Abstraction Refinement for Ontology Materialization

Birte Glimm[1], Yevgeny Kazakov[1], Thorsten Liebig[2], Trung-Kien Tran[1], and Vincent Vialard[2]

[1] University of Ulm, Ulm, Germany, <first name>.<last name>@uni-ulm.de
[2] derivo GmbH, Ulm, Germany, <lastname>@derivo.de

Abstract. We present a new procedure for ontology materialization (computing all entailed instances of every atomic concept) in which reasoning over a large ABox is reduced to reasoning over a smaller "abstract" ABox. The abstract ABox is obtained as the result of a fixed-point computation involving two stages: 1) abstraction: partition the individuals into equivalence classes based on told information and use one representative individual per equivalence class, and 2) refinement: iteratively split (refine) the equivalence classes, when new assertions are derived that distinguish individuals within the same class. We prove that the method is complete for Horn $\mathcal{ALCHOI}$ ontologies, that is, all entailed instances will be derived once the fixed-point is reached. We implement the procedure in a new database-backed reasoning system and evaluate it empirically on existing ontologies with large ABoxes. We demonstrate that the obtained abstract ABoxes are significantly smaller than the original ones and can be computed with few refinement steps.

1 Introduction

Ontology based data access (OBDA) is an increasingly popular paradigm in the area of knowledge representation and information systems. In ODBA, a TBox with background knowledge is used to enrich and integrate large, incomplete, and possibly semi-structured data, which users can then access via queries. Different approaches have been developed to compute the entailed answers to a query: 1) Query rewriting approaches answer a query by "compiling" the background knowledge of the TBox into the query [2,11]. 2) Materialization techniques take the opposite approach by pre-computing all entailed information upfront, independent of the queries [15]. After extending the ABox with all pre-computed facts, the unmodified queries can be evaluated over the enriched data only (i.e., without considering the schema). 3) Recently, also combined approaches have been proposed, which allow for smaller rewritten queries and are applicable for more expressive languages [10,9,4,13].

How queries can be answered also depends on what kinds of queries are allowed. Classical conjunctive queries include existential variables, which can be satisfied also by individuals that do not occur in the input ontology, but whose existence can be inferred. Answering such queries often requires reasoning at query time even for fully materialized ontologies. Many approaches including the entailment regimes of the Semantic Web query language SPARQL [6] are based on the simpler form of conjunctive instance queries. Such queries only involve named individuals from the input ontology

Table 1. The syntax of $\mathcal{ALCHOI}$

Concepts	$C ::= \top \mid \bot \mid A \mid o \mid \neg C \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C \mid \forall R.C$
Roles	$R ::= R \mid R^-$
Axioms	$C_1 \sqsubseteq C_2, R_1 \sqsubseteq R_2, C(a), R(a,b)$

and can be reduced to instance retrieval. Hence, answering such queries requires only look-ups over a materialized ABox. This is also the setting considered in this paper, i.e., our goal is the materialization of entailed facts for large ABoxes stored in a database.

Our work is based on an assumption that individuals with similar asserted types are likely to have the same inferred types. We can group such individuals into equivalence classes and compute the types just for one representative individual. This results in a so-called *abstraction* that contains one representative individual per equivalence class plus representative individuals for its direct role successors and predecessors. Derivations for the latter individuals can be used to split/refine equivalence classes, in which individuals no longer have the same assertions. The procedure is complete when the fixed point is reached, i.e., no new derivations are made and no refinements happen. The number of individuals in the abstraction is bounded exponentially in the number of different concepts and roles and linearly in the size of the original ABox; hence the abstraction is relatively small when the number of individuals is much larger than the number of concepts and roles used in the ontology.

We implement the technique in a database-backed system that interacts with a highly optimized in-memory reasoner that materializes the abstract ABox. The database engine needs to support only simple operations and does not require any knowledge of the TBox. We show that the procedure is sound and it is complete for computing the entailed types of individuals in Horn $\mathcal{ALCHOI}$ ontologies.

After the preliminaries, we present the theoretical foundation for our approach (Section 3) and prove its completeness (Section 4). In Section 5, we evaluate the procedure on a range of real-world ontologies with large ABoxes. We then discuss related work (Section 6) and conclude (Section 7). Full proofs and further details are available in a technical report [5].

2 Preliminaries

We recall the syntax of the Description Logic (DL) $\mathcal{ALCHOI}$ in Table 1; its semantics is defined as usual [1]. For an ontology $\mathcal{O}$, we use $\mathsf{con}(\mathcal{O})$, $\mathsf{rol}(\mathcal{O})$, and $\mathsf{ind}(\mathcal{O})$ as the sets of concept names, role names, and individual names occurring in $\mathcal{O}$, respectively. We say that $\mathcal{O}$ is *concept materialized* if $A(a) \in \mathcal{O}$ whenever $\mathcal{O} \models A(a)$, $A \in \mathsf{con}(\mathcal{O})$, and $a \in \mathsf{ind}(\mathcal{O})$; $\mathcal{O}$ is *role materialized* if $R(a,b) \in \mathcal{O}$ whenever $\mathcal{O} \models R(a,b)$, $R \in \mathsf{rol}(\mathcal{O})$, $a, b \in \mathsf{ind}(\mathcal{O})$; $\mathcal{O}$ is *materialized* if it is both concept and role materialized.

Remark 1. We present nominals as primitive symbols, which are disjoint from individuals. Some definitions use a special *nominal constructor* $\{a\}$ with individual a (in this case, $\{a\}^\mathcal{I} = \{a^\mathcal{I}\}$). We can convert such ontologies to our representation by renaming every nominal $\{a\}$ with the corresponding nominal symbol o_a and adding a concept assertion $o_a(a)$.

3 Computing ABox Materialization by Abstraction

The typical OBDA scenario is such that the ABox contains a large number of individuals. If we can identify individuals that yield the same consequences, we can compute the materialization by computing entailed consequences only for representative individuals.

3.1 Bi-homomorphic Individuals and Individual Types

In order to (syntactically) characterize individuals that yield the same consequences, we study structure-preserving transformations of ABoxes.

