Scalable Reasoning by Abstraction Beyond DL-Lite

Birte Glimm, Yevgeny Kazakov, and Trung-Kien Tran

University of Ulm, Germany, <first name>.<last name>@uni-ulm.de

Abstract. Recently, it has been shown that ontologies with large datasets can be efficiently materialized by a so-called abstraction refinement technique. The technique consists of the *abstraction* phase, which partitions individuals into equivalence classes, and the *refinement* phase, which re-partitions individuals based on entailments for the representative individual of each equivalence class. In this paper, we present an *abstraction*-based approach for materialization in DL-Lite, i.e. we show that materialization for DL-Lite does not require the refinement phase. We further show that the approach is sound and complete even when adding disjunctions and nominals to the language. The proposed technique allows not only for faster materialization and classification of the ontologies, but also for efficient consistency checking; a step that is often omitted by practical approaches based on query rewriting. A preliminary empirical evaluation on both real-life and benchmark ontologies demonstrates that the approach can handle ontologies with large datasets efficiently.

1 Introduction

Over many years, Description Logics (DLs) have been very popular languages for knowledge representation and reasoning. Among the various fragments of Description Logics, DL-Lite [3, 1] is a family of languages specifically designed for ontology-based data access (OBDA). In this setting, an ontology with background knowlege (a TBox) can be seen as a conceptual view over data repositories (ABoxes), and data can be accessed via query answering services. Common techniques for query answering in DL-Lite are (pure) *rewriting* [3] and *combined* approaches [13, 5, 6]. In the rewriting approaches, OBDA systems exploit the background knowledge and rewrite the input query so that the rewritten queries are sufficient to retrieve the complete query answer when evaluated over the unmodified data. As the rewritten queries can be very large or complex [12], several optimization techniques have been proposed with the aim of reducing or simplifying the rewritten queries [16, 17, 9, 2]. Combined approaches complement the pure rewriting approaches; they also work for DL fragments that allow for qualified existential quantification. In contrast to pure rewriting, the combined approaches not only rewrite the input query, but also partially or completely expand the data taking the ontology/schema into account. The latter operation is called *data completion* or *ontology materialization*. It plays an important role in the overall performance of the combined approaches, given the fact that the data is often very large in the OBDA applications. In addition, performing ontology materialization only, OBDA systems are already able to provide the complete answers for instance queries. In this paper, we investigate the application of the novel materialization technique via abstraction refinement [7] for DL-Lite ontologies.

The existing abstraction refinement approach consists of two phases: the *abstraction phase* and the *refinement phase*. In the abstraction phase, individuals in the ABox are partitioned into equivalence classes, which are then used to construct a so-called *abstract* ABox. Entailments of the abstract ABox are transformed to entailments for the original ABox, which might result in some individuals no longer belonging to the same equivalence class. Therefore, the previous steps are repeated in the refinement phase, e.g. individuals are re-partitioned, until, eventually, the fixed-point is reached. The approach presented in this paper can be regarded as an enhancement of the existing abstraction refinement approach tailored towards ontologies in DL-Lite and beyond. We make the following contributions:

- We present an abstraction-based approach for materialization for DL-Lite$_{core}^{\mathcal{H}\sqcup}$, an extension of DL-Lite$_{core}$ with role inclusions and disjunctions. The limited form of existential restrictions in DL-Lite enables an efficient way to transform entailments from the abstract ABox to the original ABox. In addition, the presented approach does not require the refinement phase. This allows not only for faster materialization but also for efficient consistency checking of the ontologies. Query answering only makes sense if the ontology is consistent. Therefore, checking consistency is necessary, but this step is often omitted in many query rewriting systems.[1]
- We show that the presented approach is also sound and complete when adding nominals. Moreover, it can be extended to ontology classification, a non-trivial reasoning task in the presence of nominals.
- We evaluate our approach on both real-life and benchmark ontologies. The empirical results demonstrate that the size of the ABoxes can be reduced by orders of magnitude and, as a result, reasoning via abstraction is often much faster than reasoning over the original ontology.

2 Preliminaries

The syntax of DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ is defined using a vocabulary consisting of countably infinite disjoint sets N_C of *atomic concepts*, N_O of *nominals*, N_R of *atomic roles*, and N_I of *individuals*. A role is either atomic or an *inverse role* r^-, $r \in N_R$. We define the inverse R^- of a role R by $R^- := r^-$ if $R = r$ and $R^- := r$ if $R = r^-$. Complex *concepts* and *axioms* are defined recursively in Table 1. An *ABox* is a finite set of *concept assertions* of the form $A(a)$ and *role assertions* of the form $R(a, b)$ with $A \in N_C, R \in N_R \cup \{r^- \mid r \in N_R\}$, and $a, b \in N_I$. A *TBox* is a finite set of role and concept inclusions. An *ontology* $\mathcal{O}$, written as $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, consists of an ABox $\mathcal{A}$ and a TBox $\mathcal{T}$. W.l.o.g. we do not distinguish between the axioms $R(a, b)$ and $R^-(b, a)$ as well as $R \sqsubseteq S$ and $R^- \sqsubseteq S^-$. We use $\mathsf{con}(\mathcal{O})$, $\mathsf{rol}(\mathcal{O})$, $\mathsf{ind}(\mathcal{O})$, $\mathsf{nom}(\mathcal{O})$ for the sets of atomic concepts, atomic roles, individuals, and nominals occurring in $\mathcal{O}$, respectively. By DL-Lite$_{core}^{\mathcal{H}\sqcup}$ we denote the fragment of DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ that disallows nominals.

An *interpretation* $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ consists of a non-empty set $\Delta^{\mathcal{I}}$, the *domain* of $\mathcal{I}$, and an *interpretation function* $\cdot^{\mathcal{I}}$, that assigns to each $A \in N_C$ a subset $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, to

[1] If $\bot$ is allowed in the language, consistency checking can be reduced to querying instances of $\bot$ but it also requires reasoning over the whole data.

Table 1. The syntax and semantics of DL-Lite$_{core}^{\mathcal{HO}\sqcup}$

	Syntax	Semantics
Roles:		
atomic role	R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$
inverse role	R^{-}	$\{\langle e, d\rangle \mid \langle d, e\rangle \in R^{\mathcal{I}}\}$
Concepts:		
atomic concept	A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$
nominal	o	$o^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}, \|o^{\mathcal{I}}\| = 1$
top	$\top$	$\Delta^{\mathcal{I}}$
bottom	$\bot$	$\emptyset$
negation	$\neg C$	$\Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}$
conjunction	$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
disjunction	$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
existential restriction	$\exists R$	$\{d \mid \exists e \in \Delta^{\mathcal{I}} : \langle d, e\rangle \in R^{\mathcal{I}}\}$
Axioms:		
concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
concept assertion	$A(a)$	$a^{\mathcal{I}} \in A^{\mathcal{I}}$
role assertion	$R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}}\rangle \in R^{\mathcal{I}}$

each $o \in N_O$ a singleton subset $o^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$, $\|o^{\mathcal{I}}\| = 1$, to each $r \in N_R$ a binary relation $r^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$, and to each $a \in N_I$ an element $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$. This assignment is extended to roles and to complex concepts as shown in Table 1. An interpretation $\mathcal{I}$ *satisfies* an axiom α (written $\mathcal{I} \models \alpha$) if the corresponding condition in Table 1 holds. Given an ontology $\mathcal{O}$, $\mathcal{I}$ is a *model* of $\mathcal{O}$ (written $\mathcal{I} \models \mathcal{O}$) if $\mathcal{I} \models \alpha$ for all axioms $\alpha \in \mathcal{O}$; $\mathcal{O}$ *is consistent* if $\mathcal{O}$ has a model; and $\mathcal{O}$ *entails* an axiom α (written $\mathcal{O} \models \alpha$), if every model of $\mathcal{O}$ satisfies α.

