* by auxiliary primitive concepts of the form  $\text{Def}_a$ ,  $\text{Def}_R$ , and  $\text{Def}_f$ . Intuitively, these concepts, when coupled with an appropriate absorption allow a reasoner to *ignore* parts of the original ABox: all the constants for which  $\text{Def}_a$  is not *set*, yielding considerable performance gains. For this idea to work we need to require (without loss of generality) that the TBox of  $\mathcal{K}$  only uses qualified *at-most* number restrictions of the form  $A \sqsubseteq \exists^{\leq n} R.B$  where  $A$  and  $B$  are atomic concepts or their negations. Note that subsumptions of the form  $\exists^{\geq n} R.A \sqsubseteq B$  are also considered to be at-most number restrictions and have to be equivalently rewritten in the above form and that nested 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 restriction by introducing new auxiliary concept names. Then we add the following assertions that manipulate the guards:

**Definition 4 (TBox Conversion).** *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:*

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

In the following we use  $\mathcal{T}_{\mathcal{K}}$  for  $\mathcal{T} \cup \mathcal{T}_{\mathcal{T}} \cup \mathcal{T}_{\mathcal{A}}$ .

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

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

where  $D = \text{Def}_a \sqcap (\prod_{f \text{ appears in } C} \text{Def}_f)$ .

*Proof.* Assume that there is an interpretation  $I_0$  that satisfies  $\mathcal{T}_{\mathcal{K}}$  such that  $(\{a\})^{I_0} \subseteq (D)^{I_0}$  but  $(\{a\})^{I_0} \cap (C)^{I_0} = \emptyset$  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). We construct an interpretation  $J$  for  $\mathcal{K} \cup \{a : \neg C\}$  as follows: Let  $\Gamma^{I_0}$  be the set of objects  $o \in \Delta^{I_0}$  such that either  $o \in (\{a\})^{I_0}$  and  $(\{a\})^{I_0} \subseteq (\text{Def}_a)^{I_0}$  or  $o$  is an anonymous object in  $\Delta^{I_0}$  rooted by such an object. Similarly let  $\Gamma^{I_1}$  be the set of objects  $o \in \Delta^{I_1}$  such that either  $o \in (\{a\})^{I_1}$  and  $(\{a\})^{I_1} \cap (\text{Def}_a)^{I_1} = \emptyset$  or  $o$  is an anonymous object in  $\Delta^{I_1}$  rooted by such an object. We set

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  $(t_1 \text{ op } t_2)$ );
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)  $o_1 \in (\{a\})^{I_0} \cap (\text{Def}_a)^{I_0}$ ,  $o_2 \in (\{b\})^{I_1}$  and  $R(a, b) \in \mathcal{A}$  (or vice versa).

We claim that  $(\{a\})^J \cap (C)^J = \emptyset$  (trivially) and  $J \models \mathcal{K}$ : to show the latter part we only need to consider those  $R$  edges of the form covered by the last case (4b): the edges that *cross* between the two interpretations, i.e., when  $o_1 \in (\{a\})^{I_0}$ ,  $o_2 \in (\{b\})^{I_1}$  and  $R(a, b) \in \mathcal{A}$ . Now consider an inclusion dependency expressing an *at-most* restriction  $A \sqsubseteq \exists^{\leq n} R.B \in \mathcal{T}$ . We can conclude that  $o_1 \notin (A)^{I_0}$  as otherwise  $o_1 \in (\text{Def}_R)^{I_0}$  by Definition 4 and thus  $o_2 \in (\text{Def}_B)^{I_0}$  by Definition 3 which contradicts our assumption that  $(\{b\})^{I_0} \cap (\text{Def}_B)^{I_0} = \emptyset$ . Hence the inclusion dependency is satisfied vacuously. The remaining edges, case (4a), satisfy all dependencies 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 dependencies in  $\mathcal{K}$  are satisfied by  $J$ .

The other direction of the proof follows by observing that if  $\mathcal{K} \cup \{a : \neg C\}$  is satisfiable then the satisfying interpretation  $I$  can be extended to  $(\text{Def}_a)^I = (\text{Def}_f)^I = (\text{Def}_R)^I = \Delta^I$  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  $(\{a\})^I \subseteq (D)^I \cap (\neg C)^I$ .  $\square$

Also, in  $\mathcal{ALCI}(\mathbf{D})$  we do not need to rely explicitly on the unique name assumption (UNA: the logic on its own cannot equate constants). However, we could allow explicit equalities and inequalities to the ABox and then preprocess them similarly to Definition 3, e.g.,  $a \approx b$  to  $\{a\} \sqcap \text{Def}_a \sqsubseteq \{b\} \sqcap \text{Def}_b$  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  $\text{Def}_a$  is not set in  $I_0$  are irrelevant for constructing the interpretation  $J$  even though there could be assertions of the form  $\top \sqsubseteq C$  that are applicable to such objects (one could even augment all such assertions by adding guards to avoid this effect). Therefore those constants (nominals) can be ignored completely by the reasoner and thus nodes corresponding to the constant symbols can be generated *lazily* on demand driven by the  $\text{Def}_a$  concept.