Definition 1. *Let $\mathcal{A}$ and $\mathcal{B}$ be two ABoxes and $h\colon \mathsf{ind}(\mathcal{A}) \to \mathsf{ind}(\mathcal{B})$ a mapping from the individuals in $\mathcal{A}$ to individuals in $\mathcal{B}$. We extend h to axioms in a straightforward way: $h(C(a)) = C(h(a))$, $h(R(a,b)) = R(h(a),h(b))$, and $h(\alpha) = \alpha$ for other axioms α. We say that h is a* homomorphism *(from $\mathcal{A}$ to $\mathcal{B}$) if $h(\mathcal{A}) \subseteq \mathcal{B}$. An individual a in $\mathcal{A}$ is* homomorphic *to an individual b in $\mathcal{B}$ if there exists a homomorphism h from $\mathcal{A}$ to $\mathcal{B}$ such that $h(a) = b$; in addition, if b is homomorphic to a, then a and b are* bi-homomorphic.

Example 1. Consider the ABox $\mathcal{A} = \{R(a,a), R(a,b), R(b,b)\}$. Then the mappings $h_1 = \{a \mapsto b,\ b \mapsto b\}$ and $h_2 = \{a \mapsto a,\ b \mapsto a\}$ are homomorphisms from $\mathcal{A}$ to $\mathcal{A}$. Thus, a and b are bi-homomorphic. Note that there is no *isomorphism* h from $\mathcal{A}$ to $\mathcal{A}$ (a bijective homomorphism such that its inverse is also a homomorphism) such that $h(a) = b$ or $h(b) = a$.

It is easy to show that entailed axioms are preserved under homomorphisms between ABoxes. In particular, bi-homomorphic individuals are instances of the same concepts.

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

Proof. We show that Lemma 1 even holds for $\mathcal{SROIQ}$ without unique name assumption. Suppose that $\mathcal{A} \cup \mathcal{T} \models \alpha$. Then $h(\mathcal{A} \cup \mathcal{T}) \models h(\alpha)$. Since $h(\mathcal{A} \cup \mathcal{T}) = h(\mathcal{A}) \cup h(\mathcal{T}) = h(\mathcal{A}) \cup \mathcal{T} \subseteq \mathcal{B} \cup \mathcal{T}$, by monotonicity we obtain $\mathcal{B} \cup \mathcal{T} \models h(\alpha)$. □

Corollary 1. *If individuals a and b in an ABox $\mathcal{A}$ are bi-homomorphic, then for every TBox $\mathcal{T}$ and every concept C, we have $\mathcal{A} \cup \mathcal{T} \models C(a)$ if and only if $\mathcal{A} \cup \mathcal{T} \models C(b)$.*

By Corollary 1, bi-homomorphic individuals entail the same assertions w.r.t. *every* TBox. This property is too strong for our purpose as we need to deal with just one given TBox. It can be that many (non bi-homomorphic) individuals are still materialized in a same way. To take this into account, instead of partitioning the individuals according to the bi-homomorphism relation we start with an approximation to this relation.

Definition 2. *Let $\mathcal{A}$ be an ABox. The* type *of an individual a (w.r.t. $\mathcal{A}$) is a triple $tp(a) = (tp_{\downarrow}(a), tp_{\rightarrow}(a), tp_{\leftarrow}(a))$ where $tp_{\downarrow}(a) = \{C \mid C(a) \in \mathcal{A}\}$, $tp_{\rightarrow}(a) = \{R \mid \exists b : R(a,b) \in \mathcal{A}\}$, and $tp_{\leftarrow}(a) = \{S \mid \exists c : S(c,a) \in \mathcal{A}\}$.*

Intuitively, the type of an individual is defined as a triple consisting of its asserted concepts, successor roles, and predecessor roles in the ABox. Note that bi-homomorphic individuals have the same types, so the relation between individuals of the same types is an approximation to the bi-homomorphism relation.

3.2 Abstraction of an ABox

If we compress the ABox by simply merging all individuals with the same type into one, we might obtain unexpected entailments, even if all individuals are bi-homomorphic.

Example 2. Consider the following ABox $\mathcal{A} = \{R(a, b), R(b, a)\}$. Clearly, a and b are bi-homomorphic in $\mathcal{A}$. Let $\mathcal{B} = \{R(a, a)\}$ be obtained from $\mathcal{A}$ by replacing individual b with a, and let $\mathcal{T} = \{\top \sqsubseteq B \sqcup C,\ \exists R.B \sqsubseteq C\}$. It is easy to check that $\mathcal{B} \cup \mathcal{T} \models C(a)$, but $\mathcal{A} \cup \mathcal{T} \not\models C(a)$ (and hence $\mathcal{A} \cup \mathcal{T} \not\models C(b)$).

Instead of merging all individuals with the same type into one, we realize every individual type in our *abstract ABox*.

Definition 3 (ABox Abstraction). *The* abstraction *of an ABox $\mathcal{A}$ is an ABox $\mathcal{B} = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathcal{B}_{tp(a)}$, where for each type $tp = (tp_{\downarrow}, tp_{\rightarrow}, tp_{\leftarrow})$, we define $\mathcal{B}_{tp} = \{C(x_{tp}) \mid C \in tp_{\downarrow}\} \cup \{R(x_{tp}, y^R_{tp}) \mid R \in tp_{\rightarrow}\} \cup \{S(z^S_{tp}, x_{tp}) \mid S \in tp_{\leftarrow}\}$, where x_{tp}, y^R_{tp}, and z^S_{tp} are fresh distinguished* abstract individuals.

