Ontology Materialization by Abstraction Refinement in Horn $\mathcal{SHOIF}$

Birte Glimm, Yevgeny Kazakov, and Trung-Kien Tran

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

Abstract. Abstraction refinement is a recently introduced technique which allows for reducing materialization of an ontology with a large ABox to materialization of a smaller (compressed) 'abstraction' of this ontology. Although this approach is sound for the very expressive language $\mathcal{SROIQ}$, so far it was proved to be complete only for Horn $\mathcal{ALCHOI}$ ontologies. In this paper we propose an extension of this method that is now complete for Horn $\mathcal{SHOIF}$ ontologies. Supporting functional roles, in particular, is nontrivial due to the sophisticated interaction with nominals and inverse roles. Additionally, we show how to compute not only the concept-materialization but also the role- and equality-materialization. To show completeness we use a tailored set of materialization rules that loosely decouple the ABox from the TBox, which is intrinsically interesting as reasoning with nominals in general requires both ABox and TBox.

1 Introduction

Description Logics (DLs) are popular languages for knowledge representation and reasoning. They are the underlying formalism for the standardized Web Ontology Language (OWL) and its profiles, which are widely used in many application areas. Recent years have also seen an increasing interest in ontology-based data access (OBDA), where a TBox with background knowledge, often expressed in a DL language, is used to enrich datasets (ABoxes), which are then accessible via queries. Ontology materialization is a reasoning task that computes logical consequences of the dataset w.r.t. the TBox and it is the most important task in some languages, e.g., OWL 2 RL [10]. In other languages, e.g., those that allow existential quantification, materialization is a stepping stone for query answering via rewriting [8].

To make ontology materialization useful in practice, especially for large datasets, scalable materialization is of great importance. Several approaches have been proposed to achieve this goal. The RDFox [11] and WebPIE [13] systems operate on the entire dataset and utilize parallel computing to perform a rule-based materialization for OWL 2 RL. Other approaches try to reduce the size of the dataset. Modules or so-called 'individual islands' [14] are used for reducing the set of ABox assertions to those that are sufficient for computing the entailed assertions for a given individual. The SHER system [4, 2] improves consistency checking of a large ABox by computing a so-called 'summary ABox' in which several original individuals are merged into one. If the resulting ABox is found consistent, then so is the original ABox. If not, then explanations [7] are used to pinpoint the contradictory axioms or relax the summary to avoid inconsistency. Similar to SHER, in the abstraction refinement method [5] several

individuals of the original ABox can be represented by one individual in a corresponding 'abstract ABox'. In contrast to the summary ABox, the abstract ABox provides an under-approximation rather than the over-approximation of entailments (e.g., if the abstraction is inconsistent then so is the original ABox but not necessarily vice versa). To ensure completeness of the method, the so-called refinement step is used that re-computes the abstraction based on new (sound) entailments obtained from a previous abstraction. This has the added benefit that not only consistency but also the full materialization of the ABox can be computed at once without (rather expensive) explanation computations or repeated consistency checks. This paper builds upon the abstraction refinement method and presents the following improvements:

1. We extend the approach to guarantee completeness in the presence of transitive and functional roles, thus fully supporting Horn $\mathcal{SHOIF}$ ontologies. Reasoning with nominals, inverse roles, and functionality is known to be difficult due to the loss of the tree-model property.
2. We materialize not only concept assertions, but also role and equality assertions. In $\mathcal{ALCHOI}$, role and equality assertions can be computed simply by expanding the role hierarchy and analyzing assertions of nominals. In $\mathcal{SHOIF}$, this is no longer the case. Apart from transitivity, role assertions could result from the consequence that some roles are functional; and interactions of nominals, inverse roles, and functionality could lead to a new kind of nominals. In Section 4, we discuss this issue in more detail and provide a solution in Section 5.
3. We present a new set of materialization rules (Section 6), which loosely decouple the ABox from the TBox. Although we use them only for proving completeness of the method (Section 7), these rules can be of interest on its own, e.g., as a basis of an efficient implementation for ABox reasoning. This provides a fresh view of the approach as the completeness proofs are principally different from the previous proofs [5], and the method can potentially be extended to other languages having similar rules.

2 Preliminaries

The syntax and semantics of the Description Logic (DL) $\mathcal{SHOIF}$ is briefly presented in Table 1. A $\mathcal{SHOIF}$ ontology $\mathcal{O}$ is *Horn* [9], if, for every $D(a) \in \mathcal{O}$ and $C \sqsubseteq D \in \mathcal{O}$, the concepts C and D satisfy the following grammar definitions:

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

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

Given an ontology $\mathcal{O}$, a role R is *functional* (in $\mathcal{O}$) if $\mathsf{func}(R) \in \mathcal{O}$ and *transitive* (in $\mathcal{O}$) if $\mathsf{tran}(R) \in \mathcal{O}$. (Horn) $\mathcal{ALCHOI}$ is the fragment of (Horn) $\mathcal{SHOIF}$ in which functionality and transitivity are disallowed. For an ontology $\mathcal{O}$, let $\sqsubseteq_{\mathcal{H}}^*$ be the reflexive transitive closure of the *role hierarchy* $\mathcal{H} = \{R \sqsubseteq S \in \mathcal{O}\}$. If $R \sqsubseteq_{\mathcal{H}}^* S$, then we say that R is a *sub-role* of S and S is a *super-role* of R; A role R is *simple* (in $\mathcal{O}$) if it has no transitive sub-roles; if $\mathsf{func}(R) \in \mathcal{O}$ then R must be simple.

