Assertion Absorption in Object Queries over Knowledge Bases*

Jiewen Wu and Alexander Hudek and David Toman and Grant Weddell

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

Abstract

We develop a novel absorption technique for large collections of factual assertions about individual objects. These assertions are commonly accompanied by implicit background knowledge and form a knowledge base. Both the assertions and the background knowledge are expressed in a suitable language of Description Logic and queries over such knowledge bases can be expressed as assertion retrieval queries. The proposed absorption technique significantly improves the performance of such queries, in particular in cases where a large number of object features are known for the objects represented in such a knowledge base. In addition to the absorption technique we present the results of a preliminary experimental evaluation that validates the efficacy of the proposed optimization.

1 Introduction

Computing certain answers to queries over RDF data sets in the context of an ontology that captures background knowledge of application domains has become an important service in information systems. In this paper, we consider a description logic (DL) based representation of such data sets and ontologies. In particular, we assume a data set and an ontology are given in the form of a knowledge base $K = (T, A)$ over some choice of DL dialect $L$, in which $T$ is a terminology (or TBox) that captures general ontological knowledge, and in which $A$ is a set of assertions (or ABox) that identifies objects of interest to some agent and asserts facts about those objects. In this setting, an object corresponds to an individual name occurring in an ABox.

Perhaps the most basic reasoning tasks that underlie the computation of certain answers to queries over a DL knowledge base concern knowledge base consistency, the problem of determining if a given knowledge base $K$ is consistent, and assertion membership, the problem of determining if a given assertion $a : C$, stating that object $a$ occurring in $A$ belongs to concept $C$, is a logical consequence of a consistent $K$, written $K \models a : C$. Note that these tasks are often combined in the literature in the sense that the latter is assume to include the former. However, in a more practical setting, we believe that a typical workload for a general DL reasoner will include far more instances of assertion membership tasks than of knowledge base consistency tasks. Thus, this separation of concerns can enable technology that is far more efficient for such workloads, particularly so in the case of “non-Horn” DLs that preclude the possibility of computing so-called canonical ABoxes (such as DLs that include disjunction). In this paper, we contribute to this development by introducing a novel absorption technique for knowledge bases and demonstrate that the technique is efficacious for workloads that contain many thousands of assertion membership tasks.

To date, work on absorption has focused on the concept satisfaction problem, a simple case of the assertion membership problem for knowledge bases with an ABox consisting of a single assertion $a : T$. 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 (Baader et al. 1994). It is also imperative for a large TBox to be manipulated by an absorption generation process to maximize the benefits of lazy unfolding in such algorithms, thereby reducing the combinatorial effects of disjunction in underlying chase procedures (Horrocks 1998).

We build on earlier work reported at the description logics workshop (Hudek and Weddell 2006) that proposed a generalization of the absorption theory and algorithms developed in (Horrocks and Tobies 2000a; 2000b) for the problem of concept satisfaction. 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 is the possibility of avoiding the need to internalize (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 (Tsarkov and Horrocks 2004). These ideas, in combination and when coupled with standard equivalences, make it possible for an algorithm to completely absorb, e.g., the TBox definition

$GOODCLIENT \equiv CLIENT \sqcap (\exists Recommend{"\sim"}. BANK) \sqcap (\exists Buy{COSTLY \sqcap PROFITABLE})$

as the set of TBox inclusion dependencies

Copyright © 2012, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

*An extended version of the paper is available as a technical report (Wu et al. 2012).
in which concepts $A_1$, $A_2$, $A_3$ and $A_4$ are fresh atomic concepts introduced by the absorption algorithm.

There are other reasons that binary absorption proves useful, beyond the well-documented advantages of reducing the need for internalization of general terminological axioms. For one, it works very well for the parts of a terminology that are Horn-like. A second reason is a key contribution of this paper. Our contributions are as follows:

1. We propose a generalization of the absorption theory and algorithms pioneered by Horrocks and Tobies (Horrocks and Tobies 2000a; 2000b). The generalization applies for any DL dialect that includes ALCI as a fragment for concept satisfaction tasks.