Example 3. Consider the ABox $\mathcal{A} = \{A(a), A(d), R(a, b), R(a, e), R(b, c), R(b, e), R(c, a), R(d, c), R(e, d)\}$. We have $\mathrm{tp}(b) = \mathrm{tp}(c) = \mathrm{tp}(e) = \mathrm{tp}_1 = (\emptyset, \{R\}, \{R\})$ and $\mathrm{tp}(a) = \mathrm{tp}(d) = \mathrm{tp}_2 = (\{A\}, \{R\}, \{R\})$. The abstraction of $\mathcal{A}$ is $\mathcal{B} = \mathcal{B}_{\mathrm{tp}_1} \cup \mathcal{B}_{\mathrm{tp}_2}$ with $\mathcal{B}_{\mathrm{tp}_1} = \{R(x_{\mathrm{tp}_1}, y^R_{\mathrm{tp}_1}), R(z^R_{\mathrm{tp}_1}, x_{\mathrm{tp}_1})\}$, $\mathcal{B}_{\mathrm{tp}_2} = \{A(x_{\mathrm{tp}_2}), R(x_{\mathrm{tp}_2}, y^R_{\mathrm{tp}_2}), R(z^R_{\mathrm{tp}_2}, x_{\mathrm{tp}_2})\}$.

Intuitively, the abstraction of an ABox is a disjoint union of small ABoxes witnessing each individual type realized in the ABox. The following lemma shows the soundness of concept assertions derived from the abstraction.

Lemma 2. *Let $\mathcal{A}$ be an ABox, $\mathcal{B}$ its abstraction, and $\mathcal{T}$ a TBox. Then for every type $tp = (tp_{\downarrow}, tp_{\rightarrow}, tp_{\leftarrow})$, every $a \in \mathsf{ind}(\mathcal{A})$ with $tp(a) = tp$ w.r.t. $\mathcal{A}$, and every concept C:*

(1) $\mathcal{B} \cup \mathcal{T} \models C(x_{tp})$ implies $\mathcal{A} \cup \mathcal{T} \models C(a)$,
(2) $\mathcal{B} \cup \mathcal{T} \models C(y^R_{tp})$ and $R(a, b) \in \mathcal{A}$ implies $\mathcal{A} \cup \mathcal{T} \models C(b)$, and
(3) $\mathcal{B} \cup \mathcal{T} \models C(z^S_{tp})$ and $S(c, a) \in \mathcal{A}$ implies $\mathcal{A} \cup \mathcal{T} \models C(c)$.

Proof. Consider all mappings $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ such that $h(x_{\mathrm{tp}}) \in \{a \in \mathsf{ind}(\mathcal{A}) \mid \mathrm{tp}(a) = \mathrm{tp}\}$, $h(y^R_{\mathrm{tp}}) \in \{b \mid R(h(x_{\mathrm{tp}}), b) \in \mathcal{A}\}$, and $h(z^S_{\mathrm{tp}}) \in \{c \mid S(c, h(x_{\mathrm{tp}})) \in \mathcal{A}\}$. Clearly, $h(\mathcal{B}) \subseteq \mathcal{A}$ for every such mapping h. Furthermore, for every $a \in \mathsf{ind}(\mathcal{A})$, every $R(a, b) \in \mathcal{A}$ and every $S(c, a) \in \mathcal{A}$, there exists h with $h(x_{\mathrm{tp}}) = a$, $h(y^R_{\mathrm{tp}}) = b$, and $h(z^S_{\mathrm{tp}}) = c$ for $\mathrm{tp} = \mathrm{tp}(a)$. Hence, claims (1)–(3) follow by Lemma 1. □

3.3 Abstraction Refinement

Individuals from an ABox $\mathcal{A}$ may correspond to several abstract individuals in the ABox abstraction $\mathcal{B}$: Each individual a corresponds to the abstract individual x_{tp} for $\mathrm{tp} = \mathrm{tp}(a)$. In addition, if $R(b, a) \in \mathcal{A}$ or $S(a, b) \in \mathcal{A}$ for some individual b, then a also corresponds to y^R_{tp} and z^S_{tp} respectively for $\mathrm{tp} = \mathrm{tp}(b)$. The additional individuals y^R_{tp} and z^S_{tp} were introduced intentionally to refine the initial abstraction when new assertions of abstract individuals are derived, which in turn, can be used to derive new assertions of individuals in $\mathcal{A}$. Specifically, if we add the new assertion according to cases (2) and (3) of Lemma 2, we may obtain different assertions for individuals that previously had the same types. Hence, adding the newly derived assertions using Lemma 2 may refine the types of the original individuals and, in turn, result in a new abstraction, for which new assertions can be derived once again.

The above suggests the following materialization procedure based on abstraction refinement. Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ we proceed as follows:

1. Build an initial abstraction $\mathcal{B}$ of $\mathcal{A}$ according to Definition 3.
2. Materialize $\mathcal{B} \cup \mathcal{T}$ using a reasoner.
3. Extend $\mathcal{A}$ with the newly derived assertions according to Lemma 2.
4. Update the types of the individuals in $\mathcal{A}$ and re-compute its abstraction $\mathcal{B}$.
5. Repeat from Step 2 until no new assertions can be added to $\mathcal{A}$.

