

Ontology Materialization by Abstraction Refinement in Horn *SHOIF*

Birte Glimm and Yevgeny Kazakov and Trung-Kien Tran

Institute of Artificial Intelligence, University of Ulm, Germany

<first name>.<last name>@uni-ulm.de

Abstract

Abstraction refinement is a recently introduced technique using which reasoning over large ABoxes is reduced to reasoning over small ‘abstract’ ABoxes. Although the approach is sound for any classical Description Logic such as *SRQIQ*, it is complete only for Horn *ALCHOI*. In this paper, we propose an extension of this method that is now complete for Horn *SHOIF* and also handles role- and equality-materialization. To show completeness, we use a tailored set of materialization rules that loosely decouple the ABox from the TBox. An empirical evaluation demonstrates that, despite the new features, the abstractions are still significantly smaller than the original ontologies and the materialization can be computed efficiently.

Introduction

Description Logics (DLs) are popular languages for knowledge representation and reasoning. They are the underlying formalism for the standardized Web Ontology Language OWL, which is widely used in many application areas. Recent years have also seen an increasing interest in ontology-based data access, 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. In other languages, e.g., those that allow existential quantification, materialization is a stepping stone for query answering via rewriting (Kontchakov et al. 2011).

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 R_{DF}ox (Motik et al. 2014) and WebPIE (Urbani et al. 2012) 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’ (Wandelt and Möller 2012) 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 (Dolby et al. 2009) improves consistency

checking and query answering for 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 (Kalyanpur et al. 2007) are used to pinpoint the contradictory axioms or relax the summary to avoid inconsistency. Similar to SHER, the abstraction refinement method for Horn *ALCHOI* (Glimm et al. 2014) represents several individuals of the original ABox by one individual in a corresponding ‘abstract ABox’. In contrast to the summary ABox, the abstract ABox, however, provides an under-approximation rather than an over-approximation of entailments. That is, whereas the summary ABox entails at least assertions entailed in the original ABox (when individuals are replaced with their representatives), the abstract ABox can entail at most such assertions. 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 without (rather expensive) explanation computations or repeated consistency checks. This paper significantly advances the abstraction refinement method in several directions:

1. We extend the approach to guarantee completeness in the presence of transitive and functional roles, thus fully supporting Horn *SHOIF* ontologies. Reasoning with nominals, inverse roles, and functionality is known to be challenging due to the loss of the tree-model property and the existence of implicitly cardinality constrained concepts (implicit nominals).
2. We materialize not only concept assertions, but also role and equality assertions. In *ALCHOI*, role and equality assertions can be computed by expanding the role hierarchy and analyzing assertions of nominals. In *SHOIF* special techniques are needed to properly handle functionality and the consequences of implicit nominals.
3. We present a new set of materialization rules, which loosely decouple the ABox from the TBox. Although we use them only for proving completeness of the method, 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

	Syntax	Semantics
<i>Concepts:</i>		
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	\perp	\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	$\forall R.C$	$\{d \mid \langle d, e \rangle \in R^{\mathcal{I}} \rightarrow e \in C^{\mathcal{I}}\}$
<i>Axioms:</i>		
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	$\text{tran}(R)$	$R^{\mathcal{I}} \circ R^{\mathcal{I}} \subseteq R^{\mathcal{I}}$
role functionality	$\text{func}(R)$	$\langle d, e \rangle, \langle d, e' \rangle \in R^{\mathcal{I}} \rightarrow e = e'$
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	$a \approx b$	$a^{\mathcal{I}} = b^{\mathcal{I}}$

Table 1: The syntax and semantics of the DL *SHOIF*

proofs are principally different from the proofs by Glimm et al. (2014) and the method can potentially be extended to other languages having similar rules.

4. We evaluate the approach on several real life and benchmark ontologies. The abstractions are often significantly smaller than the original ontologies (by orders of magnitude) and the materialization can be computed efficiently.