2. We introduce the notion of role and concrete feature guards in the context of a knowledge base for the DL dialect ALCIT(D). In particular, we show how assertion membership tasks in this dialect can map to concept satisfaction problems in the dialect ALCIT(D), but where binary absorption in combinations with guards can usefully avoid reasoning about, e.g., irrelevant ABox individuals.

3. We report on the results of a preliminary experimental evaluation that validates the efficacy of the proposed optimization.

2 Preliminaries

We study the proposed technique for the logic ALCIT(D); for technical reasons, however, we need to use a (controlled subset of) ALCIOT(D) defined as follows:

Definition 1 (Description Logic ALCIOT(D))

Let $ALCIO(D)$ be a DL based on disjoint infinite sets of atomic concepts $N_c$, atomic roles $N_r$, concrete features $N_f$ and nominals $N_o$. We require that if $A \in N_c$, $R \in N_r$, $o \in N_o$, $N_f$ contains $f, g$ and $C_1$ and $C_2$ are concept descriptions, then $A, A \text{ } \neg C_1, C_1 \cap C_2, C_1 \cup C_2$, $R$, $\top$, $\exists R.C_1$, $\exists R^-.C_1, \forall R.C_1, \forall R^-.C_1, f = k$ and $f < g$, where $k$ is a finite string are also concept descriptions.

An interpretation $I$ is a pair $I = (\Delta^I \uplus \underaccent\tilde{D}^I, \overline{\cdot})$, where $\Delta^I$ is a non-empty set, $\underaccent\tilde{D}^I$ a disjoint concrete domain of finite strings, and $\overline{\cdot}$ is a function mapping each feature $f$ to a total function $f^I : \Delta \to N_f$, the “$=$” symbol to the equality relation over $\underaccent\tilde{D}$, the “$<$” symbol to the binary relation for an alphabetic ordering of $\underaccent\tilde{D}$, a finite string $k$ to itself, $N_c$ to subsets of $\Delta^I$, $N_r$ to subsets of $\Delta^I \times \Delta^I$, and $N_o$ to singleton sets of $\Delta^I$. The interpretation is extended to compound descriptions in the standard way.

We also define the fragment $ALCI(D)$ that disallows the use of the nominal concept constructor.

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 $C_1 \equiv C_2$. A TBox $\mathcal{T}$ is called primitive if it consists entirely of axioms of the form $A \equiv C$ with $A \in N_c$, each $A \in N_c$ appears at most once left hand side of an axiom, and $\mathcal{T}$ is acyclic. $A \in N_c$ is defined in $\mathcal{T}$ if $T$ contains $A \sqsubseteq C$ or $A \equiv C$. An ABox $A$ is a finite set of assertions of the form $a : A$, $a : (f \text{ op } k)$ and $R(a, b)$.

Let $\mathcal{K} = (\mathcal{T}, A)$ be an $ALCIO(D)$ knowledge base (KB). An interpretation $I$ is a model of $\mathcal{K}$, written $I \models \mathcal{K}$, if $C_1^I \sqsubseteq C_2^I$ holds for each $C_1 \sqsubseteq C_2 \in \mathcal{T}$, $C_1^I = C_2^I$ holds for each $C_1 \equiv C_2 \in \mathcal{T}$, $a^I \in A^I$ for $a : A \in A$, $(a^I, b^I) \in R^I$, and $f^I(a^I) \text{ op } k$ for $a : (f \text{ op } k) \in A$. A concept $C$ is satisfiable with respect to a knowledge base $\mathcal{K}$ iff there is an $I$ such that $I \models C$ and such that $C^I \neq \emptyset$.