Table 1. The syntax and semantics of the DL $\mathcal{SHOIF}$

	Syntax	Semantics
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	$\exists R.C$	$\{d \mid \exists e \in C^{\mathcal{I}} : \langle d, e \rangle \in R^{\mathcal{I}}\}$
universal restriction	$\forall R.C$	$\{d \mid \langle d, e \rangle \in R^{\mathcal{I}} \to e \in C^{\mathcal{I}}\}$
Axioms:		
concept inclusion	$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$
role inclusion	$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$
role transitivity	$\mathsf{tran}(R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
role functionality	$\mathsf{func}(R)$	$\langle d, e \rangle, \langle d, e' \rangle \in R^{\mathcal{I}} \to e = e'$
concept assertion	$C(a)$	$a^{\mathcal{I}} \in C^{\mathcal{I}}$
role assertion	$R(a, b)$	$\langle a^{\mathcal{I}}, b^{\mathcal{I}} \rangle \in R^{\mathcal{I}}$
equality assertion	$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$

To simplify the presentation, we do not distinguish between axioms $R(a, b)$ and $R^-(b, a)$, $a \approx b$ and $b \approx a$, $R \sqsubseteq S$ and $R^- \sqsubseteq S^-$, and $\mathsf{tran}(R)$ and $\mathsf{tran}(R^-)$. For example, if $R(a, b) \in \mathcal{A}$, we assume that $R^-(b, a) \in \mathcal{A}$. 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.

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 *equality-materialized* if $\mathcal{O} \models a \approx b$ implies $a \approx b \in \mathcal{O}$ for each $a, b \in \mathsf{ind}(\mathcal{O})$; $\mathcal{O}$ is (fully) *materialized* if it is concept-, role-, and equality-materialized. Given an ontology $\mathcal{O}$, the concept-, role-, equality-, and/or (full) materialization of $\mathcal{O}$ is the smallest super-set of $\mathcal{O}$ that is concept-, role-, equality-, and/or fully materialized respectively.

3 Computing Materialization by Abstraction

The main idea of the abstraction refinement method [5] is to materialize an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ with a potentially large ABox $\mathcal{A}$ by constructing a smaller ABox $\mathcal{B}$ such that the materialization of $\mathcal{B} \cup \mathcal{T}$ can be computed by a general-purpose reasoner, and transfer the new entailments back to $\mathcal{O}$. The ABox $\mathcal{B}$ is usually called the *abstraction* of the *original* ABox $\mathcal{A}$ (or just the *abstract ABox* if $\mathcal{A}$ is clear from the context), and the individuals in $\mathcal{B}$ are called *representatives* of the *original* individuals in $\mathcal{A}$. All results in this section apply to any DL with (classical) set-theoretic semantics, e.g., $\mathcal{SROIQ}$ [6].

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*

$h(C(a)) := C(h(a))$, $h(R(a,b)) := R(h(a), h(b))$, *and* $h(a \approx b) := h(a) \approx h(b)$. *We say an individual* $b \in \mathsf{ind}(\mathcal{B})$ *is a* representative *of an individual* $a \in \mathsf{ind}(\mathcal{A})$ *if there exists a homomorphism* $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ *such that* $h(b) = a$.

Example 1. Consider the ABoxes $\mathcal{A} = \{A(a),\ A(b)\}$ and $\mathcal{B} = \{A(u)\}$. Then the individual u of $\mathcal{B}$ is a representative for both individuals a and b since the mappings $h_1 = \{u \mapsto a\}$ and $h_2 = \{u \mapsto b\}$ are homomorphisms from $\mathcal{B}$ to $\mathcal{A}$.

The following property of homomorphisms allows transferring entailments from the abstraction to the original ABox.

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

Corollary 1. *If an individual* $u \in \mathsf{ind}(\mathcal{B})$ *is a representative for an individual* $a \in \mathsf{ind}(\mathcal{A})$, *then, for every TBox* $\mathcal{T}$ *and concept* C, *if* $\mathcal{B} \cup \mathcal{T} \models C(u)$, *then* $\mathcal{A} \cup \mathcal{T} \models C(a)$.

According to Corollary 1, in Example 1 one can *transfer* any newly entailed concept assertion for u to the corresponding assertions for a and b. In fact, in this particular case, all entailed concept assertions for $\mathcal{A}$ can be computed this way because there is also a homomorphism from $\mathcal{A}$ to $\mathcal{B}$.

Example 2 (Example 1 continued). Consider the homomorphism $h : \mathsf{ind}(\mathcal{A}) \to \mathsf{ind}(\mathcal{B})$ defined by $h = \{a \mapsto u,\ b \mapsto u\}$. Then by Lemma 1, for every TBox $\mathcal{T}$ and concept C if $\mathcal{A} \cup \mathcal{T} \models C(a)$ or $\mathcal{A} \cup \mathcal{T} \models C(b)$ then $\mathcal{B} \cup \mathcal{T} \models C(u)$.

In practice, computing a sufficiently small abstraction $\mathcal{B}$ of $\mathcal{A}$ such that there are homomorphisms in both directions is rarely possible, so the set of concept assertions transferred using Corollary 1 is usually incomplete. To ensure completeness, one can employ further *refinement* steps that recompute the abstraction based on the new information derived. This method was shown to be complete for concept materialization of Horn $\mathcal{ALCHOI}$ ontologies [5]. The aim of this paper is to extend this approach to Horn $\mathcal{SHOIF}$. Before we go into further details explaining our extension, we first describe challenges that the new functionality and transitivity axioms pose for ontology materialization.