Example 4 (Example 3 continued). Let $\mathcal{A}_{\text{I}}$ the ABox $\mathcal{A}$ from Example 3 and a TBox $\mathcal{T} = \{A \sqsubseteq \forall R.B,\ B \sqsubseteq \forall R^{-}.A\}$. Let $\mathcal{B}_{\text{I}}$ be the abstraction $\mathcal{B}$ of $\mathcal{A}_{\text{I}} = \mathcal{A}$ computed in Example 3 (see Figure 1). By materializing $\mathcal{B}_{\text{I}}$ w.r.t. $\mathcal{T}$ we get $B(y^R_{\text{tp}_2})$, from which we obtain $\mathcal{A}_{\text{II}} = \mathcal{A}_{\text{I}} \cup \{B(b), B(e), B(c)\}$ using Lemma 2. Recomputing the types of individuals in $\mathcal{A}_{\text{II}}$ yields tp(b) = tp(c) = tp(e) = tp$_3$= $(\{B\}, \{R\}, \{R\})$, while the types of a and d remain unchanged. The abstraction of $\mathcal{A}_{\text{II}}$ is thus $\mathcal{B}_{\text{II}} = \mathcal{B}_{\text{tp}_2} \cup \mathcal{B}_{\text{tp}_3}$, where $\mathcal{B}_{\text{tp}_3} = \{B(x_{\text{tp}_3}),\ R(x_{\text{tp}_3}, y^R_{\text{tp}_3}),\ R(z^R_{\text{tp}_3}, x_{\text{tp}_3})\}$. By materializing $\mathcal{B}_{\text{II}}$, we get $A(z^R_{\text{tp}_3})$, from which we obtain $\mathcal{A}_{\text{III}} = \mathcal{A}_{\text{II}} \cup \{A(b)\}$. We again recompute the types of individuals in $\mathcal{A}_{\text{III}}$, which gives tp(b) = tp$_4$ $= (\{A, B\}, \{R\}, \{R\})$, while the types of the other individuals do not change. The abstraction of $\mathcal{A}_{\text{III}}$ is thus $\mathcal{B}_{\text{III}} = \mathcal{B}_{\text{tp}_2} \cup \mathcal{B}_{\text{tp}_3} \cup \mathcal{B}_{\text{tp}_4}$, where $\mathcal{B}_{\text{tp}_4} = \{A(x_{\text{tp}_4}),\ B(x_{\text{tp}_4}),\ R(x_{\text{tp}_4}, y^R_{\text{tp}_4}),\ R(z^R_{\text{tp}_4}, x_{\text{tp}_4})\}$. Materializing $\mathcal{B}_{\text{III}}$ yields $B(y^R_{\text{tp}_4})$ and $A(z^R_{\text{tp}_4})$, which correspond to $B(c)$, $B(e)$, and $A(a)$. However, those assertions already exist in $\mathcal{A}_{\text{III}}$, so the procedure terminates.

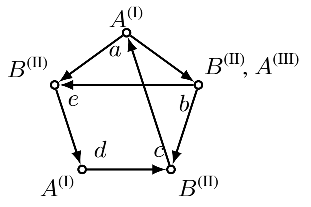
The abstraction refinement procedure terminates since after every iteration except the last one, new atomic assertions must be added to $\mathcal{A}$, and there is a bounded number of such assertions. Specifically, the number of iterations is at most $\|\mathsf{ind}(\mathcal{O})\| \times \|\mathsf{con}(\mathcal{O})\|$. The number of realized individual types in every ABox $\mathcal{A}$, and hence the size of every abstract ABox $\mathcal{B}$, is at most exponential in the number of different concepts and roles in $\mathcal{O}$. In practice, it is likely to be much smaller since not every possible type is realized. Note also that in practice, it is not necessary to add the newly derived assertions explicitly to the original ABox—one can recompute the new types using some simple operations on the sets of individuals (intersection and unions), and keep the derived assertions only once for every new equivalence class.

4 Completeness

Lemma 2 guarantees the soundness of our procedure. However, in general our procedure is not complete, as demonstrated by the following example.

Example 5. Consider the ABox $\mathcal{A} = \{A(a),\ R(a, b),\ B(b)\}$ and the TBox $\mathcal{T} = \{B \sqsubseteq C \sqcup D, \exists R.C \sqsubseteq C, A \sqcap C \sqsubseteq \forall R.D\}$. Note that $\mathcal{A} \cup \mathcal{T} \models D(b)$. We have tp$(a)$ =

ABox: $\mathcal{A} =$



TBox: $\mathcal{T} =$

$A \sqsubseteq \forall R.B$

$B \sqsubseteq \forall R^{-}.A$

Materialized abstract ABoxes	Abstractions I	II	III
$y^R_{\mathrm{tp}_1}$	$\{c, e, a, d\}$		
x_{tp_1}	$\{b, c, e\}$		
$z^R_{\mathrm{tp}_1}$	$\{a, b, d\}$		
$+B$ $y^R_{\mathrm{tp}_2}$	$\{b, e, c\}$	$\{b, e, c\}$	$\{b, e, c\}$
A x_{tp_2}	$\{a, d\}$	$\{a, d\}$	$\{a, d\}$
$z^R_{\mathrm{tp}_2}$	$\{c, e\}$	$\{c, e\}$	$\{c, e\}$
$y^R_{\mathrm{tp}_3}$		$\{c, e, a, d\}$	$\{a, d\}$
B x_{tp_3}		$\{b, e, c\}$	$\{c, e\}$
$+A$ $z^R_{\mathrm{tp}_3}$		$\{a, b, d\}$	$\{b, d, a\}$
$+B$ $y^R_{\mathrm{tp}_4}$			$\{c, e\}$
A, B x_{tp_4}			$\{b\}$
$+A$ $z^R_{\mathrm{tp}_4}$			$\{a\}$

Fig. 1. The abstractions I-III produced in Example 4. Each abstraction consists of the ABoxes corresponding to the four individual types. The inferred assertions are indicated with the "+" sign and are added to the corresponding original individuals shown in each column. The materialized assertions in the original ABox are labeled with the first iteration in which they appear.

$(\{A\}, \{R\}, \emptyset)$, $\mathrm{tp}(b) = (\{B\}, \emptyset, \{R\})$. Therefore the abstraction $\mathcal{B} = \mathcal{B}_{\mathrm{tp}(a)} \cup \mathcal{B}_{\mathrm{tp}(b)}$, where $\mathcal{B}_{\mathrm{tp}(a)} = \{A(x_{\mathrm{tp}(a)}), R(x_{\mathrm{tp}(a)}, y^R_{\mathrm{tp}(a)})\}$, $\mathcal{B}_{\mathrm{tp}(b)} = \{B(x_{\mathrm{tp}(b)}), R(z^R_{\mathrm{tp}(b)}, x_{\mathrm{tp}(b)})\}$. Since $\mathcal{B} \cup \mathcal{T}$ does not entail any new atomic concept assertions, our procedure terminates without producing the entailment $\mathcal{A} \cup \mathcal{T} \models D(b)$.

The primary reason for incompleteness in this example is that our abstraction breaks the ABox into disconnected parts, which cannot communicate the non-deterministic choices, e.g., for the disjunction $C \sqcup D$. The only communication between ABoxes happens through the entailment of new assertions. If the ontology language does not allow such non-deterministic constructors, it is possible to obtain a complete procedure.