We refer readers to a technical report (Glimm, Kazakov, and Tran 2017) for full proofs of our results and further details.

Preliminaries

The syntax of *SHOIF* 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 an atomic role or an *inverse role* r^- with $r \in N_R$. We define $R^- := r^-$ if $R = r$ and $R^- := r$ if $R = r^-$. Complex *concepts* and *axioms* are defined in Table 1. An *ontology* \mathcal{O} is a finite set of axioms, written as $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, where \mathcal{A} is an *ABox* consisting of the concept, role, and equality assertions in \mathcal{O} and \mathcal{T} a *TBox* consisting of the concept and role inclusion, transitivity, and functionality axioms in \mathcal{O} . To simplify presentation, we do not distinguish between axioms $R(a, b)$, $a \approx b$, $R \sqsubseteq S$, $\text{tran}(R)$ and, respectively, $R^-(b, a)$, $b \approx a$, $R^- \sqsubseteq S^-$, $\text{tran}(R^-)$. We use $\text{con}(\mathcal{O})$, $\text{rol}(\mathcal{O})$, and $\text{ind}(\mathcal{O})$ for the sets of atomic concepts, atomic roles, and individuals occurring in \mathcal{O} , respectively.

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 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 by $(r^-)^{\mathcal{I}} = \{\langle e, d \rangle \mid \langle d, e \rangle \in r^{\mathcal{I}}\}$ and to complex concepts as shown in Table 1. \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 . A role R is *functional* (in \mathcal{O}) if $\text{func}(R) \in \mathcal{O}$ and *transitive* (in \mathcal{O}) if $\text{tran}(R) \in \mathcal{O}$. 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 $\text{func}(R) \in \mathcal{O}$, then R must be simple.

A *SHOIF* ontology \mathcal{O} is *Horn* (Krötzsch, Rudolph, and Hitzler 2013) and in *normalized* form if (1) for every $C(a) \in \mathcal{O}$, C is an atomic concept; (2) for every $C \sqsubseteq D \in \mathcal{O}$, the concepts C and D satisfy the following grammar:

$$C_{(i)} ::= \top \mid \perp \mid A \mid o \mid C_1 \sqcap C_2 \mid C_1 \sqcup C_2 \quad (1)$$

$$D_{(i)} ::= \top \mid \perp \mid A \mid o \mid \exists R.A \mid \forall R.A \mid \neg C \quad (2)$$

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 . W.l.o.g., we assume that every ontology is normalized by applying a structural transformation and a technique for eliminating transitivity axioms; see e.g. (Kazakov 2009).

(Horn) *ALCHOI* is the fragment of (Horn) *SHOIF* in which functionality and transitivity are disallowed.

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 $A \in \text{con}(\mathcal{O})$ and $a \in \text{ind}(\mathcal{O})$; \mathcal{O} is *role-materialized* if $\mathcal{O} \models r(a, b)$ implies $r(a, b) \in \mathcal{O}$ for each $r \in \text{rol}(\mathcal{O})$ and $a, b \in \text{ind}(\mathcal{O})$; \mathcal{O} is *equality-materialized* if $\mathcal{O} \models a \approx b$ implies $a \approx b \in \mathcal{O}$ for each $a, b \in \text{ind}(\mathcal{O})$; \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. Note that the full materialization of \mathcal{O} is always finite since the sets $\text{con}(\mathcal{O})$, $\text{rol}(\mathcal{O})$ and $\text{ind}(\mathcal{O})$ are finite.

Computing Materialization by Abstraction

The main idea of the abstraction refinement method is to materialize an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ with a large ABox \mathcal{A} by constructing a smaller ABox \mathcal{B} such that the materialization of \mathcal{O} is obtained from the materialization of $\mathcal{B} \cup \mathcal{T}$ by transferring entailments to \mathcal{O} in a certain way. The ABox \mathcal{B} is usually called the *abstraction* of the *original* ABox \mathcal{A} (or just the *abstract ABox*), 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., *SRQIQ* (Horrocks, Kutz, and Sattler 2006).