3 ABox Transformation

In this section we convert an $ALCIO(D)$ knowledge base $\mathcal{K}$ to a TBox by representing individuals in $\mathcal{K}$’s ABox by nominals (a controlled fragment of $ALCIO(D)$) as follows:

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

$\mathcal{T}_A = \{ a \cap \text{Def}_a \subseteq A \mid a : A \in A \}$

$\cup \{ a \cap \text{Def}_f \subseteq (f \text{ op } k) \mid a : (f \text{ op } k) \in A \}$

$\cup \{ a \cap \text{Def}_R \subseteq \exists R. (b \cap \text{Def}_b) \}$

$\{ b \cap \text{Def}_R^- \subseteq \exists R^-.(a \cap \text{Def}_a) \mid R(a, b) \in A \}$

All axioms resulting from ABox assertions are guarded by auxiliary primitive concepts of the form $\text{Def}_a$, $\text{Def}_f$, and $\text{Def}_R$. 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 require w.l.o.g. that the TBox only uses universal restrictions of the form $A \sqsubseteq \forall R.B$ where $A$ and $B$ are (negated) primitive concepts. Subsumptions of the form $\exists R.A \sqsubseteq B$ (considered to be universal restrictions) and nested universal restrictions must be rewritten as well. Thus every $ALCIO(D)$ TBox can be transformed to an equivalent TBox that satisfies this restriction by introducing new concept names. Then the following assertions can manipulate the guards:

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

$\mathcal{T}_F = \{ A \sqsubseteq \text{Def}_R, \neg B \sqsubseteq \text{Def}_R^- \mid A \sqsubseteq \forall R.B \in \mathcal{T} \}$

$\cup \{ (t_1 \text{ op } t_2) \subseteq \text{Def}_f \mid f \text{ appears in } t_1 \text{ or in } t_2, \}$

$(t_1 \text{ op } t_2) \text{ appears in } \mathcal{T} \}.$

In the following we use $\mathcal{T}_K$ for $\mathcal{T} \cup \mathcal{T}_F \cup \mathcal{T}_A$.

Theorem 5 Let $\mathcal{K} = (\mathcal{T}, A)$ be a consistent knowledge base. Then

$\mathcal{K} \models a : C$ if and only if $\mathcal{T}_K \models a \cap \text{Def}_a \subseteq C$,

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

Proof. A sketch. Assume that there is an interpretation $I_0$ that satisfies $\mathcal{T}_K$ 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 existential restrictions are fulfilled by anonymous objects. We
assume w.l.o.g. both \( I_0 \) and \( I_1 \) are tree-shaped outside of the converted ABox. We construct an interpretation \( J \) for \( K \cup \{ a : \neg C \} \) as follows: Let \( \Gamma_{I_0} \) be the set of objects \( o \in \Delta^I_{I_0} \) 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 \( \Delta^I_{I_0} \) rooted by such an object. Similarly let \( \Gamma_{I_1} \) be the set of objects \( o \in \Delta^I_{I_1} \) such that either \( o = (a)_{I_1} \) and \( (a)_{I_1} \cap \text{Def}_{a_{I_1}} = \emptyset \) or \( o \) is an anonymous object in \( \Delta^I_{I_1} \) rooted by such an object. We set

1. \( \Delta^J = \Gamma_{I_0} \cup \Gamma_{I_1} \);
2. \( (a)^J \in (a)^I_{\Delta^I} \) if \( (a)^J \in \Delta^I_{I_0} \) and \( (a)^J = (a)_{I_1} \) for \( (a)^J \in \Delta^I_{I_1} \);
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 \( A \in N_{C} \) (similarly for concrete domain concepts);
4. \( (o_1, o_2) \in (R)^J \) if
   (a) \( (o_1, o_2) \in R_{I_0} \) (resp. \( R_{I_1} \)) and \( o_1, o_2 \in \Gamma_{I_0} \) (resp. \( o_1, o_2 \in \Gamma_{I_1} \)), or
   (b) \( o_1 \in (a)_{I_0} \cap \text{Def}_{a_{I_0}} \) and \( o_2 \in (b)_{I_1} \) if \( a, b \in A \) (or vice versa).