4.1 Horn $\mathcal{ALCHOI}$

In this section, we restrict ontologies to a Horn fragment of $\mathcal{ALCHOI}$:

Definition 4 (Horn $\mathcal{ALCHOI}$). *An $\mathcal{ALCHOI}$ ontology $\mathcal{O}$ is* Horn *if, for every concept assertion $D(a)$ and every axiom $C \sqsubseteq D$, the concepts C and D satisfy, respectively, the following grammar definitions:*

$$C ::= \top \mid \bot \mid A \mid o \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \mid \exists R.C, \tag{1}$$

$$D ::= \top \mid \bot \mid A \mid o \mid D_1 \sqcap D_2 \mid \exists R.D \mid \forall R.D \mid \neg C. \tag{2}$$

Intuitively, negations and universal restrictions should not occur negatively, and disjunctions should not occur positively. We also allow axioms equivalent to Horn axioms.

It is well-known that every consistent Horn ontology has a so-called *canonical model* that entails exactly the consequences entailed by the ontology. For our purpose, we require a weaker version of this property.

Theorem 1 (Weak Canonical Model Property for Horn $\mathcal{ALCHOI}$). *Every consistent Horn $\mathcal{ALCHOI}$ ontology $\mathcal{O}$ has a model $\mathcal{I}$ such that $\mathcal{I} \models A(a)$ implies $\mathcal{O} \models A(a)$ for every atomic concept assertion $A(a)$ with $a \in \mathsf{ind}(\mathcal{O})$ and $A \in \mathsf{con}(\mathcal{O})$.*

Theorem 1 can be proved using the property that Horn $\mathcal{ALCHOI}$ models are closed under direct products. Then a canonical model is obtained from the direct product of models refuting (finitely many) atomic non-types.

Before formulating our completeness result, we need to solve one small technical problem illustrated in the following example.

Example 6. Consider $\mathcal{A} = \{A(a),\ B(b),\ R(a,b)\}$, $\mathcal{T} = \{A \sqcap \exists R.B \sqsubseteq C\}$. Clearly, $\mathcal{A} \cup \mathcal{T} \models C(a)$. We have $\mathsf{tp}(a) = \mathsf{tp}_1 = (\{A\}, \{R\}, \emptyset)$, $\mathsf{tp}(b) = \mathsf{tp}_2 = (\{B\}, \emptyset, \{R\})$, so our abstraction $\mathcal{B} = \mathcal{B}_{\mathsf{tp}_1} \cup \mathcal{B}_{\mathsf{tp}_2}$ with $\mathcal{B}_{\mathsf{tp}_1} = \{A(x_{\mathsf{tp}_1}),\ R(x_{\mathsf{tp}_1}, y^R_{\mathsf{tp}_1}\}$, and $\mathcal{B}_{\mathsf{tp}_2} = \{B(x_{\mathsf{tp}_2}),\ R(z^R_{\mathsf{tp}_2}, x_{\mathsf{tp}_2})\}$. Since no new atomic concept assertions are entailed by $\mathcal{B} \cup \mathcal{T}$, our procedure terminates without deriving $C(a)$.

Note that $\mathcal{B}_{\mathsf{tp}_2} \cup \mathcal{T} \models (\exists R.B)(z^R_{\mathsf{tp}_2})$, so there is an entailed assertion, just not an atomic one. To capture this inference, we introduce a new concept that "defines" $\exists R.B$. Specifically, let $\mathcal{T}' = \{\exists R.B \sqsubseteq X,\ A \sqcap X \sqsubseteq C\}$ where X is a fresh concept name. Clearly, $\mathcal{T}'$ is a conservative extension of $\mathcal{T}$, and from $\mathcal{T}'$ we can derive a new assertion $\mathcal{B}_{\mathsf{tp}_2} \cup \mathcal{T}' \models X(z^R_{\mathsf{tp}_2})$. If we now add the corresponding assertion $X(a)$ to $\mathcal{A}$ and recompute the abstraction for the updated type $\mathsf{tp}(a) = \mathsf{tp}_3 = (\{A, X\}, \{R\}, \emptyset)$ ($\mathsf{tp}(b)$ does not change), we have $\mathcal{B}_{\mathsf{tp}_3} = \{A(x_{\mathsf{tp}_3}),\ X(x_{\mathsf{tp}_3}),\ R(x_{\mathsf{tp}_3}, y^R_{\mathsf{tp}_3})\}$, and obtain $\mathcal{B}_{\mathsf{tp}_3} \cup \mathcal{T}' \models C(x_{\mathsf{tp}_3})$, which gives us the intended result.

Example 6 suggests that to achieve completeness, we need to represent existential restrictions on the left hand side of the axioms using new atomic concepts. Note that $\exists R.B \sqsubseteq X$ is equivalent to $B \sqsubseteq \forall R^-.X$. Thus we can just require that there are no existential restrictions on the left hand side of concept inclusions, and all universal restrictions on the right have only atomic concepts as fillers.

Definition 5 (Normal Form for Horn $\mathcal{ALCHOI}$). *Horn $\mathcal{ALCHOI}$ axioms $D(a)$ and $C \sqsubseteq D$ are in* normal form *if they satisfy the following grammar definitions:*

$$C ::= \top \mid \bot \mid A \mid o \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \tag{3}$$

$$D ::= \top \mid \bot \mid A \mid o \mid D_1 \sqcap D_2 \mid \exists R.D \mid \forall R.A \mid \neg C \tag{4}$$

Intuitively, in addition to the constraints for Horn $\mathcal{ALCHOI}$ ontologies in Definition 4, negative occurrence of existential restrictions are not allowed, and (positive) occurrences of universal restrictions can only have concept names as fillers. It is easy to convert axioms to a normal form using a well-known structural transformation.