For an ontology $\mathcal{O}$, we say that $\mathcal{O}$ is *concept-materialized* if $\mathcal{O} \models A(a)$ implies $A(a) \in \mathcal{O}$ for each $A \in \mathsf{con}(\mathcal{O})$ and $a \in \mathsf{ind}(\mathcal{O})$; $\mathcal{O}$ is *role-materialized* if $\mathcal{O} \models r(a, b)$ implies $r(a, b) \in \mathcal{O}$ for each $r \in \mathsf{rol}(\mathcal{O})$ and $a, b \in \mathsf{ind}(\mathcal{O})$; $\mathcal{O}$ is (fully) *materialized* if it is both concept and role materialized. The concept-, role-, and/or (full) materialization of an ontology $\mathcal{O}$ is the smallest super-set of $\mathcal{O}$ that is concept-, role-, and/or fully materialized respectively. Given an ontology, traditional reasoning tasks include *ontology materialization*: computing the materialization of the ontology, *ontology classification*: computing all entailed concept inclusions between atomic concepts in the ontology, and *consistency checking*: checking if the ontology is consistent.

3 Reasoning by Abstraction

The general idea of reasoning via abstraction is to reduce reasoning over a large ABox to reasoning over a smaller one. Specifically, one first builds a suitable *abstraction* of the *original* ontology; performs reasoning over the abstraction; and then *transfers* entailments of the abstraction to corresponding entailments of the original ontology. Correctness of the reduction is based on homomorphisms between ABoxes.

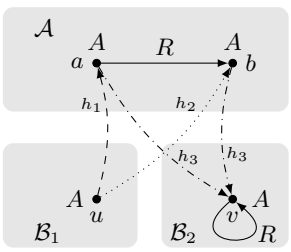

Fig. 1. Visualization of the ABoxes and homomorphisms in Example 1

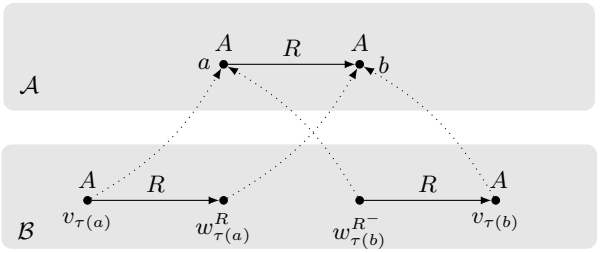

Fig. 2. Visualization of the ABoxes $\mathcal{A}$ from Example 4 and its abstraction $\mathcal{B}$ from Example 5, where the dotted lines show the homomorphism from $\mathcal{B}$ to $\mathcal{A}$ induced by the abstraction

Definition 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be ABoxes. A mapping $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ is called a* homomorphism (from $\mathcal{B}$ to $\mathcal{A}$) *if, for every assertion $\alpha \in \mathcal{B}$, we have $h(\alpha) \in \mathcal{A}$, where $h(C(a)) := C(h(a))$ and $h(R(a,b)) := R(h(a), h(b))$.*

Example 1. Consider the ABoxes $\mathcal{A} = \{A(a),\ A(b),\ R(a,b)\}$, $\mathcal{B}_1 = \{A(u)\}$, and $\mathcal{B}_2 = \{A(v),\ R(v,v)\}$ visualized in Figure 1. Then the mappings $h_1 = \{u \mapsto a\}$ and $h_2 = \{u \mapsto b\}$ are homomorphisms from $\mathcal{B}_1$ to $\mathcal{A}$; and the mapping $h_3 = \{a \mapsto v,\ b \mapsto v\}$ is a homomorphism from $\mathcal{A}$ to $\mathcal{B}_2$.

The following property of homomorphisms allows us to establish the relation between entailments of one ontology and those of the other.

Lemma 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be ABoxes, and $h\colon \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. Then, for every TBox $\mathcal{T}$ and every axiom α, $\mathcal{B} \cup \mathcal{T} \models \alpha$ implies $\mathcal{A} \cup \mathcal{T} \models h(\alpha)$.*

Note that Lemma 1 is not restricted to DL-Lite$^{\mathcal{HO}\sqcup}_{core}$ and it holds for any DL with (classical) set-theoretic semantics, e.g. $\mathcal{SROIQ}$ [10]. The following two corollaries illustrate aspects of homomorphisms that are of particular relevance for our approach, namely that consistency and (concept) entailments are preserved under homomorphisms.

Corollary 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be ABoxes, $h\colon \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ a homomorphism from $\mathcal{B}$ to $\mathcal{A}$, $a \in \mathsf{ind}(\mathcal{A})$ and $b \in \mathsf{ind}(\mathcal{B})$ such that $h(b) = a$. Then, for every TBox $\mathcal{T}$ and concept C, $\mathcal{B} \cup \mathcal{T} \models C(b)$ implies $\mathcal{A} \cup \mathcal{T} \models C(a)$.*

Corollary 2. *Let $\mathcal{A}$ and $\mathcal{B}$ be ABoxes. If there exists a homomorphism from $\mathcal{B}$ to $\mathcal{A}$ then, for every TBox $\mathcal{T}$, $\mathcal{A} \cup \mathcal{T}$ is consistent implies $\mathcal{B} \cup \mathcal{T}$ is consistent.*

The abstraction is obtained by partitioning individuals in the original ABox into equivalence classes and by using just one *representative* individual for each equivalence class. Entailments of the representatives are then *transferred* to the corresponding entailments for individuals in the equivalence classes.

If we use the individual u in Example 1 as the representative for a and b, then, for any TBox $\mathcal{T}$, one can *transfer* any newly entailed concept assertion for u to the corresponding assertions for a and b by Corollary 1. However, not all entailments for a and b can necessarily be computed this way.

Example 2 (Example 1 continued). Consider a TBox $\mathcal{T} = \{A \sqsubseteq C, \exists R^- \sqsubseteq B\}$. We have $\mathcal{B}_1 \cup \mathcal{T} \models C(u)$. By Corollary 1, we obtain $C(a)$, $C(b)$ entailed by $\mathcal{A} \cup \mathcal{T}$. We are, however, not able to obtain $B(b)$ via homomorphisms from $\mathcal{B}_1$ to $\mathcal{A}$, although $B(b)$ is entailed by $\mathcal{A} \cup \mathcal{T}$.

Also in Example 1, since there is a homomorphism from $\mathcal{A}$ to $\mathcal{B}_2$, for any TBox $\mathcal{T}$, if $\mathcal{B}_2 \cup \mathcal{T}$ is consistent then $\mathcal{A} \cup \mathcal{T}$ is consistent by Corollary 2. Furthermore, if we use the individual v as the representative for a and b (ignoring that there is no homomorphism from $\mathcal{B}_2$ to $\mathcal{A}$), then we can compute all entailments for a and b based on the entailments of v. However, we might transfer facts that are not entailed by $\mathcal{A} \cup \mathcal{T}$.

Example 3 (Example 2 continued). We have $\mathcal{B}_2 \cup \mathcal{T} \models \{B(v), C(v)\}$. If we take v as representative of both a and b, then we obtain $B(a)$, $B(b)$, $C(a)$, $C(b)$. However, $B(a)$ is not entailed by $\mathcal{A} \cup \mathcal{T}$.