Then \( (a)^J \cap (C)^J = \emptyset \) holds trivially. To show \( J \models K \) we only consider those \( R \) edges in (4b) that cross between the two interpretations, i.e., when \( o_1 \in (a)_{I_0} \) and \( o_2 \in (b)_{I_1} \) and \( R(a, b) \in A \). These edges are not needed to fulfill existential restrictions, which are fulfilled by anonymous objects. For universal restrictions expressed as \( \Delta_{\forall} R.B \in \mathcal{T} \), we can conclude that \( o_1 \notin (A)^{I_0} \) as otherwise \( o_1 \in (\text{Def}_{a_{I_0}})^{I_0} \) by Definition 4 and thus \( o_2 \in (\text{Def}_{b_{I_1}})^{I_1} \) by Definition 3 which contradicts the assumption \( (b)^{I_1} \cap (\text{Def}_{b_{I_1}})^{I_1} = \emptyset \). Hence the inclusion dependency is satisfied vacuously. The edges in (4a) satisfy all dependencies in \( K \) as the remainder of the interpretation \( J \) is copied from one of the two interpretations; hence \( J \models K \). The proof of the other direction is trivial. □

The proof idea can be extended to handle number restrictions by treating at most restrictions in a similar way as universal restrictions.

### 4 Binary Absorptions

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 labelled rooted tree with “back edges”). The data structure “encodes” a partial description of (eventual) interpretations \( \mathcal{I} \) for which \( C^I \) will be non-empty. Such a partial description will almost always abstract details on class membership for hypothetical elements of \( \Delta^W \) 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. Horrocks and Tobies have done this by introducing the following 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 6 (Witness)** Let \( L \) be a DL and \( C \in L \) a concept. A witness \( W = (\Delta^W, \mathcal{W}_{\Delta^W}, \mathcal{L}^W) \) for \( C \) consists of a non-empty set \( \Delta^W \), a function \( \mathcal{W}_{\Delta^W} \) that maps \( N_{C} \) to subsets of \( \Delta^W \times \Delta^W \), and a function \( \mathcal{L}^W \) that maps \( \Delta^W \) to subsets of \( L \) such that:

1. \( W \) there is some \( x \in \Delta^W \) with \( C \in \mathcal{L}^W(x) \),
2. \( \text{W2} \) there is an interpretation \( \mathcal{I} \in \text{Int}(L) \) that stems from \( W \), and
3. \( \text{W3} \) for each interpretation \( \mathcal{I} \in \text{Int}(L) \) that stems from \( W \), \( x \in C^\mathcal{I} \) if \( C \in \mathcal{L}^W(x) \).

An interpretation \( \mathcal{I} = (\Delta^\mathcal{I}, \mathcal{L}^\mathcal{I}) \) is said to stem from \( W \) if \( \Delta^\mathcal{I} = \Delta^W, \mathcal{L}^\mathcal{I}_{\Delta^\mathcal{I}} = \mathcal{W}_{\Delta^\mathcal{I}} \) for each \( A \in N_{C} \). \( A \in \mathcal{L}^W \) implies \( x \in A^2 \) and \( \neg A \in \mathcal{L}^W \) implies \( x \notin A^2 \) for each \( a \in N_{C} \). \( a \in \mathcal{L}^W(x) \) implies \( x \_ a \) for each \( \{ f \mathcal{op} k \} \in \mathcal{L}^W(x) \) implies \( x \notin \{ f \mathcal{op} k \}^2 \).

A witness \( W \) is called admissible with respect to a TBox \( T \) if there is an interpretation \( \mathcal{I} \in \text{Int}(L) \) that stems from \( W \) with \( \mathcal{I} \models T \).

The properties satisfied by a witness have been captured by the original lemmas 2.6 and 2.7 in (Horrocks and Tobies 2000b). We further extend binary absorption (Hudek and Weddell 2006) to allow for assertions absorbed in Section 3.

**Definition 7 (Binary Absorption)** Let \( L \) and \( K = (T, A) \) be a DL and a KB, respectively. A binary absorption of \( T \) is a pair of TBoxes \((T_u, T_v)\) such that \( T \equiv T_u \cup T_v \) and \( T_u \) contains axioms of the form \( A_1 \subset C, \neg A_1 \subset C \), and \( a \subset C \) or the form \( (\mathcal{A}_1 \cap \mathcal{A}_2) \subset C \) and \( a \cap A \subset C \) where \( \{ A_1, A_2 \} \subset N_{C} \) and \( a \subset N_{C} \).