### 3.2 On Witnesses and Binary Absorption

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

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

Fig. 1: Absorption witness conditions

the idea of a partial description. This has been done in [5] 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  $\mathcal{ALC}\mathcal{IO}\mathcal{Q}(\mathbf{D})$  concept.<sup>1</sup> 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  $\text{NR}$  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  $\mathcal{ALC}\mathcal{IO}\mathcal{Q}(\mathbf{D})$  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  $\Delta^{\mathcal{I}} = \Delta^{\mathcal{W}}$ ,  $\cdot^{\mathcal{I}}|_{\text{NR}} = \cdot^{\mathcal{W}}$ , for each  $A \in \text{NC}$ ,  $A \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \in A^{\mathcal{I}}$  and  $\neg A \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \notin A^{\mathcal{I}}$ , for each  $a \in \text{NI}$ ,  $\{a\} \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \in \{a\}^{\mathcal{I}}$  and  $\neg\{a\} \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \notin \{a\}^{\mathcal{I}}$ , for each  $(f \text{ op } k)$ ,  $(f \text{ op } k) \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \in (f \text{ op } k)^{\mathcal{I}}$  and  $\neg(f \text{ op } k) \in \mathcal{L}^{\mathcal{W}}(x)$  implies  $x \notin (f \text{ op } k)^{\mathcal{I}}$ .

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 have been captured by the original lemmas 2.6 and 2.7 in [5]. We further extend binary absorption [6] to accommodate the absorbed ABox as shown in Section 3.

**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.  $\exists R^{\neg}.\top \sqsubseteq C$ ), and the form  $(A_1 \sqcap A_2) \sqsubseteq C$  and  $(\{a\} \sqcap A) \sqsubseteq C$ , where  $\{A, A_1, A_2\} \subseteq \text{NC}$  and  $a \in \text{NI}$ .

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*.

<sup>1</sup> The definition of witness can be abstracted for any DLs that have  $\mathcal{ALC}\mathcal{IO}$  as a sublanguage and that satisfy some criteria on the interpretations stated in [5].

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 inclusion dependencies. In addition, domain and range constraints are also absorbed in a manner that resembles role absorption introduced in [13].

## 4 A Procedure for Indirect ABox Absorption

In this section, we present a procedure for ABox absorption, which extends binary absorptions [6]. Our algorithm also includes the possibility of absorbing domain and range constraints. The procedure prioritizes binary absorption to keep guarding constraints through restricted uses of nominals, yet it avoids guards for domain and range axioms by employing a variant of role absorption,

The algorithm is given a  $\mathcal{T}_K$  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  $\exists R.T \sqsubseteq C$  (or  $\exists R^-.T \sqsubseteq C$ ). 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  $\mathbf{G} = \{C_1, \dots, C_n\}$  represent the axiom  $\top \sqsubseteq (C_1 \sqcup \dots \sqcup C_n)$ . 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  $(\mathbf{G} \cup \text{NNF}(\neg C) \setminus \{\neg C\})$  to  $\mathcal{T}$ , where the function  $\text{NNF}(\cdot)$  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  $A' \in \text{NC}$ , 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  $\{\neg A_1, \neg A_2\} \subset \mathbf{G}$ , 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  $A' \in \text{NC}$ , 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.
  - (c) 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.
  - (d) 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.

Termination of our procedure can be established by a counting argument.

**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  $(\{\mathcal{T}_{prim} \cup \mathcal{T}_{uinc} \cup \mathcal{T}_{binc} \cup \mathcal{T}_{rinc}, \mathcal{T}_g \cup \mathcal{T}\})$  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  $\mathcal{T}$ .

- In Step 3(a) or Step 3(b),  $\mathcal{T}$  is a correct absorption that derives from [6].
- In Step 3(c),  $\mathcal{T}$  is a correct absorption for domain and range constraints. The correctness proof of this step follows from Lemma 4.3 and 4.4 in [6].
- In Step 3(d),  $\mathcal{T}$  is a correct absorption by [6].
- In any of Steps 1, 2, 5-8,  $\mathcal{T}$  is a correct ABox absorption as they use only equivalence preserving operations.