As demonstrated in Example 2 and Example 3, it is often easy to obtain either sound or complete results but it is challenging to obtain both. The SHER approach [4] addresses this issue by computing complete but possibly unsound entailments of the ontology using a compressed, so-called summary ABox and by using justification techniques [11] to refine the summary. The abstraction refinement approach [7] computes sound but possibly incomplete entailments. To ensure completeness further refinement steps are employed based on the newly derived entailments. In the next section, we present an enhancement of the existing abstraction refinement approach that is only based on the abstraction. We show that indeed no refinement is needed to obtain both sound and complete entailments for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ ontologies. To simplify presentation, we first present the solution for DL-Lite$_{core}^{\mathcal{H}\sqcup}$ and then discuss the extensions for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$.

4 Abstraction for DL-Lite$_{core}^{\mathcal{H}\sqcup}$

To construct the abstraction of the original ABox, we partition individuals in the original ABox into equivalence classes and use just one *representative* individual for each equivalence class. The equivalence classes are characterized by the *type* of individuals, which can be syntactically computed from the original ABox.

Definition 2. *Let $\mathcal{A}$ be an ABox and a an individual. The* type *of a (w.r.t. $\mathcal{A}$) is a pair $\tau(a) = \langle \tau_C(a), \tau_R(a) \rangle$ where $\tau_C(a) = \{A \mid A(a) \in \mathcal{A}\}$ and $\tau_R(a) = \{R \mid \exists b : R(a,b) \in \mathcal{A}\}$.*

Example 4. Let $\mathcal{A} = \{A(a), A(b), R(a,b)\}$ be as in Example 1 (cf. Figure 1). Then, we have $\tau(a) = \langle \{A\}, \{R\} \rangle$ and $\tau(b) = \langle \{A\}, \{R^-\} \rangle$.

The abstract ABox is then constructed by introducing one representative and the respective assertions for each type.

Definition 3. *The* abstraction *of an ABox $\mathcal{A}$ is an ABox $\mathcal{B} = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathcal{B}_{\tau(a)}$, where, for each type $\tau(a) = \langle \tau_C, \tau_R \rangle$, $\mathcal{B}_{\tau(a)} = \{A(v_{\tau(a)}) \mid A \in \tau_C\} \cup \{R(v_{\tau(a)}, w^R_{\tau(a)}) \mid R \in \tau_R\}$, where $v_{\tau(a)}$ and $w^R_{\tau(a)}$ are fresh, distinguished* abstract individuals *for each type $\tau(a)$.*

Example 5. The abstraction for $\mathcal{A}$ in Example 4 is the ABox $\mathcal{B} = \mathcal{B}_{\tau(a)} \cup \mathcal{B}_{\tau(b)}$, where $\mathcal{B}_{\tau(a)} = \{A(v_{\tau(a)}),\, R(v_{\tau(a)}, w^R_{\tau(a)})\}$, $\mathcal{B}_{\tau(b)} = \{A(v_{\tau(b)}), R^-(v_{\tau(b)}, w^{R^-}_{\tau(b)})\}$ (cf. Figure 2).

Note that the size of the abstraction of a small ABox may be larger than the size of the original ABox, but for ontologies with a large ABox, many individuals have the same type and, hence, abstractions are small.

Intuitively, the abstraction of an ABox is a disjoint union of small ABoxes witnessing each individual type realized in the ABox. There always exist homomorphisms from the abstraction to the original Abox.

Definition 4. *Let $\mathcal{A}$ be an ABox and $\mathcal{B}$ its abstraction as in Definition 3. The abstraction $\mathcal{B}$* induces *a mapping $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ such that:*

$$h(v_\tau) \in \{a \in \mathsf{ind}(\mathcal{A}) \mid \tau(a) = \tau\},$$
$$h(w^R_\tau) \in \{b \in \mathsf{ind}(\mathcal{A}) \mid R(h(v_\tau), b) \in \mathcal{A}\}.$$

Lemma 2. *Let $\mathcal{A}$ be an ABox, $\mathcal{B}$ the abstraction of $\mathcal{A}$. Then, for every mapping h induced by $\mathcal{B}$, h is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$.*

Proof. The mapping h is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$ since, for every $C(v_\tau) \in \mathcal{B}$, we have $h(C(v_\tau)) = C(a) \in \mathcal{A}$ and, for every $R(v_\tau, w^R_\tau) \in \mathcal{B}$, we have $h(R(v_\tau, w^R_\tau)) = R(a, b) \in \mathcal{A}$ for some a, b. □

Once the abstract ABox $\mathcal{B}$ of the original ABox $\mathcal{A}$ has been constructed, instead of performing reasoning over $\mathcal{A}$, we perform reasoning over $\mathcal{B}$ and transfer entailments from the abstraction back to the original ABox using Corollary 1. Intuitively, for each type τ, the abstract individual v_τ is the representative for all individuals of this type. Therefore, for every TBox $\mathcal{T}$ and each $A(v_\tau)$ entailed by $\mathcal{B} \cup \mathcal{T}$, we obtain $A(a)$, where $\tau(a) = \tau$ and $A(a)$, is entailed by $\mathcal{A} \cup \mathcal{T}$. This gives rise to a procedure for computing the concept materialization of an ontology, which we present in Algorithm 1.

Since materializing an inconsistent ontology would extend the ABox with all possible assertions for the atomic concepts and roles, and individuals used in the ontology, we can furthermore observe that $\mathcal{B} \cup \mathcal{T}$ can also be used to check consistency of $\mathcal{A} \cup \mathcal{T}$.

Algorithm 1 Procedure for computing the concept materialization of an ontology

Input: An ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$
Output: Returns the concept materialized ontology $\mathcal{O}$

```
Compute the abstraction B of A according to Definition 3
Compute the concept materialization B' ∪ T of B ∪ T
ΔB = {A(v_τ) ∈ B' | A(v_τ) ∉ B}
for all A(v_τ) ∈ ΔB do
    for all a ∈ ind(A) s.t. τ(a) = τ do
        A = A ∪ {A(a)}
    end for
end for
return A ∪ T
```

Algorithm 2 Procedure for checking consistency of an ontology

Input: An ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$
Output: Returns *true* if $\mathcal{O}$ is consistent and *false* otherwise
1: Compute the abstraction $\mathcal{B}$ of $\mathcal{A}$ according to Definition 3
2: **if** $\mathcal{B} \cup \mathcal{T}$ is inconsistent **then**
3: **return** *false*
4: **else**
5: **return** *true*
6: **end if**

We use this to devise a procedure for checking consistency of an ontology in Algorithm 2. In practice the steps performed by this algorithm can also be directly integrated into Algorithm 1.

Soundness of the algorithms follows directly from our previously shown results.

Lemma 3 (Soundness). *Let $\mathcal{A}$ be an ABox, $\mathcal{B}$ its abstraction, and $\mathcal{T}$ a TBox. Then, we have:*

(1) $\mathcal{B} \cup \mathcal{T}$ is inconsistent implies $\mathcal{A} \cup \mathcal{T}$ is inconsistent;
(2) for every type τ and every concept C, $\mathcal{B} \cup \mathcal{T} \models C(v_\tau)$ implies $\mathcal{A} \cup \mathcal{T} \models C(a)$, where $a \in \mathsf{ind}(\mathcal{A})$ s.t. $\tau(a) = \tau$.