4   Full Materialization for Horn $\mathcal{SHOIF}$

It is easy to show using model-theoretic arguments that an $\mathcal{ALCHOI}$ ontology $\mathcal{O}$ without equality assertions entails an equality between individuals $a \approx b$ iff either $a = b$ or both a and b are instances of some nominal concept o occurring in $\mathcal{O}$. To compute such entailed equality assertions, it is sufficient to compute instances of nominals, which can be accomplished by introducing an axiom $o \sqsubseteq A_o$ with a fresh concept A_o for each nominal o and computing instances of A_o by concept materialization. If $\mathcal{O}$ contains equality assertions, one needs to additionally perform the transitive symmetric closure of the resulting equality assertions. For role materialization, similarly, one can show that if $\mathcal{O} \models R(a,b)$ then either (i) there exists an assertion $R'(a',b') \in \mathcal{O}$ such that $\mathcal{O} \models a \approx a'$, $\mathcal{O} \models b \approx b'$, and $R' \sqsubseteq_{\mathcal{H}}^{*} R$, or (ii) a is an instance of $\exists R.o$ and b is an

instance of o for some nominal o, or (iii) a is an instance of o and b is an instance of $\exists R^-.o$ for some nominal o. All these conditions can again be checked by introducing fresh concepts and computing the concept materialization.

That is, (full) materialization of Horn $\mathcal{ALCHOI}$ ontologies can be reduced to concept materialization by simple syntactic transformations. The following examples illustrate that for Horn $\mathcal{SHOIF}$ ontologies such reductions do not work.

Example 3. Consider the ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ with $\mathcal{A} = \{A(a), A(b)\}$ and $\mathcal{T} = \{A \sqsubseteq \exists F^-.o, \mathsf{func}(F)\}$. Then $\mathcal{O} \models a \approx b$ but neither a nor b are instances of the nominal o.

As Example 3 illustrates, equality testing in (Horn) $\mathcal{SHOIF}$ becomes less trivial; the main reason is that using a combination of functional roles, inverse roles, and nominals one can express *entailed nominal concepts* such as A in Example 3, which can be interpreted by at most one element.

In the following example, we demonstrate how functional roles can also result in some non-trivial entailments of role assertions, even if no equality or nominals are used.

Example 4. Consider the ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ with $\mathcal{A} = \{A(a), R(a,b)\}$ and $\mathcal{T} = \{A \sqsubseteq \exists S.\top, R \sqsubseteq F, S \sqsubseteq F, \mathsf{func}(F)\}$. Then $\mathcal{O} \models S(a,b)$ but $\mathcal{O} \not\models R \sqsubseteq S$.

As can be seen from Examples 3 and 4, the computation of equality- and role-materialization becomes a non-trivial problem for Horn $\mathcal{SHOIF}$ ontologies. Fortunately, using the following corollary of Lemma 1, one can extend the main idea behind concept materialization described in the previous section to equality- and role-materialization.

Corollary 2. *Let $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ be a homomorphism between the ABoxes $\mathcal{B}$ and $\mathcal{A}$, $u, v \in \mathsf{ind}(\mathcal{B})$, $a = h(u), b = h(v) \in \mathsf{ind}(\mathcal{A})$, and $\mathcal{T}$ a $\mathcal{SROIQ}$ TBox. Then $\mathcal{B} \cup \mathcal{T} \models u \approx v$ implies $\mathcal{A} \cup \mathcal{T} \models a \approx b$, and $\mathcal{B} \cup \mathcal{T} \models R(u,v)$ implies $\mathcal{A} \cup \mathcal{T} \models R(a,b)$ for every role R.*

Unfortunately, the abstract ABoxes that are sufficient to guarantee completeness of concept-materialization are not sufficient to guarantee completeness of equality- and role-materialization as demonstrated in the following example.

Example 5. Consider the ABox $\mathcal{A}$ and its abstraction $\mathcal{B}$ in Example 1. As stated in Example 2, for any TBox $\mathcal{T}$ all entailed concept assertions of $\mathcal{T} \cup \mathcal{A}$ can be obtained using Corollary 1 for the abstraction $\mathcal{B}$. However, the abstraction $\mathcal{B}$ may be insufficient for computing all entailed role or equality assertions using Corollary 2. Indeed, consider $\mathcal{T} = \{A \sqsubseteq o\}$. Then $\mathcal{A} \cup \mathcal{T} \models a \approx b$, but, clearly, there is no homomorphism $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ such that $h(u) = a$ and $h(u) = b$ required to derive this assertion using Corollary 2. Similarly, it is possible that $\mathcal{A} \cup \mathcal{T} \models R(a,b)$ for some role R, but we are not able to derive this assertion using Corollary 2, for example, for $\mathcal{T} = \{A \sqsubseteq \exists T.o, A \sqsubseteq \exists T^-.o, \mathsf{tran}(T), T \sqsubseteq R\}$.

5 Abstraction Refinement for Horn $\mathcal{SHOIF}$

The general algorithm for ontology reasoning using the abstraction refinement method can be summarized as follows:

1. Build a *suitable* abstraction of the original ontology;
2. Compute the entailments from the abstraction using a reasoner and *transfer* them to the original ontology using homomorphisms (Lemma 1);
3. *Compute the deductive closure* of the original ontology using some (light-weight) *rules*;
4. Repeat from Step 1 until no new entailments can be added to the original ontology.

The efficiency and theoretical properties of this method depend on the choices of how the abstraction is computed in Step 1, which entailments are transferred in Step 2, and which rules are used to compute the deductive closure in Step 3. In the following we detail these choices for Horn $\mathcal{SHOIF}$.

To compute the abstraction of the original ABox (Step 1), we define *types* of individuals based on their assertions.

Definition 2. *Let $\mathcal{A}$ be an ABox and a an individual. The* concept type *of a is a set of concepts $\tau_C(a) = \{C \mid C(a) \in \mathcal{A}\}$. The* role type *of a is a set of roles $\tau_R(a) = \{R \mid \exists b : R(a, b) \in \mathcal{A}\}$. The* (combined) type *of a is a pair $\tau(a) = \langle \tau_C(a), \tau_R(a) \rangle$ where $\tau_C(a)$ is the concept type of a and $\tau_R(a)$ is the role type of a.*

Note that we assume w.l.o.g. that, for every individual a, $\top(a)$ is included in the ontology and, therefore, $\top$ occurs in every (concept) type. To simplify the presentation, however, we usually omit $\top$ in the remainder.