Definition 1. Let \mathcal{A} and \mathcal{B} be ABoxes. A mapping $h : \text{ind}(\mathcal{B}) \rightarrow \text{ind}(\mathcal{A})$ 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 $b \in \text{ind}(\mathcal{B})$ is a *representative* of an individual $a \in \text{ind}(\mathcal{A})$ if there exists a homomorphism $h : \text{ind}(\mathcal{B}) \rightarrow \text{ind}(\mathcal{A})$ 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 .

The following property of homomorphisms allows transferring entailments from abstractions to original ABoxes.

Lemma 1. *Let $h : \text{ind}(\mathcal{B}) \rightarrow \text{ind}(\mathcal{A})$ 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 2. *If an individual $u \in \text{ind}(\mathcal{B})$ is a representative for an individual $a \in \text{ind}(\mathcal{A})$, 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 2, in Example 1 one can transfer any 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 $h : \text{ind}(\mathcal{A}) \rightarrow \text{ind}(\mathcal{B})$ 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 2 is usually incomplete (e.g., homomorphisms from \mathcal{B} to \mathcal{A} guarantee only soundness). 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 *ALCHOI* ontologies (Glimm et al. 2014). The aim of this paper is to extend this approach to Horn *SHOIF*. Before we go into further details of our extension, we first describe challenges that the new functionality and transitivity axioms pose for ontology materialization.

Full Materialization for Horn *SHOIF*

It is easy to show using model-theoretic arguments that an *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 . 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 $R'(a', b') \in \mathcal{O}$ 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 be checked by introducing fresh concepts and computing the concept-materialization.

That is, (full) materialization of Horn *ALCHOI* ontologies can be reduced to concept-materialization by syntactic transformations. The following examples illustrate that for Horn *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 $\mathcal{T} = \{A \sqsubseteq \exists F^-.o, \text{func}(F)\}$. 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) *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 $\mathcal{T} = \{A \sqsubseteq \exists S.\top, R \sqsubseteq F, S \sqsubseteq F, \text{func}(F)\}$. 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 *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 3. *Let $h : \text{ind}(\mathcal{B}) \rightarrow \text{ind}(\mathcal{A})$ be a homomorphism between the ABoxes \mathcal{B} and \mathcal{A} , $u, v \in \text{ind}(\mathcal{B})$, $a = h(u)$, $b = h(v) \in \text{ind}(\mathcal{A})$, and \mathcal{T} a *SRQIQ* 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} from 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 2 for the abstraction \mathcal{B} . However, the abstraction \mathcal{B} may be insufficient for computing all entailed role or equality assertions using Corollary 3. Indeed, consider $\mathcal{T} = \{A \sqsubseteq o\}$. Then $\mathcal{A} \cup \mathcal{T} \models a \approx b$, but, clearly, there is no homomorphism $h : \text{ind}(\mathcal{B}) \rightarrow \text{ind}(\mathcal{A})$ such that $h(u) = a$ and $h(u) = b$ required to derive this assertion using Corollary 3. 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 3, for example, for $\mathcal{T} = \{A \sqsubseteq \exists T.o, A \sqsubseteq \exists T^-.o, \text{tran}(T), T \sqsubseteq R\}$.

Abstraction Refinement for Horn *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.

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 the 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 .

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 choosing one representative for each type with the respective assertions.

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

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

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

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

is a homomorphism from \mathcal{B} to \mathcal{A} . This allows us to transfer entailments back to the original ABox using Corollaries 2 and 3. Note that each original individual a in \mathcal{A} has at least two representatives in \mathcal{B} : $u_{\tau_C(a)}$, which has exactly the same concept assertions as a , and $v_{\tau(a)}$, which additionally has assertions with the same roles. The use of two representatives distinguishes the abstractions from the previously introduced ones (Glimm et al. 2014) and solves the problem with role and equality assertions in Example 5.