Proof. The lemma is a straightforward consequence of the fact that the abstraction induces homomorphisms to the original ABox according to Lemma 2. Applying the contrapositive of Corollary 2 and Corollary 1 yields the desired result and, hence, soundness of the algorithms. □

Example 6. Consider $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ with $\mathcal{T} = \{A \sqsubseteq C, \exists R^- \sqsubseteq B\}$ from Example 2 and $\mathcal{A}$ from Example 1 as input for Algorithm 1. Figure 2 visualizes $\mathcal{A}$ and its abstraction $\mathcal{B}$. By materializing $\mathcal{B} \cup \mathcal{T}$, we obtain $\Delta\mathcal{B} = \{C(v_{\tau(a)}),\ C(v_{\tau(b)}),\ B(v_{\tau(b)})\}$ in Line 3. Note that while $B(w^R_{\tau(a)})$ is in the materialized abstraction $\mathcal{B}'$, it is not part of $\Delta\mathcal{B}$. By updating $\mathcal{A}$ using $\Delta\mathcal{B}$ (Lines 4 to 8), we obtain $\mathcal{A} = \mathcal{A} \cup \{C(a),\ C(b),\ B(b)\}$, where all added concept assertions are entailed by the original ontology.

The procedure in Algorithm 1 differs from the abstraction refinement procedure in the existing approach for Horn $\mathcal{ALCHOI}$ [7] in that, for each type τ, only assertions of v_τ are used to update the original ABox. As demonstrated in Example 6, although $B(w^R_{\tau(a)}) \in \mathcal{B}'$, it is not in $\Delta\mathcal{B}$ and, hence, it is not used for extending $\mathcal{A}$. In addition, unlike the algorithm in the existing approach, Algorithm 1 incorporates no refinement step, i.e. there is no repetition of Lines 1–8 until no new assertions can be added to the original ABox $\mathcal{A}$. Such a repetition is required to obtain completeness for the Horn $\mathcal{ALCHOI}$ procedure. We next show that the current procedure is nevertheless *complete* for DL-Lite$^{\mathcal{H}\sqcup}_{core}$, that is, the resulting ontology is (concept) materialized when the procedure terminates.

We can immediately show soundness of the algorithms as there always exist homomorphisms from the abstraction $\mathcal{B}$ to the corresponding original ABox $\mathcal{A}$ as in Definition 4. But we do not have a similar property for completeness, i.e. there might exist no

homomorphism from $\mathcal{A}$ to $\mathcal{B}$. To show completeness, we construct an extension of $\mathcal{B}$ such that there exists a homomorphism from $\mathcal{A}$ to the extension that maps a to $v_{\tau(a)}$ for each individual $a \in \mathsf{ind}(\mathcal{A})$; and we show that the abstraction entails exactly the same concept assertions as its extension does.

Example 7 (Example 6 continued). Let $\mathcal{B}^+$ be an ABox obtained from $\mathcal{B}$ in Example 6 by adding the role assertion $R(v_{\tau(a)}, v_{\tau(b)})$, and h a mapping from $\mathcal{A}$ to $\mathcal{B}^+$ defined as $h(a) = v_{\tau(a)}, h(b) = v_{\tau(b)}$. Since $h(\mathcal{A}) = \{A(v_{\tau(a)}), A(v_{\tau(b)}), R(v_{\tau(a)}, v_{\tau(b)})\} \subseteq \mathcal{B}^+$, h is a homomorphism from $\mathcal{A}$ to $\mathcal{B}^+$. Therefore, using Corollary 1, we can obtain all entailed assertions of a and b based on entailed assertions of $v_{\tau(a)}$ and $v_{\tau(b)}$ w.r.t. $\mathcal{B}^+ \cup \mathcal{T}$. Furthermore, $\mathcal{B}^+ \cup \mathcal{T}$ and $\mathcal{B} \cup \mathcal{T}$ entail the same set of concept assertions. Hence, the abstraction $\mathcal{B}$ is sufficient for obtaining all entailed assertions of $\mathcal{A} \cup \mathcal{T}$. Indeed, the ABox $\mathcal{A}$ after updating already contains all entailed concept assertions.

As demonstrated in Example 7, for this particular TBox and ABox, the abstraction is sufficient to obtain all entailed concept assertions of the original ontology. In the following lemma, we show that the same property holds for any TBox and ABox.

Lemma 4. *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be a* DL-Lite$_{core}^{\mathcal{H}\sqcup}$ *ontology, $\mathcal{B}$ the abstraction of $\mathcal{A}$, and $\mathcal{B}^+ = \mathcal{B} \cup \{R(v_{\tau(a)}, v_{\tau(b)}) \mid R(a,b) \in \mathcal{A}\}$. Then, we have:*

(1) $\mathcal{B} \cup \mathcal{T}$ is consistent implies $\mathcal{B}^+ \cup \mathcal{T}$ is consistent;
(2) for every atomic concept A and individual v, $\mathcal{B}^+ \cup \mathcal{T} \models A(v)$ implies $\mathcal{B} \cup \mathcal{T} \models A(v)$.

Proof. If $\mathcal{B} \cup \mathcal{T}$ is inconsistent, then the lemma trivially holds. We assume $\mathcal{B} \cup \mathcal{T}$ is consistent and let $\mathcal{I}$ be an arbitrary model of $\mathcal{B} \cup \mathcal{T}$. Next, we construct a model $\mathcal{J}$ of $\mathcal{B}^+ \cup \mathcal{T}$ such that $\mathcal{J} \models A(v)$ implies $\mathcal{I} \models A(v)$ for every atomic concept A and individual v. Then it follows that $\mathcal{B}^+ \cup \mathcal{T}$ is consistent, i.e. Claim (1) holds, and $\mathcal{B}^+ \cup \mathcal{T} \models A(v)$ implies $\mathcal{J} \models A(v)$, which implies $\mathcal{I} \models A(v)$. Since $\mathcal{I}$ is arbitrary, we obtain $\mathcal{B} \cup \mathcal{T} \models A(v)$, i.e. Claim (2) holds. Such a model $\mathcal{J}$ is obtained from $\mathcal{I}$ by setting $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$ and defining the interpretation function as follows:

$$
\begin{aligned}
v^{\mathcal{J}} &= v^{\mathcal{I}} \text{ for every individual } v \\
A^{\mathcal{J}} &= A^{\mathcal{I}} \text{ for every atomic concept } A \\
r^{\mathcal{J}} &= r^{\mathcal{I}} \cup \{\langle v^{\mathcal{I}}_{\tau(a)}, v^{\mathcal{I}}_{\tau(b)}\rangle \mid R(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}^+ \text{ and } \mathcal{O} \models R \sqsubseteq r\} \\
&\quad \cup \{\langle v^{\mathcal{I}}_{\tau(b)}, v^{\mathcal{I}}_{\tau(a)}\rangle \mid R(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}^+ \text{ and } \mathcal{O} \models R \sqsubseteq r^-\} \\
&\quad \text{for every atomic role } r
\end{aligned}
$$

We will show $\mathcal{J} \models \mathcal{B}^+ \cup \mathcal{T}$ by showing that it entails every axiom in $\mathcal{B}^+ \cup \mathcal{T}$. Since $\mathcal{I} \models \mathcal{B} \cup \mathcal{T}$ and the interpretation of atomic concepts and individuals remains the same in $\mathcal{J}$, we have $\mathcal{J}$ entails every concept assertion in $\mathcal{B}^+$. And, clearly, from the definition of $\mathcal{J}$, it follows that $\mathcal{J}$ entails every role assertion in $\mathcal{B}^+$.