Example 6. Let $\mathcal{A} = \{A(a), A(b), R(a, b)\}$. Then $\tau_C(a) = \tau_C(b) = \tau_1 = \{A\}$, $\tau_R(a) = \{R\}$, $\tau_R(b) = \{R^-\}$, $\tau(a) = \tau_2 = \langle \{A\}, \{R\} \rangle$, and $\tau(b) = \tau_3 = \langle \{A\}, \{R^-\} \rangle$.

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

Definition 3. *The* abstraction *of an ABox $\mathcal{A}$ is an ABox $\mathcal{B} = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} (\mathcal{B}_{\tau_C(a)} \cup \mathcal{B}_{\tau(a)})$, where:*

- *for each concept type τ_C, $\mathcal{B}_{\tau_C} = \{C(u_{\tau_C}) \mid C \in \tau_C\}$,*
- *for each combined type $\tau = \langle \tau_C, \tau_R \rangle$, $\mathcal{B}_\tau = \{C(v_\tau) \mid C \in \tau_C\} \cup \{R(v_\tau, w_\tau^R) \mid R \in \tau_R\}$,*

where u_{τ_C}, v_τ, and w_τ^R are distinguished abstract individuals *for each concept type τ_C and each combined type τ.*

Example 7. The abstraction for $\mathcal{A}$ in Example 6 is the ABox $\mathcal{B} = \mathcal{B}_{\tau_C(a)} \cup \mathcal{B}_{\tau_C(b)} \cup \mathcal{B}_{\tau(a)} \cup \mathcal{B}_{\tau(b)}$, where $\mathcal{B}_{\tau_C(a)} = \mathcal{B}_{\tau_C(b)} = \mathcal{B}_{\tau_1} = \{A(u_{\tau_1})\}$, $\mathcal{B}_{\tau(a)} = \mathcal{B}_{\tau_2} = \{A(v_{\tau_2}), R(v_{\tau_2}, w_{\tau_2}^R)\}$, $\mathcal{B}_{\tau(b)} = \mathcal{B}_{\tau_3} = \{A(v_{\tau_3}), R^-(v_{\tau_3}, w_{\tau_3}^{R^-})\}$.

Intuitively, the abstraction is a disjoint union of ABoxes simulating concept and combined types. Note that each mapping $h : \mathsf{ind}(\mathcal{B}) \to \mathsf{ind}(\mathcal{A})$ such that:

$$h(u_{\tau_C}) \in \{a \in \mathsf{ind}(\mathcal{A}) \mid \tau_C(a) = \tau_C\}, \tag{3}$$

$$h(v_\tau) \in \{a \in \mathsf{ind}(\mathcal{A}) \mid \tau(a) = \tau\}, \tag{4}$$

$$h(w_\tau^R) \in \{b \in \mathsf{ind}(\mathcal{A}) \mid R(h(v_\tau), b) \in \mathcal{A}\}, \tag{5}$$

is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$. This allows us to transfer entailments from the abstraction back to the original ABox using Corollaries 1 and 2. Note that each original individual a in $\mathcal{A}$ has at least two representatives in $\mathcal{B}$: $u_{\tau_C(a)}$ that has exactly the same concept assertions as a and $v_{\tau(a)}$ which additionally has assertions with the same roles. Two representatives are necessary to solve the problem mentioned in Example 5 for transferring role and equality assertions.

Example 8. Consider the ABox $\mathcal{A} = \{A(a), A(b)\}$ and TBox $\mathcal{T} = \{A \sqsubseteq o\}$ mentioned in Example 5. We have $\tau_C(a) = \tau_C(b) = \tau_1 = \{A\}$, and $\tau(a) = \tau(b) = \tau_2 = \langle\{A\}, \emptyset\rangle$. The abstraction of $\mathcal{A}$ is defined as $\mathcal{B} = \mathcal{B}_{\tau_1} \cup \mathcal{B}_{\tau_2}$ with $\mathcal{B}_{\tau_1} = \{A(u_{\tau_1})\}$, $\mathcal{B}_{\tau_2} = \{A(v_{\tau_2})\}$. Since $\mathcal{B} \cup \mathcal{T} \models u_{\tau_1} \approx v_{\tau_2}$, and $h = \{u_{\tau_1} \mapsto a, v_{\tau_2} \mapsto b\}$ is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$, using Corollary 2 we obtain $\mathcal{A} \cup \mathcal{T} \models a \approx b$.

Next, we detail how entailed assertions are transferred from $\mathcal{B}$ to $\mathcal{A}$ in Step 2 of the algorithm. To achieve completeness it is not necessary to transfer all of them and we detail which assertions are to be transferred after the definition.

Definition 4. *Let $\mathcal{B}$ be the abstraction of an ABox $\mathcal{A}$ (according to Definition 3), and $\Delta\mathcal{B}$ a set of assertions. The* update *of $\mathcal{A}$ (using $\Delta\mathcal{B}$) is the smallest set of assertions $\Delta\mathcal{A}$ such that:*

$$
\begin{array}{rcl}
C(v_{\tau(a)}) \in \Delta\mathcal{B} & \Rightarrow & C(a) \in \Delta\mathcal{A}, \quad (6) \\
R(a,b) \in \mathcal{A},\, C(w_{\tau(a)}^R) \in \Delta\mathcal{B}, & \Rightarrow & C(b) \in \Delta\mathcal{A}, \quad (7) \\
R(a,b) \in \mathcal{A},\, S(v_{\tau(a)}, w_{\tau(a)}^R) \in \Delta\mathcal{B} & \Rightarrow & S(a,b) \in \Delta\mathcal{A}, \quad (8) \\
S(v_{\tau(a)}, v_{\tau(a)}) \in \Delta\mathcal{B} & \Rightarrow & S(a,a) \in \Delta\mathcal{A} \quad (9) \\
u_{\tau_C(a)} \approx v_{\tau(b)} \in \Delta\mathcal{B} & \Rightarrow & a \approx b \in \Delta\mathcal{A}, \quad (10) \\
R(u_{\tau_C(a)}, v_{\tau(b)}) \in \Delta\mathcal{B} & \Rightarrow & R(a,b) \in \Delta\mathcal{A}. \quad (11)
\end{array}
$$