4.2 Completeness Proof

We are now ready to show the following completeness result:

Theorem 2. *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be a normalized Horn $\mathcal{ALCHOI}$ ontology and $\mathcal{B}$ the abstraction of $\mathcal{A}$. $\mathcal{A}$ is concept materialized if, for every type $tp = (tp_{\downarrow}, tp_{\rightarrow}, tp_{\leftarrow})$, every individual $a \in \mathsf{ind}(\mathcal{A})$ with $tp(a) = tp$, and every atomic concept A, we have:*

(1) $\mathcal{B} \cup \mathcal{T} \models A(x_{tp})$ implies $A(a) \in \mathcal{A}$,
(2) $\mathcal{B} \cup \mathcal{T} \models A(y_{tp}^{R})$ and $R(a, b) \in \mathcal{A}$ implies $A(b) \in \mathcal{A}$, and
(3) $\mathcal{B} \cup \mathcal{T} \models A(z_{tp}^{S})$ and $S(c, a) \in \mathcal{A}$ implies $A(c) \in \mathcal{A}$.

Proof. To prove Theorem 2, we extend the abstraction $\mathcal{B}$ of $\mathcal{A}$ with new role assertions $R(x_{\mathrm{tp}(a)}, x_{\mathrm{tp}(b)})$ for every $R(a, b) \in \mathcal{A}$. Let us denote this extended abstract ABox by $\mathcal{B}'$. Since, for every $C(a) \in \mathcal{A}$, we also have $C \in \mathrm{tp}_{\downarrow}(a)$ and, thus, $C(x_{\mathrm{tp}(a)}) \in \mathcal{B} \subseteq \mathcal{B}'$, the mapping $h : \mathsf{ind}(\mathcal{A}) \to \mathsf{ind}(\mathcal{B}')$ defined by $h(a) = x_{\mathrm{tp}(a)}$ is a homomorphism from $\mathcal{A}$ to $\mathcal{B}'$. Therefore, by Lemma 1, if $\mathcal{A} \cup \mathcal{T} \models A(a)$, then $\mathcal{B}' \cup \mathcal{T} \models A(x_{\mathrm{tp}(a)})$. The key part of the proof is to demonstrate that in this case we also have $\mathcal{B} \cup \mathcal{T} \models A(x_{\mathrm{tp}(a)})$. That is, the extended abstract ABox $\mathcal{B}'$ does not entail new atomic concept assertions compared to $\mathcal{B}$. It follows then that $A(a) \in \mathcal{A}$ by condition (1) of the theorem. This implies that $\mathcal{A}$ is concept materialized.

To prove that $\mathcal{B}'$ entails the same atomic concept assertions as $\mathcal{B}$, we use the remaining conditions (2) and (3) of Theorem 2 and the canonical model property formulated in Theorem 1. Note that since new role assertions of the form $R(x_{\mathrm{tp}(a)}, x_{\mathrm{tp}(b)})$ are added to $\mathcal{B}'$ only if $R(a, b) \in \mathcal{A}$, we have $R \in \mathrm{tp}_{\rightarrow}(a)$ and $R \in \mathrm{tp}_{\leftarrow}(b)$ by Definition 2. Therefore, we already had role assertions $R(x_{\mathrm{tp}(a)}, y_{\mathrm{tp}(a)}^{R}) \in \mathcal{B}$ and likewise $R(z_{\mathrm{tp}(b)}^{R}, x_{\mathrm{tp}(b)}) \in \mathcal{B}$ for the same role R. Furthermore, by condition (2) of Theorem 2, if $\mathcal{B} \cup \mathcal{T} \models A(y_{\mathrm{tp}(a)}^{R})$, then since $R(a, b) \in \mathcal{A}$, we also have $A(b) \in \mathcal{A}$, and thus $A(x_{\mathrm{tp}(b)}) \in \mathcal{B}$. Likewise, by condition (3), if $\mathcal{B} \cup \mathcal{T} \models A(z_{\mathrm{tp}(b)}^{R})$, then $A(x_{\mathrm{tp}(a)}) \in \mathcal{B}$. The following lemma shows that with these properties for $\mathcal{B}$, after adding the new role assertion $R(x_{\mathrm{tp}(a)}, x_{\mathrm{tp}(b)})$ to $\mathcal{B}$, no new atomic concept assertions can be entailed.

Lemma 3 (Four-Individual Lemma). *Let $\mathcal{O}$ be a normalized Horn $\mathcal{ALCHOI}$ ontology such that $\{R(x_1, y_1), R(z_2, x_2)\} \subseteq \mathcal{O}$ for some x_1, y_1, z_2, x_2, and R. Further, assume that for every concept name A we have:*

(1) $\mathcal{O} \models A(y_1)$ implies $\mathcal{O} \models A(x_2)$, and
(2) $\mathcal{O} \models A(z_2)$ implies $\mathcal{O} \models A(x_1)$.

Finally, let $\mathcal{O}' = \mathcal{O} \cup \{R(x_1, x_2)\}$. Then for every concept name A and every individual a, we have $\mathcal{O}' \models A(a)$ implies $\mathcal{O} \models A(a)$.

Proof (Sketch). Suppose that $\mathcal{O}' \models A(a)$, we will show $\mathcal{O} \models A(a)$. If $\mathcal{O}$ is inconsistent then this holds trivially. Otherwise, there exists a canonical model $\mathcal{I}$ of $\mathcal{O}$ satisfying Theorem 1. From $\mathcal{I}$ we construct an interpretation $\mathcal{I}'$ that coincides with $\mathcal{I}$ apart from the interpretation of roles S such that: $S^{\mathcal{I}'} = S^{\mathcal{I}} \cup \{(x_1^{\mathcal{I}}, x_2^{\mathcal{I}})\}$ if $\mathcal{O} \models R \sqsubseteq S$; and $S^{\mathcal{I}'} = S^{\mathcal{I}} \cup \{(x_2^{\mathcal{I}}, x_1^{\mathcal{I}})\}$ if $\mathcal{O} \models R \sqsubseteq S^{-}$. We prove that $\mathcal{I}' \models \mathcal{O}'$, which implies $\mathcal{I}' \models A(a)$ since $\mathcal{O}' \models A(a)$, and thus $\mathcal{I} \models A(a)$ by definition of $\mathcal{I}'$, from which $\mathcal{O} \models A(a)$ follows since $\mathcal{I}$ satisfies Theorem 1. $\square$