We now show by induction that, for every DL-Lite$_{core}^{\mathcal{H}\sqcup}$ concept C, we have $C^{\mathcal{J}} = C^{\mathcal{I}}$. Then, for every concept inclusion $C \sqsubseteq D \in \mathcal{T}$, we have $C^{\mathcal{J}} = C^{\mathcal{I}} \subseteq D^{\mathcal{I}} = D^{\mathcal{J}}$, i.e. $\mathcal{J} \models C \sqsubseteq D$.

- Cases $C = A \mid \neg A \mid \top \mid \bot$ are trivial as the interpretation of atomic concepts in $\mathcal{I}$ and in $\mathcal{J}$ are identical.

- Case $C = \exists r$, where $r \in N_R$; the case $\exists r^-$ is symmetric. We have $d \in (\exists r)^{\mathcal{J}}$ iff there exists $e \in \Delta^{\mathcal{J}}$ s.t. $\langle d, e\rangle \in r^{\mathcal{J}}$. If $\langle d, e\rangle \in r^{\mathcal{I}}$, then $d \in (\exists r)^{\mathcal{I}}$. Otherwise, from the definition of $\mathcal{J}$, $\langle d, e\rangle$ results from one of the cases in the role extension. We consider the case $d = v^{\mathcal{I}}_{\tau(a)}, e = v^{\mathcal{I}}_{\tau(b)}$ for some individuals a and b, where $R(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}^+, \mathcal{O} \models R \sqsubseteq r, \mathcal{O} \not\models R \sqsubseteq r^-$; other cases are analogous. By definition of $\mathcal{B}^+$, we have $R(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}^+$ iff $R(a, b) \in \mathcal{A}$. This is the case iff $R(v_{\tau(a)}, w^R_{\tau(a)}) \in \mathcal{B}$ by Definition 3. Since $\mathcal{O} \models R \sqsubseteq r$ and $\mathcal{I} \models \mathcal{B}$, we obtain $\langle v^{\mathcal{I}}_{\tau(a)}, (w^R_{\tau(a)})^{\mathcal{I}}\rangle \in r^{\mathcal{I}}$, i.e. $d = v^{\mathcal{I}}_{\tau(a)} \in (\exists r)^{\mathcal{I}}$. Since d is arbitrary, we have $(\exists r)^{\mathcal{J}} = (\exists r)^{\mathcal{I}}$.
- Case $C = \neg D$. By induction hypothesis $D^{\mathcal{J}} = D^{\mathcal{I}}$ and since $\Delta^{\mathcal{J}} = \Delta^{\mathcal{I}}$, we have $(\neg D)^{\mathcal{J}} = \Delta^{\mathcal{J}} \setminus D^{\mathcal{J}} = \Delta^{\mathcal{I}} \setminus D^{\mathcal{I}} = (\neg D)^{\mathcal{I}}$, i.e. $C^{\mathcal{J}} = C^{\mathcal{I}}$.
- Cases $C = C_1 \sqcup C_2$ and $C = C_1 \sqcap C_2$. By induction hypothesis, we have $C_1^{\mathcal{J}} = C_1^{\mathcal{I}}$ and $C_2^{\mathcal{J}} = C_2^{\mathcal{I}}$. Therefore, $(C_1 \sqcup C_2)^{\mathcal{J}} = C_1^{\mathcal{J}} \cup C_2^{\mathcal{J}} = C_1^{\mathcal{I}} \cup C_2^{\mathcal{I}} = (C_1 \sqcup C_2)^{\mathcal{I}}$. Similarly, we obtain $(C_1 \sqcap C_2)^{\mathcal{J}} = (C_1 \sqcap C_2)^{\mathcal{I}}$.

For every role inclusion $R \sqsubseteq S \in \mathcal{T}$, by the definition of $\mathcal{J}$ and from $\mathcal{I} \models R \sqsubseteq S$, we have $\mathcal{J} \models R \sqsubseteq S$, which proves $\mathcal{J} \models \mathcal{B}^+ \cup \mathcal{T}$ and, hence, finishes this proof. □

Using Lemma 4, we can establish completeness of Algorithm 1 and Algorithm 2.

Lemma 5 (Completeness). *Let $\mathcal{A}$ be an ABox, $\mathcal{B}$ its abstraction, and $\mathcal{T}$ a* DL-Lite$^{\mathcal{H}\sqcup}_{core}$ *TBox, then we have:*

(1) $\mathcal{B} \cup \mathcal{T}$ is consistent implies $\mathcal{A} \cup \mathcal{T}$ is consistent;
(2) for every atomic concept A and individual a, $\mathcal{A} \cup \mathcal{T} \models A(a)$ implies $\mathcal{B} \cup \mathcal{T} \models A(v_{\tau(a)})$.

Proof. Let $\mathcal{B}^+$ be the ABox in Lemma 4 and h a mapping from $\mathcal{A}$ to $\mathcal{B}^+$ s.t. $h(a) = v_{\tau(a)}$, for every $a \in \mathsf{ind}(\mathcal{A})$. By the definitions of $\mathcal{B}$ and of $\mathcal{B}^+$, for each $A(a) \in \mathcal{A}$, we have $A(v_{\tau(a)}) \in \mathcal{B}$, which implies $A(v_{\tau(a)}) \in \mathcal{B}^+$. By the definition of $\mathcal{B}^+$, for each $R(a, b) \in \mathcal{A}$, we have $R(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}^+$. Hence, h is a homomorphism from $\mathcal{A}$ to $\mathcal{B}^+$. By Claim (1) of Lemma 4 and Corollary 2, consistency of $\mathcal{B} \cup \mathcal{T}$ implies consistency of $\mathcal{B}^+ \cup \mathcal{T}$, which implies consistency of $\mathcal{A} \cup \mathcal{T}$, i.e. Claim (1) holds. Similarly, by Corollary 1 and Claim (2) of Lemma 4, we have, for each atomic concept A and individual a, $\mathcal{A} \cup \mathcal{T} \models A(a)$ implies $\mathcal{B}^+ \cup \mathcal{T} \models h(A(a))$. Since $h(A(a)) = A(v_{\tau(a)})$, this implies $\mathcal{B} \cup \mathcal{T} \models A(v_{\tau(a)})$ and Claim (2) holds. □

5 Implementation and Evaluation

We have implemented a prototype system Orar[2] for reasoning in DL-Lite$^{\mathcal{H}\sqcup}_{core}$. To evaluate the feasibility of our approach, we tested Orar on several real-life and benchmark ontologies and compared the performance of Orar with that of the other popular reasoners. The empirical evaluation results show the approach can reduce the size of the ABoxes significantly (by orders of magnitude), which results in great performance improvements.