Lemma 2. *Let $\mathcal{B}$ be the abstraction of $\mathcal{A}$, $\Delta\mathcal{A}$ an update for $\Delta\mathcal{B}$, and $\mathcal{T}$ a TBox. Then $\mathcal{B} \cup \mathcal{T} \models \Delta\mathcal{B}$ implies $\mathcal{A} \cup \mathcal{T} \models \Delta\mathcal{A}$.*

Proof. It is easy to see that for each $\alpha \in \Delta\mathcal{A}$ and $\beta \in \Delta\mathcal{B}$ in cases (6)–(11), there is a homomorphism h from $\mathcal{B}$ to $\mathcal{A}$ satisfying conditions (3)–(5) such that $h(\beta) = \alpha$. Hence, by Lemma 1, $\mathcal{B} \cup \mathcal{T} \models \beta$ implies $\mathcal{A} \cup \mathcal{T} \models h(\beta) = \alpha$. $\qquad\square$

Let $\mathcal{B}'$ be the materialization of $\mathcal{B} \cup \mathcal{T}$, then we transfer assertions from $\Delta\mathcal{B} = \mathcal{B}' \setminus \mathcal{B}$ according to Definition 4 in Step 2. Next, we compute the (deductive) closure of the ABox $\mathcal{A}$ under equality, transitivity, and functionality in Step 3.

Definition 5. *We say that an ABox $\mathcal{A}$ is* equality-closed *if:*

$$a \in \mathsf{ind}(\mathcal{A}) \text{ implies } a \approx a \in \mathcal{A} \tag{12}$$

$$\{a \approx b, b \approx c\} \subseteq \mathcal{A} \text{ implies } a \approx c \in \mathcal{A}, \tag{13}$$

$$\{a \approx b, A(a)\} \subseteq \mathcal{A} \text{ implies } A(b) \in \mathcal{A}, \tag{14}$$

$$\{a \approx b, R(a,c)\} \subseteq \mathcal{A} \text{ implies } R(b,c) \in \mathcal{A}, \tag{15}$$

$\mathcal{A}$ *is* closed under the axiom $\mathsf{tran}(T)$ *if:*

$$\{T(a,b), T(b,c)\} \subseteq \mathcal{A} \text{ implies } T(a,c) \in \mathcal{A}. \tag{16}$$

$\mathcal{A}$ *is* closed under the axiom $\mathsf{func}(F)$ *if:*

$$\{F(a,b), F(a,c)\} \subseteq \mathcal{A} \text{ implies } b \approx c \in \mathcal{A}. \tag{17}$$

The closure of $\mathcal{A}$ *(w.r.t. a TBox $\mathcal{T}$) under equality, transitivity, and/or functionality *is the smallest super-set of $\mathcal{A}$ that is closed under equality, for each $\mathsf{tran}(T) \in \mathcal{T}$ and/or each $\mathsf{func}(F) \in \mathcal{T}$, respectively.*

Computing the closure of an ABox under equality, functionality, and transitivity is a relatively lightweight operation that does not require using a reasoner. Note that all these assertions must be derived in order to compute the full materialization. In the previous method [5], the abstraction of an ABox $\mathcal{A}$ is defined as $\mathcal{B} = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathcal{B}_{\tau(a)}$, and there is no computation of the closure as in Step 3. One can easily check that the previous method is incomplete not only for role and equality materializations but also for concept materialization of Horn $\mathcal{SHOIF}$ ontologies, even with the computation of the closure in Step 3.

Since Steps 2 and 3 can only extend the original ABox with entailed atomic assertions, the procedure is *sound*, and since the number of such assertions is bounded, the procedure eventually terminates. We next show that the procedure is *complete* for Horn $\mathcal{SHOIF}$ ontologies, that is, the resulting ontology is (fully) materialized when the procedure terminates.

6   A Criterion for Ontology Materialization

To prove completeness of the abstraction refinement procedure in the case of Horn $\mathcal{SHOIF}$, we characterize when such ontologies are fully materialized by means of closure of the ABox assertions under certain rules. The rules are similar to the rules for reasoning in Horn $\mathcal{SHIQ}$ and Horn $\mathcal{SHOIQ}$ [12, 3] in the sense that they derive logical consequences of axioms. Since we are only interested in ABox consequences and not going to use these rules for computing the materialization (but merely for proving completeness of our abstraction refinement algorithm), however, we will not derive TBox axioms explicitly but use their entailments in side conditions of the rules.

Recall from the discussion after Example 3, that in $\mathcal{SHOIF}$ one can express some non-trivial *nominal concepts*. To be able to reason about such concepts, we need to extend the language with new axiom types.

Definition 6. *A* concept cardinality restriction *is an axiom of the form* $|C| \geq n$ *or* $|C| = n$ *with C a concept and* $n \in \mathbb{N}$. *An interpretation* $\mathcal{I}$ *satisfies* $|C| \geq n$ ($|C| = n$), *written* $\mathcal{I} \models |C| \geq n$ ($\mathcal{I} \models |C| = n$), *iff* $|C^{\mathcal{I}}| \geq n$ ($|C^{\mathcal{I}}| = n$).

We also use *role conjunctions* $R \sqcap S$, interpreted by $(R \sqcap S)^{\mathcal{I}} = R^{\mathcal{I}} \cap S^{\mathcal{I}}$. The new constructors and axioms are used only in the conditions of rules and not in the ontology.