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 3 we obtain $\mathcal{A} \cup \mathcal{T} \models a \approx b$.

Next, we detail which 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.

Definition 4. Let \mathcal{B} be the abstraction of an ABox \mathcal{A} (by 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:

$$C(v_{\tau(a)}) \in \Delta\mathcal{B} \Rightarrow C(a) \in \Delta\mathcal{A}, \quad (6)$$

$$C(w_{\tau(a)}^R) \in \Delta\mathcal{B}, R(a, b) \in \mathcal{A} \Rightarrow C(b) \in \Delta\mathcal{A}, \quad (7)$$

$$S(v_{\tau(a)}, w_{\tau(a)}^R) \in \Delta\mathcal{B}, R(a, b) \in \mathcal{A} \Rightarrow S(a, b) \in \Delta\mathcal{A}, \quad (8)$$

$$S(v_{\tau(a)}, v_{\tau(a)}) \in \Delta\mathcal{B} \Rightarrow S(a, a) \in \Delta\mathcal{A} \quad (9)$$

$$u_{\tau_C(a)} \approx v_{\tau(b)} \in \Delta\mathcal{B} \Rightarrow a \approx b \in \Delta\mathcal{A}, \quad (10)$$

$$R(u_{\tau_C(a)}, v_{\tau(b)}) \in \Delta\mathcal{B} \Rightarrow R(a, b) \in \Delta\mathcal{A}. \quad (11)$$

The following lemma can be established using homomorphisms from \mathcal{B} to \mathcal{A} satisfying conditions (3)–(5).

Lemma 4. 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 $\mathcal{A} \cup \mathcal{T} \models \Delta\mathcal{A}$.

After transferring the entailed assertions according to Definition 4, in Step 3 we compute the closure of the ABox \mathcal{A} under equality, transitivity, and functionality.

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

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

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

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

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

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

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

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

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

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 $\text{tran}(T) \in \mathcal{T}$ and/or each $\text{func}(F) \in \mathcal{T}$, respectively.

Computing the closure of an ABox under equality, functionality, and transitivity is a relatively lightweight operation that does not require using a DL reasoner. Note that all these assertions must be derived in order to compute the full materialization. The previous method (Glimm et al. 2014) does not involve the computation of the closure as in Step 3. One can easily check that this and the lack of the additional individuals for the concept types results in incompleteness even for concept-materialization of Horn *SHOIQ* ontologies.

Soundness of the algorithm is a direct consequence of Lemma 4 and soundness of the additional rules (12)–(17), which hold for any DL with (classical) set-theoretic semantics, e.g., *SRQIQ*. Steps 2 and 3 can only extend the original ABox with entailed atomic assertions. Since the number of such assertions is bounded by the size of the materialization, the procedure eventually terminates and the number of repeat loops is polynomial in the size of the ontology. We next show that the procedure is *complete*.