Thus,  $\mathcal{T}$  is a correct binary absorption by induction.

## 5 Empirical Evaluation

The proposed absorption technique has been implemented in our CARE Assertion Retrieval Engine (CARE) with an underlying  $\mathcal{ALCI}(\mathcal{D})$  DL reasoner. The DL reasoner has no additional ABox reasoning optimizations other than the technique presented here, and it has implemented only axiom absorption and blocking for TBox reasoning. All experiments were conducted on a MacBook with a 2.4GHz Intel Duo processor and 4GB RAM. All times, given in seconds, were averaged out over three independent runs for all reasoners and a reasoning timeout (denoted  $-$ ) is set to 1500 seconds. Queries were posed over a suite of datasets (KBs) describing digital cameras. The KBs consist of digital camera specifications extracted from [DPreview.com](http://DPreview.com) and pricing information from [Amazon.com](http://Amazon.com) in which the seed KB, called DPC1, has one camera instance for each price found through Amazon for a camera model. The other KBs were generated from the seed KB by supplying  $n$  camera instances per price in DPC $n$ . These KBs share the same TBox, i.e., 15 axioms, but have different ABox data, as shown in Figure 2a: it reports the number of individuals, concept, and role assertions, and the number of instances retrieved by each query over these datasets. Test queries are shown in Figure 3, which vary in query forms and selectivity.

|       | Inds  | CAs  | RAs   | Q1 | Q2   | Q3 | Q4   |    | Q1    | Q2    | Q3    | Q4    |
|-------|-------|------|-------|----|------|----|------|----|-------|-------|-------|-------|
| DPC1  | 5387  | 3225 | 7174  | 3  | 244  | 5  | 1426 |    | -     | -     | -     | -     |
| DPC3  | 9711  | 3225 | 18672 | 9  | 732  | 11 | 1432 | NG | -     | -     | -     | -     |
| DPC5  | 14035 | 3225 | 30170 | 15 | 1220 | 17 | 1438 | PG | 178.1 | 183.0 | 186.2 | 181.5 |
| DPC7  | 18359 | 3225 | 41668 | 21 | 1708 | 23 | 1444 | FG | 6.5   | 6.0   | 6.1   | 6.1   |
| DPC10 | 24845 | 3225 | 58915 | 30 | 2440 | 32 | 1453 |    |       |       |       |       |

(a) Description of KBs

(b) Guarding Strategies (DPC1)

Fig. 2: Summary of results

Q1:  $\exists hasInstance^-(Digital\_SLR \sqcap (user\_review = "5.00"))$   
 Q2:  $\exists hasInstance^-(Digital\_SLR \sqcap \neg (user\_review = "5.00"))$   
 Q3:  $\exists hasInstance^-(Digital\_SLR \sqcap (user\_review = "5.00")) \sqcup (price = "01400.00")$   
 Q4:  $\exists hasInstance^-(Digital\_SLR \sqcap (user\_review = "5.00")) \sqcup \neg (price = "01400.00")$

Fig. 3: Test queries

Query response times in Figure 2b compare different guarding strategies. Specifically, “No Guarding” (NG) is a straightforward implementation of the Tableaux algorithm (without assuming the consistency of the KB), “Partial Guarding” (PG) guards individuals so that only relevant individuals will be explored in Tableaux expansions, and “Full Guarding” (FG), in addition to “Partial Guarding,” further guards feature concepts that describe objects in the data so that only query-relevant feature concepts participate in reasoning. Figure 2b shows that CARE timed out under the NG strategy, while it managed to answer all queries under the PG strategy and efficiency increased substantially under the