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

The distinguishing feature of our extension is the addition of the first three implications in Figure 1. Binary absorption allows additional axioms in \( T_u \) to be dealt with in a deterministic manner, as we illustrate in our introductory example. Assertion absorption, treating ABox assertions as axioms, also contributes to binary absorption via nominals.

### 5 A Binary Absorption Algorithm

In this section, we extend binary absorptions (Hudek and Weddell 2006) that derive from the absorption algorithm for the FaCT system outlined in earlier work (Baader et al. 2003; Horrocks and Tobies 2000a; 2000b). In particular, when coupled with assertion absorption the extended algorithm makes it possible to retain guarding constraints as much as possible by prioritizing binary absorptions.

The algorithm is given a \( T \) that consists of arbitrary axioms. It proceeds by constructing four TBoxes such that:

- \( T \equiv T_u \cup T_{\mathcal{prim}} \cup T_{\mathcal{inc}} \cup T_{\mathcal{bin}} \) \( T_{\mathcal{prim}} \) is primitive, \( T_{\mathcal{inc}} \) consists of axioms of the form \( A_1 \subset C, a \subset C \), and \( T_{\mathcal{bin}} \) consists of axioms of the form \( (\mathcal{A}_1 \cap \mathcal{A}_2) \subset C \) and \( (a \cap A) \subset C \) and none of the above primitive concept are defined in \( T_{\mathcal{prim}} \). Here, \( T_{\mathcal{inc}} \) contains unary absorptions and \( T_{\mathcal{bin}} \) contains binary absorptions.

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

1. If \( X \) is of the form \( A \equiv C, A \) is not defined in \( T_{\mathcal{prim}} \), and \( T_{\mathcal{prim}} \cup \{ X \} \) is primitive, then move \( X \) to \( T_{\mathcal{prim}} \).
\[ a \in L^W(x) \text{ and } a \in L^W(y) \quad \text{implies} \quad x = y. \]
\[ \{a, A\} \subseteq L^W(x), \text{ and } (a \cap A) \subseteq C \in T_u \quad \text{implies} \quad C \in L^W(x), \]
\[ \{a\} \subseteq L^W(x) \text{ and } \{a\} \subseteq C \in T_u \quad \text{implies} \quad C \in L^W(x), \]
\[ \{A_1, A_2\} \subseteq L^W(x) \text{ and } (A_1 \cap A_2) \subseteq C \in T_u \quad \text{implies} \quad C \in L^W(x), \]
\[ A \in L^W(x) \text{ and } A \subseteq C \in T_u \quad \text{implies} \quad C \in L^W(x), \]
\[ \neg A \in L^W(x) \text{ and } \neg A \subseteq C \in T_u \quad \text{implies} \quad C \in L^W(x), \]
\[ C_1 \subseteq C_2 \in T_g \quad \text{implies} \quad \neg C_1 \cup C_2 \in L^W(x), \]
\[ C_1 \subseteq C_2 \in T_g \quad \text{implies} \quad \neg C_1 \cup C_2 \in L^W(x), \]
\[ C_1 \subseteq C_2 \in T_g \quad \text{implies} \quad C_1 \cup \neg C_2 \in L^W(x). \]

Figure 1: Absorption Witness Conditions

1. If \( T \) is empty, then return the binary absorption \( \{(A \subseteq C, \neg A \subseteq \neg C) \mid A \subseteq C \in T_{\text{prim}} \cup T_{\text{uinc}} \cup T_{\text{binc}}, T_g \} \).

2. Otherwise, remove an axiom \( G \) from \( T \).

3. Simplify \( G \).

(a) If there is some \( \neg C \in G \) such that \( C \) is not a primitive concept, then add \( (G \cup \text{NNF}(\neg C)) \setminus \{\neg C\} \) to \( T \), where the function \( \text{NNF}(\cdot) \) converts concepts to negative normal form. Return to Step 1.

(b) If there is some \( C \in G \) such that \( C \) is of the form \( (C_1 \cap C_2) \), then add both \( (G \cup \{C_1\}) \setminus \{C\} \) and \( (G \cup \{C_2\}) \setminus \{C\} \) to \( T \). Return to Step 1.