By repeatedly applying Lemma 3 for each $x_1 = x_{\mathrm{tp}(a)}$, $y_1 = y_{\mathrm{tp}(a)}^{R}$, $x_2 = x_{\mathrm{tp}(b)}$, $z_2 = z_{\mathrm{tp}(b)}^{R}$ and R such that $R(a, b) \in \mathcal{A}$, we obtain that $\mathcal{B}'$ entails only those atomic assertions that are entailed by $\mathcal{B}$, which completes the proof of Theorem 2. $\square$

Table 2. Test suite statistics with the number of atomic concepts in the ontology ($\#A$ and $\#A_N$ for the normalized ontology), roles ($\#R$), individuals ($\#I$), role ($\#R(a,b)$) and concept ($\#A(a)$) assertions, and the number of entailed concept assertions

Ontology	$\#A$	$\#A_N$	$\#R$	$\#I$	$\#R(a,b)$	$\#A(a)$	$\#A(a)$ entailed
Gazetteer	710	711	15	516 150	604 164	11 112	538 799
Robert	61	135	80	405	1 089	1	4 255
Coburn	719	1 161	109	123 515	237 620	297 002	535 124
UOBM 1	69	90	35	24 925	153 571	44 680	142 747
UOBM 10	69	90	35	242 549	1 500 628	434 115	1 381 722
UOBM 50	69	90	35	1 227 445	7 594 996	2 197 035	6 991 583
UOBM 100	69	90	35	2 462 012	15 241 909	4 409 891	14 027 772

5 Implementation and Evaluation

To evaluate the feasibility of our approach, we implemented the procedure sketched in Section 3.3 prototypically in Java. The system relies on a database for storing the ABox and use external reasoners for materializing the abstractions. We focus on features, which indicate a potential gain of our approach: the size of abstract ABoxes and the number of refinements steps. In addition, to see how the abstraction changes, we present statistics for the first and last abstractions, as well as statistics of individual types.

We selected non-trivial ontologies, for which we present some metrics in Table 2.[3] Gazetteer and Corburn are bio ontologies from the NCBO BioPortal and the Phenoscape project, respectively. Robert is sophisticated and deliberately serves as a worst case example for our approach. UOBM ontologies are generated from the University Ontology Benchmark,[4] where UOBM n denotes a data set for n universities. Note that the increase of normalized concepts (A_N) in comparison to the original concepts (A) in Table 2 is a rough indicator of TBox complexity, which adds extra workload to reasoners.

Table 3 shows the results of computing and iteratively refining the abstract ABoxes until the fixpoint is reached. In general, there are only few refinement steps, and the abstraction reduces the size of the ABox significantly. This can be observed in particular for the UOBM with growing ABox sizes. Note that for ontologies such as UOBM 1 or Roberts family ontology the abstraction can be larger than the original ABox. The latter ontology also requires more refinement steps. This is a result of the heterogeneity of the ABox (concept and role assertions) in combination with the complexity of the TBox.

Our qualitative performance evaluations reveals that the compression degree of an ABox has a correspondence to the gain in time for materializing the abstract ABoxes. We compared the respective reasoning times of the original ABox with the sum of reasoning times for all abstractions using the OWL DL reasoning systems HermiT and Konclude.[5] For ontologies with high compression rate such as Gazetteer and Coburn the sum of the reasoning times for all abstractions is less than a tenth of the reasoning

[3] Download and references at `http://www.derivo.de/dl14-ontologies/`
[4] `http://www.cs.ox.ac.uk/isg/tools/UOBMGenerator/`
[5] See `http://www.hermit-reasoner.com` and `http://www.konclude.com`

Table 3. Number of refinement steps, relative sizes of the abstractions, and statistics about individual types (averaged over all types) for the first and the last abstraction of ontologies

Ontology	# of	Abstract ABox size (%)						Individual types statistics					
	steps	% indiv.		$\%A(\cdot)$		$\%R(\cdot,\cdot)$		# indiv. / type		$\#A_N$ / type		$\#R$ / type	
Gazetteer	2	1	1	12	1	< 1	1	279.76	279.76	0.73	2.56	2.05	2.05
Robert	7	89	142	2 000	68	26	41	5.00	3.12	0.24	22.16	3.50	3.44
Coburn	3	3	3	< 1	1	1	1	116.96	115.22	2.49	5.27	2.79	2.79
UOBM 1	3	102	158	21	47	15	23	8.03	5.25	3.02	14.18	7.18	7.35
UOBM 10	3	41	70	9	21	6	10	21.18	13.98	3.34	15.48	7.58	7.86
UOBM 50	3	18	32	4	10	3	5	47.88	28.36	3.52	16.29	7.82	8.14
UOBM 100	3	13	22	3	7	2	3	70.36	41.60	3.59	16.62	7.91	8.24

time for the original ABox and requires at most a quarter of RAM. Roberts, as expected from the abstraction statistics, shows a degradation in runtime (4 to 8 times) and RAM usage (up to 2 times) over the abstractions. The sequence of growing UOBM ABoxes ($1, 10, 50, 100$ universities) effectively shows the benefits of our abstraction procedure with large ABoxes. Whereas the runtimes for the abstractions of UOBM 1 are still 2 to 4 times that of the original ABox, the runtimes for UOBM 50 are already down by 50% after abstraction. The original UOBM 100 ontology could neither be processed within eight hours by HermiT nor by Konclude with a 32 GB RAM limit, but its abstraction can easily be materialized, e.g., within 84 seconds and 8 GB RAM by Konclude.

The purpose of the presented evaluation was to estimate the potential gains of using abstraction for ontology materialization, and for this reason we did not focus yet on efficient computation of abstractions and of their refinements. As of now we just recompute the abstraction from scratch after each refinement step. This takes, e.g., about 200 seconds for each abstraction of UOBM 100, but there is certainly plenty of room for optimizations.