In the following, we denote by N and M conjunctions of atomic concepts and/or nominals and by H conjunctions of roles (possibly with indexes and superscripts). If we write $C \in N$, $N \subseteq N'$, $N \subsetneq N'$, we treat N and N' as the corresponding sets of conjuncts, where $N = \top$ denotes the empty conjunction. Similarly for conjunctions of roles. For a role conjunction H we denote by H^- the conjunction of inverses of roles in H. We write $N(a) \in \mathcal{A}$ if $C(a) \in \mathcal{A}$ for every $C \in N$. Note that if $N(a) \in \mathcal{A}$ then $\mathcal{A} \models |N| \geq 1$.

Definition 7. *Let* $\mathcal{A}$ *be an ABox. Then* $\mathfrak{N}(\mathcal{A}) = \{|N| \geq 1 \mid N(a) \in \mathcal{A}\}$ *is the* set of cardinality axioms induced by $\mathcal{A}$.

$$\mathbf{R}_{\approx}^1 \frac{}{a \approx a} : a \in \mathsf{ind}(\mathcal{A}) \quad \mathbf{R}_{\approx}^2 \frac{a \approx b \quad b \approx c}{a \approx c} \quad \mathbf{R}_{\approx}^3 \frac{a \approx b \quad R(a,c)}{R(b,c)} \quad \mathbf{R}_{\approx}^4 \frac{a \approx b \quad A(a)}{A(b)}$$

$$\mathbf{R}_t^1 \frac{T(a,b) \quad T(b,c)}{T(a,c)} : \mathsf{tran}(T) \in \mathcal{T} \quad \mathbf{R}_t^2 \frac{N(a)}{T(a,a)} : \mathcal{T}' \models N \sqsubseteq \exists(T \sqcap T^-).\top, \mathsf{tran}(T) \in \mathcal{T}$$

$$\mathbf{R}_t^3 \frac{N(a) \quad M(b)}{T(a,b)} : \mathcal{T}' \models \{N \sqsubseteq \exists T.N', M \sqsubseteq \exists T^-.N', |N'| = 1\}, \mathsf{tran}(T) \in \mathcal{T}$$

$$\mathbf{R}_{||} \frac{N(a) \quad N(b)}{a \approx b} : \mathcal{T}' \models |N| = 1 \qquad \mathbf{R}_{\forall} \frac{N(a) \quad R(a,b)}{B(b)} : \mathcal{T}' \models N \sqsubseteq \forall R.B$$

$$\mathbf{R}_{\exists} \frac{N(a) \quad M(b)}{R(a,b)} : \mathcal{T}' \models \{N \sqsubseteq \exists R.M, |M| = 1\} \qquad \mathbf{R}_{\sqsubseteq}^1 \frac{R(a,b)}{S(a,b)} : R \sqsubseteq S \in \mathcal{T}$$

$$\mathbf{R}_{\leq}^1 \frac{F(a,b) \quad F(a,c)}{b \approx c} : \mathsf{func}(F) \in \mathcal{T} \qquad \mathbf{R}_{\sqsubseteq}^2 \frac{N(a)}{B(a)} : \mathcal{T}' \models N \sqsubseteq B$$

$$\mathbf{R}_{\leq}^2 \frac{M(a) \quad F(a,b)}{H(a,b) \quad N(b)} : \mathcal{T}' \models M \sqsubseteq \exists(F \sqcap H).N, \mathsf{func}(F) \in \mathcal{T}$$

Fig. 1. Materialization rules with $\mathcal{T}' = \mathcal{T} \cup \mathfrak{N}(\mathcal{A})$

The materialization rules are presented in Figure 1. These rules are complete for ontology materialization provided the input ontology is in normal form.

Definition 8. *A Horn* $\mathcal{SHOIF}$ *ontology* $\mathcal{O}$ *is in* normal form *if* (1) *for every* $D(a) \in \mathcal{O}$, *D is an atomic concept;* (2) *for every* $C \sqsubseteq D \in \mathcal{O}$, *the concepts C and D satisfy the following grammar:*

$$C_{(i)} ::= \top \mid \bot \mid A \mid o \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \tag{18}$$

$$D_{(i)} ::= \top \mid \bot \mid A \mid o \mid \exists R.A \mid \forall R.A \mid \neg C \tag{19}$$

and (3) *for every* $C \sqsubseteq \forall R.A \in \mathcal{O}$ *and every transitive sub-role* T *of* R, *there exists an atomic concept* B *that occurs only in* $\{C \sqsubseteq \forall T.B, B \sqsubseteq \forall T.B, B \sqsubseteq A\} \subseteq \mathcal{O}$ *and not in* C *or* A.

Theorem 1. *Let* $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ *be a normalized Horn* $\mathcal{SHOIF}$ *ontology and* $\mathcal{T}' = \mathcal{T} \cup \mathfrak{N}(\mathcal{A})$. *Then* $\mathcal{A}$ *is closed under the rules in Figure 1 w.r.t.* $\mathcal{T}$ *iff* $\mathcal{O}$ *is fully materialized.*

Proof (sketch). The *if* direction is trivial since all rules derive logical consequences of the axioms in premises and side conditions. For the *only if* direction, if $\mathcal{T}'$ is inconsistent then we can derive all assertions using rules $\mathbf{R}^2_{\sqsubseteq}$, $\mathbf{R}_{\exists}$, and $\mathbf{R}_{||}$. If $\mathsf{ind}(\mathcal{O}) = \emptyset$, then Theorem 1 trivially holds. We assume $\mathsf{ind}(\mathcal{O}) \neq \overline{\overline{\emptyset}}$ and $\mathcal{T}'$ is consistent. We then construct a model $\mathcal{J}$ of $\mathcal{O}$ such that $\mathcal{J} \models \alpha$ implies $\alpha \in \mathcal{A}$, for every atomic assertion α. Then, $\mathcal{O} \models \alpha$ implies $\mathcal{J} \models \alpha$, which implies $\alpha \in \mathcal{A}$, i.e., $\mathcal{O}$ is materialized. Such model $\mathcal{J}$ is obtained in two steps: 1) construct an interpretation $\mathcal{I}$ satisfying all but transitivity axioms in $\mathcal{O}$ such that $\mathcal{I} \models \alpha$ implies $\alpha \in \mathcal{A}$ for every atomic assertion α; 2) obtain $\mathcal{J}$ from $\mathcal{I}$ by extending the interpretation of non-simple roles to satisfy transitivity axioms.