$\mathbf{R}_{\approx}^1 \frac{}{a \approx a} : a \in \text{ind}(\mathcal{A})$	$\mathbf{R}_{\approx}^2 \frac{a \approx b \quad b \approx c}{a \approx c}$	$\mathbf{R}_{\approx}^3 \frac{a \approx b \quad R(a, c)}{R(b, c)}$	$\mathbf{R}_{\approx}^4 \frac{a \approx b \quad A(a)}{A(b)}$
$\mathbf{R}_t^1 \frac{T(a, b) \quad T(b, c)}{T(a, c)} : \text{tran}(T) \in \mathcal{T}$	$\mathbf{R}_t^2 \frac{N(a) \quad M(b)}{T(a, b)} : \mathcal{T}' \models \{N \sqsubseteq \exists T.N', M \sqsubseteq \exists T^-.N', N' = 1\}, \text{tran}(T) \in \mathcal{T}$		
$\mathbf{R}_t^3 \frac{N(a)}{T(a, a)} : \mathcal{T}' \models N \sqsubseteq \exists (T \sqcap T^-). \top, \text{tran}(T) \in \mathcal{T}$	$\mathbf{R}_{\exists} \frac{N(a) \quad M(b)}{R(a, b)} : \mathcal{T}' \models \{N \sqsubseteq \exists R.M, M = 1\}$		
$\mathbf{R}_{\leq}^1 \frac{F(a, b) \quad F(a, c)}{b \approx c} : \text{func}(F) \in \mathcal{T}$	$\mathbf{R}_{\parallel} \frac{N(a) \quad N(b)}{a \approx b} : \mathcal{T}' \models N = 1$	$\mathbf{R}_{\forall} \frac{N(a) \quad R(a, b)}{B(b)} : \mathcal{T}' \models N \sqsubseteq \forall R.B$	
$\mathbf{R}_{\leq}^2 \frac{M(a) \quad F(a, b)}{S(a, b) \quad A(b)} : \mathcal{T}' \models M \sqsubseteq \exists (F \sqcap S).A, \text{func}(F) \in \mathcal{T}$	$\mathbf{R}_{\sqsubseteq}^1 \frac{R(a, b)}{S(a, b)} : R \sqsubseteq S \in \mathcal{T}$	$\mathbf{R}_{\sqsubseteq}^2 \frac{N(a)}{B(a)} : \mathcal{T}' \models N \sqsubseteq B$	

Table 2: Materialization rules for Horn *SHOIF* ontologies

A Criterion for Ontology Materialization

To prove completeness of the abstraction refinement procedure in the case of Horn *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 *SHOIQ* (Ortiz, Rudolph, and Simkus 2010) 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 the 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 *SHOIF* one can express some non-trivial *nominal concepts*. We extend the language with such 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 or nominals. If we write $C \in N$, we treat N as the corresponding set of conjuncts, where $N = \top$ denotes the empty conjunction. We write $N(a) \in \mathcal{A}$ if $C(a) \in \mathcal{A}$ for every $C \in N$.

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}* .

The materialization rules for $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ are presented in Table 2 with $\mathcal{T}' = \mathcal{T} \cup \mathfrak{N}(\mathcal{A})$. These rules are complete for ontology materialization.

Theorem 5. Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be a normalized Horn *SHOIF* ontology. Then \mathcal{A} is closed under the rules in Table 2 w.r.t. \mathcal{T} iff \mathcal{O} is fully materialized.

Proof sketch. The non-trivial case is the *only if* direction when $\text{ind}(\mathcal{O}) \neq \emptyset$ and \mathcal{T}' is consistent. We show that there exists 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. To construct \mathcal{J} , we construct an interpretation \mathcal{I} , which satisfies all but transitivity axioms, using a *chase-like* technique (Abiteboul, Hull, and Vianu 1995). Intuitively, \mathcal{I} has a forest-like structure with a graph part containing elements for individuals and concepts that have exactly one instance. The tree parts grow from the graph part to satisfy entailed existential axioms. \mathcal{J} is then obtained from \mathcal{I} by extending the interpretation of non-simple roles to satisfy transitivity axioms. \square

Completeness

Once the abstraction refinement procedure terminates, we claim that the ontology is fully materialized by showing that it is closed under the rules in Table 2.

Lemma 6. Let $\mathcal{A} \cup \mathcal{T}$ be an ontology such that \mathcal{A} is equality-, transitivity-, and functionality-closed, \mathcal{B} the abstraction of \mathcal{A} , \mathcal{B}' an ABox such that $\mathcal{B} \subseteq \mathcal{B}'$, $\mathfrak{N}(\mathcal{B}') = \mathfrak{N}(\mathcal{A})$ 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 Table 2 w.r.t. \mathcal{T} implies that \mathcal{A} is also closed under the rules w.r.t. \mathcal{T} .