6 Related Work

Many approaches have been proposed to handle large ABoxes, e.g., by rule-based materialization in a (relational) database [14,8]. Since we focus on the abstraction of the ABox, we discuss work that is closely related to our aims.

The SHER approach [3] merges similar individuals to obtain a compressed, so-called *summary* ABox, which is then used for (refutation-based) consistency checking. The technique (as well as ours) is based on the observation that individuals with the same asserted types are likely to have the same entailed types. Since merging in SHER is only based on asserted concepts, the resulting summary ABox might be inconsistent even if the original ABox is consistent w.r.t. the TBox. To remedy this, justifications [7] are used to decide which merges caused the inconsistency and to refine the summary accordingly. Justification-based refinements are also needed for query answering since SHER does reasoning at query time. We avoid justification computation by partitioning individuals of the same type into equivalence classes. Such partitioning guarantees the soundness of derived atomic concept assertions. We also have to perform refinement steps, but the refinement is to incrementally derive more consequences. What is computed before remains sound.

Wandelt and Möller propose a technique for instance retrieval over $\mathcal{SHI}$ ontologies based on modularization [16]. With modularization they refer to the process of partitioning the ABox such that an OWL reasoner can perform (refutation-based) instance checking over the smaller partitions. To achieve such partitions, the authors allow for breaking up role assertions when it can be guaranteed (syntactically) that no concepts are propagated over them. While this does not reduce the overall size of the data, they also propose an optimization that uses a notion similar to our types, but extended to also consider the concept assertions for successors and predecessors. As in our case, consequences derived from the ABoxes for these extended types are sound. Completeness is only guaranteed if the ABox for the extended type completely covers the partition for the individual. If that is not the case, the whole partition for the individual has to be processed by the reasoner. While this optimization is similar to our initial abstraction, Wandelt and Möller resolve to reasoning over the (possibly large) ABox partition for an individual if necessary for completeness, whereas we propose to refine the abstraction instead. The former approach syntactically identifies individual partitions, while our approach semantically, on-demand propagates concept assertions via refinements.

7 Conclusions and Future Work

We have presented an approach for ontology materialization based on abstraction refinement. The main idea is to partition ABox individuals into equivalent classes such that information derived for their abstract representatives can be used to refine the abstraction. Our experiment demonstrates that the approach particularly pays off for ontologies with large ABoxes and simple TBoxes.

Currently, our approach is complete for Horn $\mathcal{ALCHOI}$ due to the property that only (deterministically) derived assertions are used for abstraction refinement. We could potentially extend our approach to non-Horn ontology languages by exploiting the additional information about non-deterministic "possible" instances such as those provided by the HermiT reasoner. Directly supporting other Horn features, e.g. cardinality and role chains, is not possible. However, some could be supported without a considerable modification of the procedure. For example, it is easy to support transitive roles and role chains by using the well-known encoding of these axioms via concept inclusions [12].

In this paper we mainly focus on concept materialization since role materialization for $\mathcal{ALCHOI}$ can be essentially computed by expanding role hierarchies (special care needs to be taken about nominals though). When ontologies contain role chains and functional roles, however, materialization of role assertions becomes less trivial (e.g., the encoding of role chains is not enough). It is left for future work to investigate how these features can be fully supported.

The abstract ABoxes could serve not only as a generic interface for communication with the reasoner, but also as a compact representation of the materialization. This can be useful for other reasoning tasks. Specifically, when answering instance and conjunctive queries over the materialized ABoxes, the abstraction can be used to prune the search space.

References

1. Baader, F., Calvanese, D., McGuinness, D., Nardi, D., Patel-Schneider, P. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge University Press, second edn. (2007)
2. 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)
3. Dolby, J., Fokoue, A., Kalyanpur, A., Schonberg, E., Srinivas, K.: Scalable highly expressive reasoner (SHER). J. of Web Semantics 7(4), 357–361 (2009)
4. 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). AAAI Press (2012)
5. Glimm, B., Kazakov, Y., Liebig, T., Tran, T.K., Vialard, V.: Abstraction refinement for ontology materialization. Tech. rep., University of Ulm and derivo GmbH (2014), `https://www.uni-ulm.de/fileadmin/website_uni_ulm/iui.inst.090/Publikationen/2014/abstractionrefinementTR.pdf`
6. Glimm, B., Krötzsch, M.: SPARQL beyond subgraph matching. In: Proc. of the 9th Int. Semantic Web Conf. (ISWC 2010). LNCS, vol. 6496, pp. 241–256. Springer (2010)
7. 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). LNCS, vol. 4825, pp. 267–280. Springer (2007)
8. Kolovski, V., Wu, Z., Eadon, G.: Optimizing enterprise-scale OWL 2 RL reasoning in a relational database system. In: Proc. of the Int. Semantic Web Conf. (ISWC 2010). LNCS, vol. 6496, pp. 436–452. Springer (2010)
9. 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)
10. Lutz, C., Toman, D., Wolter, F.: Conjunctive query answering in the description logic $\mathcal{EL}$ using a relational database system. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009). pp. 2070–2075 (2009)
11. Rodriguez-Muro, M., Calvanese, D.: High performance query answering over DL-Lite ontologies. In: Proc. of the 13th Int. Conf. on the Principles of Knowledge Representation and Reasoning (KR 2012) (2012)
12. Simancik, F.: Elimination of complex RIAs without automata. In: Proc. of the 2012 Int. Workshop on Description Logics (DL 2012). CEUR Workshop Proceedings, vol. 846. CEUR-WS.org (2012)
13. Stefanoni, G., Motik, B., Horrocks, I.: Introducing nominals to the combined query answering approaches for $\mathcal{EL}$. In: Proc. of the 27th National Conf. on Artificial Intelligence (AAAI 2013). AAAI Press (2013)
14. 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)
15. Volz, R., Staab, S., Motik, B.: Incrementally maintaining materializations of ontologies stored in logic databases. J. Data Semantics 2, 1–34 (2005)
16. Wandelt, S., Möller, R.: Towards ABox modularization of semi-expressive description logics. J. of Applied Ontology 7(2), 133–167 (2012)