Intuitively, $\mathcal{I}$ has a forest-like structure consisting of a graph part and tree parts. The graph part contains domain elements for individuals and concepts that have exactly one instance. The tree parts grow from the graph part to satisfy entailed existential axioms. To construct $\mathcal{I}$, we use a *chase-like* technique [1] in which new domain elements are introduced to satisfy axioms of the form $M \sqsubseteq \exists H.N$ entailed by $\mathcal{O}$ such that H and N are maximal (w.r.t M), i.e., there is no $M \sqsubseteq \exists H'.N'$ entailed by $\mathcal{O}$ and $H \cup N \subsetneq H' \cup N'$. This guarantees that new domain elements (and their respective assertions) also satisfy universal restrictions and role inclusions. Additionally, a new domain element is introduced only when an axiom cannot be satisfied by the existing domain elements. This ensures all the existential axioms are satisfied while the functionality axioms are not violated.

To obtain the model $\mathcal{J}$ from $\mathcal{I}$ that also satisfies the transitivity axioms in $\mathcal{O}$, we extend the interpretation of non-simple roles to include the transitive closure, i.e., for every transitive role T, (d_0, d_n) is added to $T^{\mathcal{J}}$ and to the interpretations of its super-roles if $(d_{i-1}, d_i) \in T^{\mathcal{I}}, 1 \leq i \leq n$. This extension makes $\mathcal{J}$ satisfy transitivity axioms and the rules $\mathbf{R}^1_t$–$\mathbf{R}^3_t$ ensure that $\mathcal{J} \models R(a, b)$ implies $R(a, b) \in \mathcal{A}$ for every role R and individuals a, b. Moreover, since only the interpretation of non-simple roles has been extended, it is also possible to show $\mathcal{J}$ entails the other axioms in $\mathcal{O}$, i.e. $\mathcal{J} \models \mathcal{O}$, and $\mathcal{J} \models \alpha$ implies $\alpha \in \mathcal{A}$ for every atomic assertion α, i.e., we obtain $\mathcal{J}$ as required. $\quad\square$

7 Completeness

Once the abstraction refinement procedure terminates, we claim that the input ontology is fully materialized by showing that it is closed under the rules in Figure 1.

Lemma 3. *Let* $\mathcal{A} \cup \mathcal{T}$ *be an ontology s.t.* $\mathcal{A}$ *is equality-, transitivity-, and functionality-closed,* $\mathcal{B}$ *the abstraction of* $\mathcal{A}$, $\mathcal{B}'$ *an ABox s.t.* $\mathcal{B} \subseteq \mathcal{B}'$, $\mathfrak{N}(\mathcal{B}') = \mathfrak{N}(\mathcal{A})$ *and* $\Delta\mathcal{A} \subseteq \mathcal{A}$ *with* $\Delta\mathcal{A}$ *the update of* $\mathcal{A}$ *using* $\mathcal{B}' \setminus \mathcal{B}$. *Then,* $\mathcal{B}'$ *is closed under the rules in Figure 1 w.r.t.* $\mathcal{T}$ *implies that* $\mathcal{A}$ *is also closed under the rules w.r.t.* $\mathcal{T}$.

Proof. Since $\mathfrak{N}(\mathcal{B}') = \mathfrak{N}(\mathcal{A})$, the side condition of each rule holds for $\mathcal{B}' \cup \mathcal{T}$ iff it holds for $\mathcal{A} \cup \mathcal{T}$. We examine each rule in Figure 1 and show that $\mathcal{A}$ is closed under that rule. Clearly, $\mathcal{A}$ is closed under $\mathbf{R}^1_\approx$–$\mathbf{R}^4_\approx$, $\mathbf{R}^1_\leq$, $\mathbf{R}^1_t$. For the other rules, the intuition is: if the premises of a rule $\mathbf{R}$ hold for some assertions γ in $\mathcal{A}$, then the premises of $\mathbf{R}$ also hold for the corresponding abstract assertions γ' in $\mathcal{B}$, and in $\mathcal{B}'$ consequently. Since $\mathcal{B}'$ is closed under $\mathbf{R}$, the conclusion κ' of γ' w.r.t. $\mathbf{R}$ is already in $\mathcal{B}'$. Then, the condition $\Delta\mathcal{A} \subseteq \mathcal{A}$ guarantees that the conclusion κ of γ w.r.t. $\mathbf{R}$ is already in $\mathcal{A}$, which implies that $\mathcal{A}$ is closed under $\mathbf{R}$. We present one interesting case: $\mathbf{R}^3_t$; the remaining cases are analogous. Let $\{N(a), M(b)\} \subseteq \mathcal{A}$, and the corresponding side condition holds. We have $\{N(v_{\tau(a)}), M(v_{\tau(b)})\} \subseteq \mathcal{B} \subseteq \mathcal{B}'$. Since $\mathcal{B}'$ is closed under $\mathbf{R}^3_t$, we have $T(v_{\tau(a)}, v_{\tau(b)}) \in \mathcal{B}'$. If $v_{\tau(a)} = v_{\tau(b)} = v_\tau$, i.e. a and b have the same type τ, we are not able to obtain $T(a, b)$ to add to $\mathcal{A}$. That explains why concept types are required. In this case, since $\{N(u_{\tau_C}), M(v_\tau)\} \subseteq \mathcal{B} \subseteq \mathcal{B}'$ and $\mathcal{B}'$ is closed under $\mathbf{R}^3_t$, we have $T(u_{\tau_C}, v_\tau) \in \mathcal{B}'$, and consequently $T(u_{\tau_C}, v_\tau) \in \mathcal{B}' \setminus \mathcal{B}$. By Definition 4, we have $T(a, b) \in \Delta\mathcal{A} \subseteq \mathcal{A}$. Therefore, $\mathcal{A}$ is closed under $\mathbf{R}^3_t$. $\square$

Using Lemma 3, we show that the procedure is complete.