Proof sketch. 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 $\mathcal{A} \cup \mathcal{T}$. Clearly, \mathcal{A} is closed under \mathbf{R}_{\approx}^1 – \mathbf{R}_{\approx}^4 , \mathbf{R}_{\leq}^1 , \mathbf{R}_t^1 . For the other rules, the intuition is that 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} . \square

Using Lemma 6, we show that the procedure is complete.

ontology	TBox	ABox	mater.	1st abstraction		2nd abstraction		loading and reasoning times in seconds						
	# ax.	# assert	# assert	# types	# assert	# types	# assert	O _{load}	K _{reas} ^{abst}	O _{reas}	K _{load}	K _{reas} ^{CLI}	P _{load}	P _{reas}
NPD	773	911 K	1.7 M	1 K	21 K	—	—	12	4	6	9	5	21	49
DBP ⁺	1 762	25.2 M	28.5 M	188 K	3.2 M	187 K	3.9 M	470	297	412	240	145	101	167
IMDb ⁺	135	27.7 M	83.8 M	366	2 K	—	—	171	1	173	280	172	36	209
L10	96	850 K	1.3 M	29	227	29	321	6	2	7	10	5	3	8
L50	96	4.4 M	6.9 M	27	217	27	309	33	2	27	55	31	20	48
L100	96	8.9 M	13.9 M	27	217	27	309	68	2	61	102	67	43	97
L500	96	44.6 M	69.3 M	27	217	27	309	433	2	304	592	351	221	621
U10	157	1.9 M	3.9 M	4 K	91 K	9 K	369 K	35	35	54	23	25	7	25
U50	157	9.7 M	19.8 M	8 K	173 K	19 K	781 K	129	69	155	120	144	43	138
U100	157	19.5 M	39.8 M	10 K	217 K	24 K	1 M	199	95	267	254	315	89	298
U500	157	98.3 M	200.3 M	15 K	336 K	42 K	1.8 M	1 152	143	1 213	1 440	2107	509	1 591

Table 3: Test ontologies, sizes of abstractions, and reasoning times with and without abstraction (K=thousands, M=millions)

Theorem 7. *The ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ obtained from the abstraction refinement procedure 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} . For every $A(a) \in \mathcal{A}$, we have $A(v_{\tau(a)}) \in \mathcal{B} \subseteq \mathcal{B}'$, which implies $\mathfrak{N}(\mathcal{A}) \subseteq \mathfrak{N}(\mathcal{B}')$. There always exists a homomorphism h from \mathcal{B} to \mathcal{B}' such that $h(u_{\tau_C(a)}) = v_{\tau(a)}$. Therefore, for every $A(u_{\tau_C(a)}) \in \mathcal{B}'$, we have $A(v_{\tau(a)}) \in \mathcal{B}'$. Since the procedure terminated, i.e. $\Delta\mathcal{A} \subseteq \mathcal{A}$, for every $A(v_{\tau(a)}) \in \mathcal{B}'$ or $A(w_{\tau(a)}^R) \in \mathcal{B}'$, we have $A(a) \in \mathcal{A}$ for some $a \in \text{ind}(\mathcal{A})$. Hence, we also obtain $\mathfrak{N}(\mathcal{B}') \subseteq \mathfrak{N}(\mathcal{A})$. Thus, $\mathfrak{N}(\mathcal{A}) = \mathfrak{N}(\mathcal{B}')$, which allows us to apply Lemma 6. By Theorem 5, \mathcal{B}' is closed under the rules in Table 2. Therefore, by Lemma 6, \mathcal{A} is closed under those rules. Consequently, by Theorem 5, \mathcal{O} is materialized. \square

Implementation and Evaluation

We implemented a prototype system Orar for full materialization of Horn *SHOIF* ontologies, evaluated Orar on popular ontologies, and compared it with other reasoners.