[2] `https://github.com/kieen/OrarHSHOIF`

Table 2. Test ontologies with the number of TBox axioms (# ax.), atomic concepts (# con.), roles (# rol.), individuals (# ind.), concept and role assertions (# ast.), inferred assertions (# inferred ast.) by our system

Ontology	# ax.	# con.	# rol.	# indiv.	# assert.	# inferred assert.
NPD	354	208	90	785 656	1 392 196	1 517 844
DBPedia$^+$	1 748	442	806	3 822 351	27 094 909	30 239 281
IMDb	131	88	39	6 505 584	27 757 894	33 769 170
LUBM 10	80	43	25	207 426	850 433	1 086 472
LUBM 50	80	43	25	1 082 818	4 445 949	5 676 226
LUBM 100	80	43	25	2 179 766	8 954 615	11 434 996
LUBM 500	80	43	25	10 847 183	44 573 624	56 914 960
UOBM 10	110	69	35	242 491	1 926 897	2 324 962
UOBM 50	110	69	35	1 227 123	9 751 681	11 768 772
UOBM 100	110	69	35	2 461 347	19 571 755	23 617 264
UOBM 500	110	69	35	12 375 804	98 374 692	118 717 591

The test ontologies are from popular benchmarks and also used in the evaluations of other approaches. NPD[3] is an ontology about petroleum activities, DBPedia$^+$[4] is an extension of the DBPedia ontology, and IMDb[5] consists of the Movie ontology and the dataset extracted from the IMDb website. While NPD, DBPedia$^+$, and IMDb contain real-life data, LUBM and UOBM are popular benchmarks with synthetic data of the university domain. The datasets in LUBM and UOBM can be generated in arbitrary sizes, indicated by the number of universities. We use LUBM n and UOBM n to denote the datasets for n universities of LUBM and UOBM, respectively. We extracted the relevant DL fragment from those ontologies, i.e. we eliminated axioms not in $\mathrm{DL\text{-}Lite}_{core}^{\mathcal{H}\sqcup}$. Table 2 presents detailed information about the test ontologies with the number of TBox axioms, atomic concepts, roles, individuals, and (inferred) assertions. NPD, IMDb, and LUBM are in $\mathrm{DL\text{-}Lite}_{core}^{\mathcal{H}}$ while DBPedia$^+$ and UOBM are in $\mathrm{DL\text{-}Lite}_{core}^{\mathcal{H}\sqcup}$.

We used Orar to check consistency and compute the concept materialization of the test ontologies and compared the reasoning time of Orar and of the other well-known reasoners HermiT 1.3.8, JFact 5.0.0, Pellet 2.3.6, and Konclude 0.6.2. All tests were run on an Intel Xeon E5-2660V3 2.60GHz machine with 250 GB heap size for the Java VM and with a timeout of five hours. Table 3 presents information about the abstractions and the size of the abstract ABoxes in comparison with the size of the original ABoxes. In NPD, IMDb, and LUBM, many individuals have the same types. For those ontologies, the size of the original ABoxes are reduced by up to four orders of magnitude. Particularly, for LUBM the abstract ABoxes are of nearly constant size regardless of the sizes of the original ABoxes. This can be explained by the simple patterns used to generate data in LUBM. The individuals in DBPedia$^+$ and UOBM are more diverse. For DBPedia$^+$, the number of types is relatively large due to the large number of concepts and roles; the size of the abstract ABox is approximately 10% of the original one.

[3] `http://sws.ifi.uio.no/project/npd-v2`

[4] `https://www.cs.ox.ac.uk/isg/tools/PAGOdA`

[5] `https://sites.google.com/site/ontopiswc13/home/imdb-mo`

Table 3. Number of types, abstract individuals, assertions, and size of the abstract ABox in comparison with the original ABox

Ontology	Abstraction			% of Original ABox	
	# types	# indiv.	# assert.	% indiv.	% assert.
NPD	1 005	15 580	18 244	1.983	1.310
DBPedia$^+$	226 530	1 775 630	2 770 261	46.454	10.224
IMDb	438	1 224	1 692	0.019	0.006
LUBM 10	29	154	158	0.074	0.019
LUBM 50	27	148	152	0.014	0.003
LUBM 100	27	148	152	0.007	0.002
LUBM 500	27	148	152	0.001	0.001
UOBM 10	11 391	97 944	124 661	40.391	6.470
UOBM 50	25 541	225 420	289 762	18.370	2.971
UOBM 100	34 872	310 513	400 938	12.616	2.049
UOBM 500	64 903	593 539	769 843	4.796	0.783

Table 4. Reasoning time (without loading time) in seconds, where "−" stands for timeout

Ontology	Concept Materialization					Consistency Checking				
	Orar	Konclude	Pellet	HermiT	JFact	Orar	Konclude	Pellet	HermiT	JFact
NPD	5	11	39	579	−	3	8	27	284	−
DBPedia$^+$	163	176	631	2 029	−	48	148	417	292	−
IMDb	34	220	775	983	−	30	7	684	568	−
LUBM 10	2	4	9	9	3 651	2	1	8	6	2 606
LUBM 50	10	28	53	61	−	10	2	52	34	−
LUBM 100	18	67	149	133	−	16	3	135	94	−
LUBM 500	90	359	1 601	979	−	80	10	1 476	642	−
UOBM 10	8	18	23	−	−	3	16	16	47	3 073
UOBM 50	25	106	148	−	−	13	90	115	624	−
UOBM 100	42	227	353	−	−	23	187	274	1 421	−
UOBM 500	160	1 636	3 846	−	−	117	1 225	3 287	−	−

For UOBM, the sizes of the abstract ABoxes are approximately 6% and 1% of the sizes of the original ones for UOBM 10 and UOBM 500, respectively. Table 4 shows the reasoning time of Orar (with Konclude as the internal reasoner for the abstraction) in comparison with the reasoning time of the other reasoners. In general, the reasoning time correlates with the size reduction of the ontologies. For concept materialization, Orar outperforms the other reasoners on all ontologies. For consistency checking, Konclude is faster than Orar for IMDb and LUBM. The reason is that reasoning on those ontologies is even faster than other operations required in Orar like computing types and generating the abstract ABoxes. For the other ontologies, Orar outperforms all reasoners. Note that the purpose of our evaluation was not to show the superiority of Orar, but to demonstrate that our approach can improve the performance of any existing reasoner when handling large data. Although we used Konclude inside Orar, it can be replaced by any reasoner.

6 Extensions and Variations

In this section, we discuss the extension of the presented approach to DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ and present a variation of the abstract ABox, which can also be used in our approach.

6.1 Reasoning with Nominals

Since Lemma 1 even holds for the very expressive language $\mathcal{SROIQ}$, Algorithm 1 and Algorithm 2 are sound for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$. Before we show that they are also complete for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$, we first illustrate the advantage of using the abstraction-based approach for classification of DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ ontologies. This reasoning task has not been covered so far as for DL-Lite$_{core}^{\mathcal{H}\sqcup}$ classification requires reasoning only over the TBox (after checking consistency of the ontology).

Classification of an ontology containing nominals requires reasoning over both TBox and ABox. This even holds for rather simple languages such as DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ and even OWL 2 RL where nominals can only occur in a restricted form ($\exists R.o$) [14]. The following example demonstrates that class subsumption between two concepts might depend on the existence of some assertions.

Example 8. Consider a TBox $\mathcal{T} = \{A \sqsubseteq o, \exists R^- \sqsubseteq o, \exists R^- \sqsubseteq B, C \sqsubseteq \exists R\}$. We observe that $A \sqsubseteq B$ holds depending on the existence of instances of the role R, which can be enforced if C has some instances. Indeed, consider $\mathcal{A} = \{C(a)\}$, we have $\mathcal{A} \cup \mathcal{T} \models A \sqsubseteq B$. In any interpretation $\mathcal{I}$ with $A^{\mathcal{I}} = \emptyset$ the subsumption trivially holds. If, however, there is some element $d \in A^{\mathcal{I}}$, we show that d must also be in $B^{\mathcal{I}}$. By $A \sqsubseteq o$, we get $d \in o^{\mathcal{I}}$. Since $C(a) \in \mathcal{A}$, $C \sqsubseteq \exists R \in \mathcal{T}$, and $a^{\mathcal{I}} \in C^{\mathcal{I}}$, there is some d' such that $\langle a^{\mathcal{I}}, d' \rangle \in R^{\mathcal{I}}$. Since $\exists R^- \sqsubseteq o, \exists R^- \sqsubseteq B \in \mathcal{T}$, $d' \in o^{\mathcal{I}} \cap B^{\mathcal{I}}$ and, since o is a nominal concept, we have $d = d' \in A^{\mathcal{I}} \cap B^{\mathcal{I}}$ and the subsumption also holds. It is easy to see, however, that $\emptyset \cup \mathcal{T} \not\models A \sqsubseteq B$.