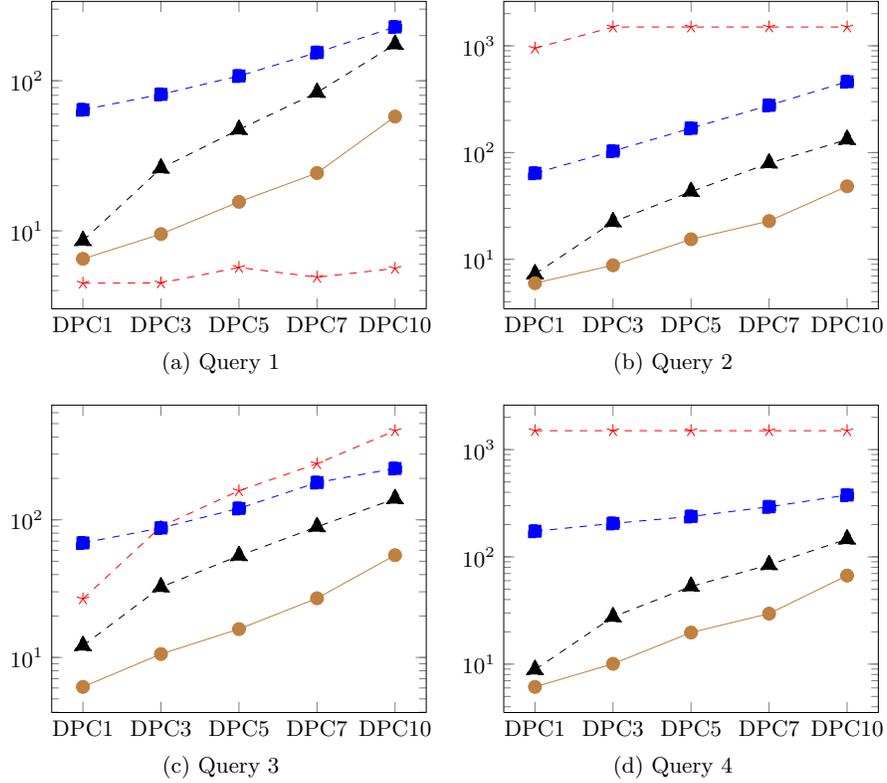


Fig. 4: Query response time comparison (logarithmic scale)

FG strategy. As numerous other optimization techniques have been developed (e.g., [2]) and implemented in most state-of-the-art reasoners, we juxtaposed their performance with CARE to show the efficacy of our proposed technique for ABox reasoning. All queries were posed via OWL API 3 for Pellet 2.3.0 and FaCT++ 1.5.3, and via JRacer in the form of nRQL for RacerPro 2.0. The query response time in Figure 4 does not consider the loading or preprocessing time for other reasoners, yet it includes the ABox absorption time (cf. Sect. 3) for CARE. The results show that CARE outperformed all other reasoners in all queries except Q1 (we believe this is due to deterministic precomputing at KB load time). Given that CARE is not as optimized as other reasoners, the results are significant.

## 6 Summary

We show how, with the presumption that a knowledge base is consistent, one can avoid considering irrelevant ABox individuals to the posed question while preserving soundness and completeness of answers to instance checking tasks.

Our experiments show that in realistic situations arising, e.g., in implementations of *assertion retrieval* [11] 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* [9, 7, 12, 8], 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 [10] could further improve performance.

## References

1. Baader, F., Franconi, E., Hollunder, B., Nebel, B., Profitlich, H.J.: An empirical analysis of optimization techniques for terminological representation systems, or: Making KRIS get a move on. *Applied Artificial Intelligence* 4, 109–132 (1994)
2. Haarslev, V., Möller, R.: On the scalability of description logic instance retrieval. *J. Autom. Reason.* 41(2), 99–142 (Aug 2008), <http://dx.doi.org/10.1007/s10817-008-9104-7>
3. Horrocks, I.: Using an expressive description logic: FaCT or Fiction? In: KR'98. pp. 636–647 (1998)
4. Horrocks, I., Tobies, S.: Optimisation of terminological reasoning. In: *Description Logics'00*. pp. 183–192 (2000)
5. Horrocks, I., Tobies, S.: Reasoning with axioms: Theory and practice. In: KR'00. pp. 285–296 (2000)
6. Hudek, A.K., Weddell, G.E.: Binary absorption in tableaux-based reasoning for description logics. In: *Description Logics'06* (2006)
7. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyashev, M.: The combined approach to query answering in DL-Lite. In: KR'10 (2010)
8. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyashev, M.: The combined approach to query answering in DL-Lite. In: *IJCAI'11*. pp. 2656–2661 (2011)
9. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic EL using a relational database system. In: *IJCAI'09*. pp. 2070–2075 (2009)
10. Motik, B., Shearer, R., Horrocks, I.: Hypertableau reasoning for description logics. *J. Artif. Intell. Res. (JAIR)* 36, 165–228 (2009)
11. Pound, J., Toman, D., Weddell, G.E., Wu, J.: An assertion retrieval algebra for object queries over knowledge bases. In: *IJCAI'11*. pp. 1051–1056 (2011)
12. Rosati, R., Almatelli, A.: Improving query answering over DL-Lite ontologies. In: KR'10 (2010)
13. Tsarkov, D., Horrocks, I.: Efficient reasoning with range and domain constraints. In: *Description Logics'04* (2004)
14. Wu, J., Hudek, A., Toman, D., Weddell, G.: Assertion absorption in object queries over knowledge bases. In: KR'12 (2012)