Table 3 presents detailed information about the test ontologies and the experimental results. NPD is an ontology about petroleum activities, DBPedia⁺ (DBP⁺) is an extension of the DBPedia ontology, and IMDb⁺ consists of the Movie ontology and the dataset from the IMDb website. For the popular benchmarks LUBM (Guo, Pan, and Heflin 2005) and UOBM (Ma et al. 2006), we use L_n and U_n to denote the datasets for n universities respectively. LUBM is in Horn *SHI* and IMDb⁺ is in Horn *SHIF*. For the other ontologies, we extracted the relevant Horn fragment by removing axioms with disjunctions; NPD is in Horn *SHIF*, DBPedia⁺ and UOBM are in Horn *SHOIF*. The test ontologies and our system are available online.¹

For each ontology, Orar reached the fixpoint after at most two abstraction steps. As seen in Table 3, the sizes of the abstractions are significantly smaller than the sizes of the original ABoxes. For LUBM and IMDb⁺, the reduction is over four orders of magnitude. LUBM is also known to be

easily handled by other related approaches such as SHER. For NPD, the reduction is also significant: the abstraction is just 2% of the size of the original ABox. UOBM and DBPedia⁺ are more challenging as the ontologies additionally have nominals. For U10, the size of the abstract ABox is approximately 19% of the size of the original one. Unsurprisingly, the size reductions improve with the sizes of datasets, e.g., the abstraction for U500 is less than 2% in the size of the original ABox. For DBPedia⁺, the abstract ABox is merely 15% of the size of the original one. This is because individuals in this ontology are more diverse due to the relatively large number of atomic concepts and roles.

The last part of Table 3 aims at providing a comparison of reasoning times with and without abstraction. We limit this comparison to the reasoners Konclude 0.6.2 (Steigmiller, Liebig, and Glimm 2014) and PAGOdA 2.0 (Zhou et al. 2015), which we found to perform best for our test ontologies (see the technical report for evaluations of other reasoners). Unfortunately, even with Konclude and PAGOdA we had difficulties in computing full materialization. The reasoning results of PAGOdA (P_{reas}) do not include equality materialization. Konclude can only be controlled via the OWLLink API (Liebig et al. 2011), which causes a significant communication overhead due to the large number of individuals. In particular, Konclude timed out for all our test ontologies. Instead, for reference, we provide the (incomplete) results of the command-line client of Konclude (K_{reas}^{CLI}), which does not use OWLLink, but supports only concept materialization. All results were obtained using a compute server with two Intel Xeon E5-2660V3 processors and 512 GB RAM and a timeout of five hours.

We present the full reasoning times of Orar (O_{reas}) excluding loading time (O_{load}) and including reasoning time of Konclude via OWLLink on the abstract ABoxes (K_{reas}^{abst}). Currently, the computation of abstractions is not optimized. In particular, the total reasoning time is dominated by copying the entailed assertions from the abstract ABoxes to the original ABox. This step could be avoided by recomputing the abstractions directly. Also, unlike PAGOdA and Konclude, our implementation is not parallel. In any case, the purpose of our evaluation was not to show the superiority

¹<https://www.uni-ulm.de/en/in/ki/software/orar>

of Orar, but to see if our approach can improve the performance of existing reasoners on large data sets. As the abstract ABoxes are often significantly smaller than the original ones, directly implementing the technique in existing reasoners could bring even better improvements.

Discussion and Future Work

The presented approach for *full* materialization of Horn *SHOLF* ontologies results in abstractions that are significantly smaller than the original ontologies (by orders of magnitude) and can be computed efficiently. This is despite the more complex structure of the abstractions and the additionally required closure rules to achieve completeness in the presence of nominals, inverse roles, and functionality.

A remaining challenge is the extension to non-Horn ontologies, which is non-trivial since reasoners do not communicate non-deterministic information.