(c) If there is some \( C \in G \) such that \( C \) is of the form \( C_1 \cup C_2 \), then apply associativity by adding \( (G \cup \{C_1, C_2\}) \setminus \{C_1 \cup C_2\} \) to \( T \). Return to Step 1.

4. Partially absorb \( G \).

(a) If \( \{\neg a, \neg A\} \subseteq G \), and \( A \) is a guard, then do the following. If an axiom of the form \( (a \cap A) \subseteq A' \) is in \( T_{\text{binc}} \), add \( G \cup \{\neg A'\} \setminus \{\neg a, \neg A\} \) to \( T \). Otherwise, introduce a new concept \( A' \in \mathbb{N}_c \), add \( (G \cup \{\neg A'\}) \setminus \{\neg a, \neg A\} \) to \( T \), and \( (a \cap A) \subseteq A' \) to \( T_{\text{binc}} \). Return to Step 1.

(b) If \( \{\neg A_1, \neg A_2\} \subseteq G \), and neither \( A_1 \) nor \( A_2 \) are defined in \( T_{\text{prim}} \), then do the following. If an axiom of the form \( (A_1 \cap A_2) \subseteq A' \) is in \( T_{\text{binc}} \), add \( G \cup \{\neg A'\} \setminus \{\neg A_1, \neg A_2\} \) to \( T \). Otherwise, introduce a new concept \( A' \in \mathbb{N}_c \), add \( (G \cup \{\neg A'\}) \setminus \{\neg A_1, \neg A_2\} \) to \( T \), and \( (A_1 \cap A_2) \subseteq A' \) to \( T_{\text{binc}} \). Return to Step 1.

(c) If \( \forall R.A \) (resp. \( \forall R.\neg A \)) \in G, then do the following. Introduce a new internal primitive concept \( A' \) and add both \( A \subseteq \forall R.\neg A' \) (resp. \( A \subseteq \forall R.A' \)) and \( (G \cup \{\neg A'\}) \setminus \{\forall R.\neg A\} \) (resp. \( \{\forall R.A \} \)) to \( T \). Return to Step 1.

4. Unfold \( G \). If, for some \( A \in G \) (resp. \( \neg A \in G \)), there is an axiom \( A \equiv C \) in \( T_{\text{prim}} \), then substitute \( A \in G \) (resp. \( \neg A \in G \)) with \( C \) (resp. \( \neg C \)), and add \( G \) to \( T \). Return to Step 1.

5. Absorb \( G \). If \( \neg a \in G \), add \( a \subseteq C \) to \( T_{\text{uinc}} \) where \( C \) is the disjunction of \( G \setminus \{\neg a\} \). Return to Step 1.

6. Absorb \( G \). If \( \neg A \in G \) and \( A \) is not defined in \( T_{\text{prim}} \), add \( A \subseteq C \) to \( T_{\text{uinc}} \) where \( C \) is the disjunction of \( G \setminus \{\neg A\} \). Return to Step 1.

7. If none of the above are possible (\( G \) cannot be absorbed), add \( G \) to \( T_g \). Return to Step 1.

Termination of our procedure can be established by a counting argument involving concept constructors in \( T \).

Theorem 8 For any TBox \( T \), the binary absorption algorithm computes a correct absorption of \( T \).

Proof. The proof is by induction on iterations of our algorithm. We abbreviate the pair \((T_{\text{prim}} \cup T_{\text{uinc}} \cup T_{\text{binc}}, T_g)\) as \( T \) and claim that this pair is always a correct binary absorption. Initially, \( T_{\text{uinc}}, T_{\text{binc}}, \) and \( T_g \) are empty, primitive axioms are in \( T_{\text{prim}} \), and the remaining axioms are in \( T \).

- In Step 3(a) or Step 3(b), a newly introduced primitive concept only appears on the left hand side of an axiom once, hence \( T \) is a correct binary absorption.
- In Step 3(c), \( T \) is a correct binary absorption.
- In any of Steps 1, 2, 5-8, \( T \) is a correct binary absorption as they use only equivalence preserving operations.