Since the abstractions are often smaller than the original ABoxes, classification over the abstraction will be more efficient than classification over the original ontology.

Lemma 6. *Let $\mathcal{A} \cup \mathcal{T}$ be a* DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ *ontology and $\mathcal{B}$ the abstraction of $\mathcal{A}$. Then, for every atomic concepts $A, B \in \mathsf{con}(\mathcal{A} \cup \mathcal{T})$, $\mathcal{A} \cup \mathcal{T} \models A \sqsubseteq B$ iff $\mathcal{B} \cup \mathcal{T} \models A \sqsubseteq B$.*

By Lemma 1 and Lemma 2, the "only-if" direction of Lemma 6 holds. We now briefly show the "if" direction of Lemma 6; and also show that Algorithm 1 and Algorithm 2 are complete for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$. We rely on the following extension of Lemma 4.

Lemma 7. *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be a* DL-Lite$_{core}^{\mathcal{HO}\sqcup}$ *ontology, $\mathcal{B}$ the abstraction of $\mathcal{A}$, and $\mathcal{B}^+ = \mathcal{B} \cup \{R(v_{\tau(a)}, v_{\tau(b)}) \mid R(a,b) \in \mathcal{A}\}$. Then, we have:*

(1) $\mathcal{B} \cup \mathcal{T}$ is consistent implies $\mathcal{B}^+ \cup \mathcal{T}$ is consistent;
(2) for every atomic concept A and individual v, $\mathcal{B}^+ \cup \mathcal{T} \models A(v)$ implies $\mathcal{B} \cup \mathcal{T} \models A(v)$; and
(3) for every atomic concepts A and B, $\mathcal{B}^+ \cup \mathcal{T} \models A \sqsubseteq B$ implies $\mathcal{B} \cup \mathcal{T} \models A \sqsubseteq B$.

Proof (Sketch). Intuitively, we follow similar steps as in the proof of Lemma 4. The only difference is to extend the interpretations to cover nominals. Reconsider the interpretations $\mathcal{I}$ and $\mathcal{J}$ in the proof of Lemma 4. We define $\mathcal{J}$ as before and let the interpretations of nominals in $\mathcal{J}$ and in $\mathcal{I}$ be identical. Then, all claims in the existing proof remain sound. Furthermore, since the interpretations of atomic concepts in $\mathcal{I}$ and in $\mathcal{J}$ are identical, we have $\mathcal{B}^+ \cup \mathcal{T} \models A \sqsubseteq B$ implies $A^{\mathcal{J}} \subseteq B^{\mathcal{J}}$, which implies $A^{\mathcal{I}} \subseteq B^{\mathcal{I}}$, i.e. $\mathcal{I} \models A \sqsubseteq B$. Since $\mathcal{I}$ is arbitrary, we have $\mathcal{B} \cup \mathcal{T} \models A \sqsubseteq B$. □

As shown in the proof of Lemma 5, there exists a homomorphism h from $\mathcal{A}$ to $\mathcal{B}^+$ that maps a to $v_{\tau(a)}$ for each $a \in \mathsf{ind}(\mathcal{A})$. By Lemma 7 and Lemma 1, it follows that the "if" direction of the Lemma 6 holds and that Lemma 5 holds also for DL-Lite$_{core}^{\mathcal{HO}\sqcup}$.

6.2 Alternative Abstraction

The key idea of the abstraction-based approach is to build a suitable, ideally small abstract ABox, which can be used to obtain sound and complete entailments of the input ontology. The abstract ABoxes from Definition 3 induce homomorphisms to the original ABox but not necessarily vice versa. This directly guarantees soundness but not completeness. Completeness of the approach is guaranteed by Lemma 4 and Lemma 5, which show that the abstract ABox $\mathcal{B}$ entails exactly the same assertions as its extension $\mathcal{B}^+$, to which there is a homomorphism from the original ABox. This suggests an alternative definition of abstractions similar to the extension $\mathcal{B}^+$ of $\mathcal{B}$ in Lemma 4.

Definition 5. *The* abstraction *of an ABox $\mathcal{A}$ is an ABox $\mathcal{C} = \{A(v_{\tau(a)}) \mid A(a) \in \mathcal{A}\} \cup \{R(v_{\tau(a)}, v_{\tau(b)}) \mid R(a,b) \in \mathcal{A}\}$, where $\tau(a)$ and $\tau(b)$ are the types of a and b, respectively, and $v_{\tau(a)}$ and $v_{\tau(b)}$ are a fresh, distinguished* abstract individuals.

Example 9. Consider the ABox $\mathcal{A} = \{A(a), A(b), R(a,b)\}$ as in Example 1. The abstraction of $\mathcal{A}$ by Definition 5 is the ABox $\mathcal{C} = \{A(v_{\tau(a)}), A(v_{\tau(b)}), R(v_{\tau(a)}, v_{\tau(b)})\}$.

In Example 9 there are homomorphisms both from $\mathcal{A}$ to $\mathcal{C}$ and from $\mathcal{C}$ to $\mathcal{A}$; $\mathcal{C}$ is just a copy under renaming of $\mathcal{A}$. Therefore, it is easy to see that using $\mathcal{C}$ as an abstract ABox, we obtain both sound and complete entailments for $\mathcal{A}$ w.r.t. any TBox. In general, there is always a homomorphism from $\mathcal{A}$ to $\mathcal{C}$, e.g. the mapping h defined as $h(a) = v_{\tau(a)}, a \in \mathsf{ind}(\mathcal{A})$, but not necessarily a homomorphism from $\mathcal{C}$ to $\mathcal{A}$. This immediately guarantees completeness of the approach but not soundness. However, based on our previously shown results, we can show that all results we obtained using $\mathcal{C}$ are sound. Let $\mathcal{A}$ be an ABox, $\mathcal{C}$ the abstraction of $\mathcal{A}$ by Definition 5, $\mathcal{B}$ the abstraction of $\mathcal{A}$ by Definition 3, and $\mathcal{B}^+$ the extension of $\mathcal{B}$ defined in Lemma 4. Since $\mathcal{C} \subseteq \mathcal{B}^+$, by monotonicity, for every concept A and individual $v_{\tau(a)} \in \mathsf{ind}(\mathcal{C})$, we obtain that $\mathcal{C} \cup \mathcal{T} \models A(v_{\tau(a)})$ implies $\mathcal{B}^+ \cup \mathcal{T} \models A(v_{\tau(a)})$, which implies $\mathcal{B} \cup \mathcal{T} \models A(v_{\tau(a)})$ by Lemma 4. Furthermore, by Lemma 3 we have $\mathcal{B} \cup \mathcal{T} \models A(v_{\tau(a)})$ implies $\mathcal{A} \cup \mathcal{T} \models A(a)$. Therefore, we have $\mathcal{C} \cup \mathcal{T} \models A(v_{\tau(a)})$ implies $\mathcal{A} \cup \mathcal{T} \models A(a)$.