References

- Abiteboul, S.; Hull, R.; and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.
- Dolby, J.; Fokoue, A.; Kalyanpur, A.; Schonberg, E.; and Srinivas, K. 2009. Scalable Highly Expressive Reasoner (SHER). *J. of Web Semantics* 7(4):357–361.
- Glimm, B.; Kazakov, Y.; Liebig, T.; Tran, T.; and Vialard, V. 2014. Abstraction Refinement for Ontology Materialization. In *Proc. of the 13th Int. Semantic Web Conference, ISWC 2014*, 180–195.
- Glimm, B.; Kazakov, Y.; and Tran, T.-K. 2017. Ontology Materialization by Abstraction Refinement in Horn *SHOLF*. Technical report, University of Ulm. Available at <https://www.uni-ulm.de/en/in/ki/publications>.
- Guo, Y.; Pan, Z.; and Heflin, J. 2005. LUBM: A Benchmark for OWL Knowledge Base Systems. *J. Web Sem.* 3(2-3):158–182.
- Horrocks, I.; Kutz, O.; and Sattler, U. 2006. The Even More Irresistible *SROLQ*. In *Proc. of 10th International Conference on Principles of Knowledge Representation and Reasoning, KR 2006*, 57–67.
- Kalyanpur, A.; Parsia, B.; Horridge, M.; and Sirin, E. 2007. Finding all justifications of OWL DL entailments. In *Proc. of the 6th Int. Semantic Web Conference, ISWC 2007*, 267–280.
- Kazakov, Y. 2009. Consequence-Driven Reasoning for Horn *SHIQ* Ontologies. In *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence, IJCAI 2009*, 2040–2045.
- Kontchakov, R.; Lutz, C.; Toman, D.; Wolter, F.; and Zhakharyashev, M. 2011. The Combined Approach to Ontology-Based Data Access. In *Proc. of the 22nd International Joint Conference on Artificial Intelligence, IJCAI 2011*, 2656–2661.
- Krötzsch, M.; Rudolph, S.; and Hitzler, P. 2013. Complexities of Horn Description Logics. *ACM Trans. Comput. Log.* 14(1).
- Liebig, T.; Luther, M.; Noppens, O.; and Wessel, M. 2011. OWLlink. *Semantic Web* 2(1):23–32.
- Ma, L.; Yang, Y.; Qiu, Z.; Xie, G. T.; Pan, Y.; and Liu, S. 2006. Towards a Complete OWL Ontology Benchmark. In *Proc. of European Semantic Web Conference, ESWC 2006*, 125–139.
- Motik, B.; Nenov, Y.; Piro, R.; Horrocks, I.; and Olteanu, D. 2014. Parallel Materialisation of Datalog Programs in Centralised, Main-Memory RDF Systems. In *Proc. of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2014*, 129–137.
- Ortiz, M.; Rudolph, S.; and Simkus, M. 2010. Worst-Case Optimal Reasoning for the Horn-DL Fragments of OWL 1 and 2. In *Proc. of the Twelfth International Conference on Principles of Knowledge Representation, KR 2010*.
- Steigmiller, A.; Liebig, T.; and Glimm, B. 2014. Konclude: System Description. *J. Web Sem.* 27:78–85.
- Urbani, J.; Kotoulas, S.; Maassen, J.; van Harmelen, F.; and Bal, H. E. 2012. WebPIE: A Web-scale Parallel Inference Engine using MapReduce. *J. Web Semantics* 10:59–75.
- Wandelt, S., and Möller, R. 2012. Towards ABox Modularization of Semi-Expressive Description Logics. *J. of Applied Ontology* 7(2):133–167.
- Zhou, Y.; Grau, B. C.; Nenov, Y.; Kaminski, M.; and Horrocks, I. 2015. PAGOdA: Pay-As-You-Go Ontology Query Answering Using a Datalog Reasoner. *J. Artif. Intell. Res. (JAIR)* 54:309–367.