Thus, \( T \) is a correct binary absorption by induction.

6 Experiments

Our empirical studies focus on an important class of search queries, assertion retrieval, proposed in earlier work (Pound et al. 2011). Assertion retrieval queries are of the form \( (C, P) \), in which \( C \) is a query concept in some DL \( L \) serving the same role as in instance retrieval and in which \( P \) is a projection description that defines a subset of the concepts of \( L \). Assertion retrieval thus reduces to instance retrieval if \( P \) = \( L \). Assertion queries return assertions of the form \( a : C_a \) such that \( K \models a : (C \cap C_a) \), where \( C_a \) provides the most informative answer for which this condition holds. A projection description generalizes the effect of the relational projection operation by controlling both the format and information content of query results.

Our experiments were carried out on a digital camera KB, where each camera has a list of over 70 feature-value pairs in specification. This ACC KB has 15 axioms, 1931 assertions, 1029 instances, and 2117 constants. We evaluated five different assertion queries of the form \( (C, P) \), which have the
same selection concept C and different Pd’s (in increasing difficulty) over the qualifying instances. The KB and queries are available at http://db-tom.cs.uwaterloo.ca.

The times given in Table 1, averaged over three independent runs on a Macbook with Intel Core 2 Duo processor and 4GB memory, compare five queries under three environments: NG, PG and FG, which refer to No Guarding that checks all individuals in K, Partial Guarding that checks relevant individuals instead, and Full Guarding that, in addition to PG, introduces guards for features to reason about only relevant domain concepts, respectively. In NG, a Tableaux procedure needs to check all named individuals in K, which makes query evaluation infeasible (timed out after 1000 seconds, as denoted by –). The dramatic differences depicted in Table 1 suggest that, without guarding, evaluating object queries in large datasets often becomes infeasible. The results for these five queries imply that FG reduces the total running time by at least 75 percent of that employing PG. Particularly, FG effectively trims the graph size by more than an order of magnitude during clash finding in concrete domain, thus leading to a substantial runtime improvement for query answering in KBs that have a considerable number of features. In a nutshell, the proposed optimization in this paper is a promising technique for querying large ABoxes, especially when objects are described by numerous domain concepts.

Table 1: Experimental Results (time in seconds)

<table>
<thead>
<tr>
<th>Q1</th>
<th>Q2</th>
<th>Q3</th>
<th>Q4</th>
<th>Q5</th>
</tr>
</thead>
<tbody>
<tr>
<td>NG</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PG</td>
<td>29.98</td>
<td>29.80</td>
<td>30.33</td>
<td>32.28</td>
</tr>
<tr>
<td>FG</td>
<td>1.48</td>
<td>1.72</td>
<td>3.64</td>
<td>5.98</td>
</tr>
</tbody>
</table>

7 Conclusion

We have developed a technique that allows DL reasoners to avoid exploring a large fraction of individuals in a knowledge base that include a very large ABox in order to perform assertion membership tasks. We show how, with the presumption that the knowledge base is consistent, one can avoid considering irrelevant ABox individuals to the posed question while preserving soundness and completeness 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 (Horrocks and Tobies 2000a; 2000b). This extension, called binary absorption, originally designed for TBox reasoning alone (Hudek and Weddell 2006), allows terminological axioms of the form (a ⊓ A) ⊆ C to qualify for lazy unfolding in model building satisfaction procedures for description logics, such as those based on tableau techniques. 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 tableau algorithms will avoid creating instances of such individuals altogether.) Such an optimization cannot be achieved when only unary absorption is available.

The experiments show that in realistic situations arising, e.g., in implementations of assertion retrieval (Pound et al. 2011) in which a number of assertion membership queries are needed to answer a single user query, or in the case of ontology-based query answering (Lutz, Toman, and Wolter 2009; Kontchakov et al. 2010; Rosati and Almatelli 2010; Kontchakov et al. 2011), 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 produced in few seconds, attempting the same tasks without the optimization is not feasible.

References