In Definition 5, the abstract ABox $\mathcal{C}$ uses just one individual v_τ for each type τ, and, therefore, it requires less individuals than the abstract ABox $\mathcal{B}$ in Definition 3. However, for each type τ and each role R occurring in τ there is exactly one role assertion, e.g. $R(v_\tau, w_\tau^R)$, in $\mathcal{B}$, whereas v_τ could have many R-successors/predecessors in $\mathcal{C}$. In our experiment with both types of abstract ABoxes, the abstract ABoxes constructed according to Definition 5 are often larger than the ones using Definition 3.

7 Related Work

Several ontology reasoning techniques have been proposed to handle large data. The RDFox [15] and WebPIE [18] systems utilize parallel computing to perform a rule-based materialization for OWL 2 RL. The PAGOdA system [20] approximates the TBox and then performs OWL 2 RL rules to compute lower-bound and upper-bound entailments, which help to determine entailments for individuals quickly. Wandelt and Möller propose a technique for instance retrieval based on modularization [19]. A closely related work to our approaches is the SHER approach [4]. It merges individuals to obtain a compressed, *summary* ABox, which is then used for (refutation-based) consistency checking or query answering. Since merging is only based on concept assertions, the resulting summary ABox is an over-approximation of the original ABox. Therefore, if the summary ABox is consistent, then so is the original ABox, but not vice versa. In case the summary ABox is inconsistent, explanation techniques [11] are used to repair the summary. In contrast to the summary approach, the abstract ABox created in the presented approach immediately allows for both sound and complete results. Note that for DL-Lite ontologies, the previous abstraction refinement approach [7] performs reasoning over the abstract ABox twice as it continues doing refinement after transferring the entailments from the first abstraction to the original ABox.

8 Conclusions

We have presented a scalable abstraction-based approach for reasoning in $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ and its extensions for $\text{DL-Lite}_{core}^{\mathcal{HO}\sqcup}$. For $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$, we focus on concept materialization and consistency checking of the ontology as computing the role materialization can be done simply by expanding the existing role assertions according to the role hierarchy; and computing the class hierarchy requires only the TBox (after checking consistency of the ontology). For $\text{DL-Lite}_{core}^{\mathcal{HO}\sqcup}$, we show that the presented approach for concept materialization and consistency checking remains sound and complete. Furthermore, it can be easily extended to ontology classification, a non-trivial task in $\text{DL-Lite}_{core}^{\mathcal{HO}\sqcup}$. Computing the role materialization of $\text{DL-Lite}_{core}^{\mathcal{HO}\sqcup}$ ontologies is not as simple as in $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ as role assertions can be derived not only from the role hierarchy but also from axioms of nominals. It is possible to use the abstraction to obtain also the role assertions as presented in our recent work for Horn $\mathcal{SHOIF}$ [8].

The languages we consider in this paper do not make the *Unique Name Assumption* (UNA), which is often adopted in DL-Lite. But the presented approach also works for $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ with UNA; for $\text{DL-Lite}_{core}^{\mathcal{HO}\sqcup}$, it does not make sense to adopt UNA. As noted in the work about different dialects of the DL-Lite family [1], we can construct a model for a $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ ontology with UNA from a model of that ontology without UNA by "cloning" the domain elements so that different individuals are interpreted differently. Entailments are preserved in the resulting model, therefore, the results for $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ without UNA remain valid in $\text{DL-Lite}_{core}^{\mathcal{H}\sqcup}$ with UNA.

References

1. Artale, A., Calvanese, D., Kontchakov, R., Zakharyaschev, M.: The DL-Lite family and relations. J. Artificial Intelligence Research 36, 1–69 (2009)
2. Bienvenu, M., Kikot, S., Kontchakov, R., Podolskii, V.V., Zakharyaschev, M.: Theoretically optimal datalog rewritings for OWL 2 QL ontology-mediated queries. In: Proc. of the 29th Int. Workshop on Description Logics (DL 2016) (2016)
3. Calvanese, D., De Giacomo, G., Lembo, D., Lenzerini, M., Rosati, R.: Tractable reasoning and efficient query answering in description logics: The DL-Lite family. J. of Automated Reasoning 39(3), 385–429 (2007)
4. Dolby, J., Fokoue, A., Kalyanpur, A., Schonberg, E., Srinivas, K.: Scalable highly expressive reasoner (SHER). J. of Web Semantics 7(4), 357–361 (2009)
5. Eiter, T., Ortiz, M., Simkus, M., Tran, T.K., Xiao, G.: Query rewriting for Horn-$\mathcal{SHIQ}$ plus rules. In: Proc. of the 26th National Conf. on Artificial Intelligence (AAAI 2012) (2012)
6. Feier, C., Carral, D., Stefanoni, G., Cuenca Grau, B., Horrocks, I.: The combined approach to query answering beyond the OWL 2 profiles. In: Proc. of the 24th Int. Joint Conf. on Artificial Intelligence (IJCAI 2015). pp. 2971–2977 (2015)
7. Glimm, B., Kazakov, Y., Liebig, T., Tran, T., Vialard, V.: Abstraction refinement for ontology materialization. In: Proc. of the 13th Int. Semantic Web Conf. (ISWC 2014). pp. 180–195. Springer (2014)
8. Glimm, B., Kazakov, Y., Tran, T.: Ontology materialization by abstraction refinement in Horn $\mathcal{SHOIF}$. In: Proc. of the 29th Int. Workshop on Description Logics (DL 2016) (2016)
9. Gottlob, G., Orsi, G., Pieris, A.: Query rewriting and optimization for ontological databases. ACM Transactions on Database Systems 39(3), 25:1–25:46 (2014)
10. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible $\mathcal{SROIQ}$. In: Proc. of 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006). pp. 57–67. AAAI Press (2006)
11. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Proc. of the 6th Int. Semantic Web Conf. (ISWC 2007) (2007)
12. Kikot, S., Kontchakov, R., Podolskii, V.V., Zakharyaschev, M.: Exponential lower bounds and separation for query rewriting. In: Proc. of the 39th Int. Colloquium on Automata, Languages, and Programming (ICALP 2012). vol. 7392, pp. 263–274. Springer (2012)
13. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to query answering in DL-Lite. In: Proc. of the 12th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2010). AAAI Press (2010)
14. Krötzsch, M.: The not-so-easy task of computing class subsumptions in OWL RL. In: Proc. of the Int. Semantic Web Conf. (ISWC 2012). vol. 7649, pp. 279–294. Springer (2012)
15. Motik, B., Nenov, Y., Piro, R., Horrocks, I., Olteanu, D.: Parallel materialisation of datalog programs in centralised, main-memory RDF systems. In: Proc. of the 28th National Conf. on Artificial Intelligence (AAAI 2014). pp. 129–137 (2014)
16. Pérez-Urbina, H., Motik, B., Horrocks, I.: Tractable query answering and rewriting under description logic constraints. J. Applied Logic 8(2), 186–209 (2010)
17. Trivela, D., Stoilos, G., Chortaras, A., Stamou, G.: Optimising resolution-based rewriting algorithms for OWL ontologies. J. Web Semantics 33, 30–49 (2015)
18. Urbani, J., Kotoulas, S., Maassen, J., van Harmelen, F., Bal, H.E.: WebPIE: A web-scale parallel inference engine using MapReduce. J. Web Semantics 10, 59–75 (2012)
19. Wandelt, S., Möller, R.: Towards ABox modularization of semi-expressive description logics. J. of Applied Ontology 7(2), 133–167 (2012)
20. Zhou, Y., Cuenca Grau, B., Nenov, Y., Kaminski, M., Horrocks, I.: PAGOdA: Pay-as-you-go ontology query answering using a datalog reasoner. J. Artificial Intelligence Research 54, 309–367 (2015)