Theorem 2. *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be the ontology obtained from the abstraction refinement procedure in Section 5. Then, $\mathcal{O}$ is fully materialized.*

Proof. Let $\mathcal{B}$ be the abstraction of $\mathcal{A}$, $\mathcal{B}' \cup \mathcal{T}$ the materialization of $\mathcal{B} \cup \mathcal{T}$, $\Delta\mathcal{B} = \mathcal{B}' \setminus \mathcal{B}$, and $\Delta\mathcal{A}$ the update of $\mathcal{A}$ using $\Delta\mathcal{B}$. For every $A(a) \in \mathcal{A}$, we have $A(v_{\tau(a)}) \in \mathcal{B} \subseteq \mathcal{B}'$, which implies $\mathfrak{N}(\mathcal{A}) \subseteq \mathfrak{N}(\mathcal{B}')(\star)$. There always exists a homomorphism h from $\mathcal{B}$ to $\mathcal{B}$ s.t. $h(u_{\tau_C(a)}) = v_{\tau(a)}$. Therefore, for every $A(u_{\tau_C(a)}) \in \mathcal{B}'$, we have $A(v_{\tau(a)}) \in \mathcal{B}'$. Since the procedure has terminated, i.e. $\Delta\mathcal{A} \subseteq \mathcal{A}$, for every $A(v_{\tau(a)}) \in \mathcal{B}'$ or $A(w^R_{\tau(a)}) \in \mathcal{B}'$ we have $A(a) \in \mathcal{A}$ for some $a \in \mathsf{ind}(\mathcal{A})$. Hence, we obtain $\mathfrak{N}(\mathcal{B}') \subseteq \mathfrak{N}(\mathcal{A})(\dagger)$. From $(\star)$ and $(\dagger)$ we have $\mathfrak{N}(\mathcal{A}) = \mathfrak{N}(\mathcal{B}')$, which allows us to apply Lemma 3.

By Theorem 1, $\mathcal{B}'$ is closed under the rules in Figure 1. Therefore, by Lemma 3, $\mathcal{A}$ is closed under those rules. Consequently, by Theorem 1, $\mathcal{O}$ is materialized. $\square$

8 Discussion and Future Work

Our next step is to evaluate the procedure in practice. We believe that it is particularly efficient for ontologies with large ABoxes. We have demonstrated in our previous work [5] that for such ontologies, the abstractions are often significantly smaller than the original ones. Given the (intentionally) close similarity with the previous procedure, it is reasonable to assume that the proposed extension would yield similar reductions. We also plan to extend the approach to allow more constructors in the Horn fragments. A possibility to handle qualified number restrictions is to extend the combined type of individuals by considering also concept assertions of their predecessors/successors. Materialization of concept assertions in the presence of role chains could be done using well-known encoding techniques but computing role assertions, especially for non-simple roles, might be more involved. Handing non-Horn ontologies requires major changes in the current approach as the abstraction can communicate only deterministic entailments; this will be the subject of future work.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley (1995)
2. Dolby, J., Fokoue, A., Kalyanpur, A., Schonberg, E., Srinivas, K.: Scalable highly expressive reasoner (SHER). J. of Web Semantics 7(4), 357–361 (2009)
3. 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)
4. Fokoue, A., Kershenbaum, A., Ma, L., Schonberg, E., Srinivas, K.: The summary ABox: Cutting ontologies down to size. In: Proc. of the 5th Int. Semantic Web Conference (ISWC 2006). LNCS, vol. 4273, pp. 343–356. Springer (2006)
5. Glimm, B., Kazakov, Y., Liebig, T., Tran, T., Vialard, V.: Abstraction refinement for ontology materialization. In: Proc. of the 13th Int. Semantic Web Conference (ISWC 2014). pp. 180–195. Springer (2014)
6. Horrocks, I., Kutz, O., Sattler, U.: The even more irresistible SROIQ. In: Proc. of 10th International Conference on Principles of Knowledge Representation and Reasoning (KR 2006). pp. 57–67 (2006)
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. Kontchakov, R., Lutz, C., Toman, D., Wolter, F., Zakharyaschev, M.: The combined approach to ontology-based data access. In: IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011. pp. 2656–2661 (2011)
9. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexities of horn description logics. ACM Trans. Comput. Log. 14(1), 2 (2013)
10. Motik, B., Cuenca Grau, B., Horrocks, I., Wu, Z., Fokoue, A., Lutz, C. (eds.): OWL 2 Web Ontology Language: Profiles (Second Edition). W3C Recommendation (2012), available at http://www.w3.org/TR/owl2-profiles/
11. 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 Twenty-Eighth AAAI Conference on Artificial Intelligence. pp. 129–137 (2014)
12. Ortiz, M., Rudolph, S., Simkus, M.: Worst-case optimal reasoning for the horn-dl fragments of OWL 1 and 2. In: Principles of Knowledge Representation and Reasoning: Proceedings of the Twelfth International Conference, KR 2010 (2010)
13. 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)
14. Wandelt, S., Möller, R.: Towards ABox modularization of semi-expressive description logics. J. of Applied Ontology 7(2), 133–167 (2012)