Incremental Materialization Update via Abstraction Refinement

Markus Brenner and Birte Glimm

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. In this paper, we show how abstraction refinement can be adopted for incremental ABox materialization by combining it with the well-known DRed algorithm for materialization maintenance. Similarly to reasoning using abstraction refinement, we can perform computationally expensive operations such as overdeletion on smaller, disconnected abstractions instead of operating on a large ABox. We also show that the obtained procedure is sound and complete for Horn $\mathcal{ALCHI}$ ontologies.

1 Introduction

Most ontology reasoners support the task of materialization (i.e., they compute and explicitly store all entailed atomic concept and role assertions for the individuals in the ontology), which allows for the evaluation of conjunctive instance queries directly over the stored facts. Computing the materialisation is computationally expensive and approaches such as summarization [3], ABox modularization [16], or abstraction refinement [5] attempt to "compress" the size of the dataset over which the materialization is computed, thereby enabling processing of large ABoxes and reducing the number of recurring reasoning steps. Even using efficient materialization techniques, recomputing all consequences whenever input data changes can cause a significant delay before user queries can be answered again, which might be prohibitive for some application scenarios. Incremental maintenance algorithms originating from the database and Datalog communities (see, e.g., [14, 10]) have been applied to description logics and the semantic web for incremental classification [6, 13], incremental materialization via Datalog [15] and RDF stream reasoning [2]. While the basic idea of incremental maintenance algorithms is generally applicable, the presented algorithms so far focus on and are optimized for ontologies that can be expressed in the form of (Datalog) rules, i.e., proper existentials are not supported. Furthermore, how incremental maintenance algorithms can be combined with data compression techniques is an open problem, only addressed in a sketchy way by Steigmiller et al. [12] in the form of a representative cache maintaining individuals in an incremental fashion.

In this paper we address this problem by combining the abstraction refinement technique for materialization of Horn $\mathcal{ALCHI}$ ontologies with the well-known Delete/Rederive (DRed) algorithm for materialization management.

2 Preliminaries

The syntax of $\mathcal{ALCHI}$ is defined using a vocabulary consisting of countably infinite disjoint sets $\mathsf{N_C}$ of *atomic concepts*, $\mathsf{N_R}$ of *atomic roles*, and $\mathsf{N_I}$ of *individuals*. A *role* is either an atomic role or an *inverse role* r^- with $r \in \mathsf{N_R}$. We define $R^- := r^-$ if $R = r$ and $R^- := r$ if $R = r^-$. For general semantics and complex *concepts* and *axioms*, we conform to the standard definitions and refer the reader to e.g. Baader et al. [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 and role assertions in $\mathcal{O}$ and $\mathcal{T}$ a *TBox* consisting of the concept and role inclusion axioms in $\mathcal{O}$. We also refer to concept and role assertions simply as assertions. To simplify the presentation, we do not distinguish between axioms $R(a, b)$, $R \sqsubseteq S$ and, respectively, $R^-(b, a)$, $R^- \sqsubseteq S^-$. We use $\mathsf{con}(\mathcal{O})$, $\mathsf{rol}(\mathcal{O})$, and $\mathsf{ind}(\mathcal{O})$ for the sets of atomic concepts, atomic roles, and individuals occurring in $\mathcal{O}$, respectively. Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, an *ABox justification* w.r.t. $\mathcal{T}$ for an axiom α with $\mathcal{O} \models \alpha$ is any set $J \subseteq \mathcal{A}$ s.t. $J \cup \mathcal{T} \models \alpha$. An ABox justification J is *minimal*, if $\forall J' \subset J : J' \cup \mathcal{T} \not\models \alpha$. An $\mathcal{ALCHI}$ ontology $\mathcal{O}$ is *Horn* [9] and in *normalized* form if, for every $C(a) \in \mathcal{O}$, C is an atomic concept and every concept inclusion $C \sqsubseteq D \in \mathcal{O}$, is in one of the following forms $\top \sqsubseteq A, A \sqsubseteq B, A \sqsubseteq \exists R.B, A \sqsubseteq \bot, A \sqcap B \sqsubseteq C, A \sqsubseteq \forall R.B$ where A, B, and C are atomic concepts and R is a role. W.l.o.g., we assume in the remainder every ontology as normalized by applying a structural transformation; see e.g. [8].

For an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, we say that $\mathcal{A}$ is *concept-materialized* w.r.t. $\mathcal{T}$, if $\mathcal{O} \models A(a)$ implies $A(a) \in \mathcal{A}$ for each $A \in \mathsf{con}(\mathcal{O})$ and $a \in \mathsf{ind}(\mathcal{O})$; $\mathcal{A}$ is *role-materialized* w.r.t. $\mathcal{T}$ if $\mathcal{O} \models r(a, b)$ implies $r(a, b) \in \mathcal{A}$ for each $r \in \mathsf{rol}(\mathcal{O})$ and $a, b \in \mathsf{ind}(\mathcal{O})$; $\mathcal{A}$ is (fully) *materialized* w.r.t. $\mathcal{T}$ if it is concept-, and role-materialized. Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the concept-, role-, and/or (full) materialization of $\mathcal{A}$ w.r.t. $\mathcal{T}$ of is the smallest super-set of $\mathcal{A}$ that is concept-, role-, and/or fully materialized w.r.t. $\mathcal{T}$, respectively. Note that the full materialization of $\mathcal{A}$ w.r.t. $\mathcal{T}$ is always finite since the sets $\mathsf{con}(\mathcal{O})$, $\mathsf{rol}(\mathcal{O})$ and $\mathsf{ind}(\mathcal{O})$ are finite. Since the role materialization of an $\mathcal{ALCHI}$ ontology can be determined quickly using a precomputed role hierarchy, we focus on concept materialization in the remainder.

2.1 Abstraction Refinement

The main idea of the abstraction refinement method [4, 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 transferring the new entailments back to $\mathcal{O}$. The ABox $\mathcal{B}$ is usually called the *abstraction* of the *original* ABox $\mathcal{A}$. For transferring back entailments, we use homomorphisms:

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

Entailments can be transferred due to the following property.

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 $\mathcal{A} \cup \mathcal{T} \models h(\beta)$.*

Suitable abstractions for $\mathcal{ALCHI}$ can be constructed by considering asserted roles and concepts for single individuals using *types*:

Definition 2 (Type). *Let $\mathcal{A}$ be an ABox and a an individual. The* concept type *of a w.r.t. $\mathcal{A}$ is a set of concepts $\tau_C(a) = \{C \mid C(a) \in \mathcal{A}\}$. The* role type *of a w.r.t. $\mathcal{A}$ is a set of roles $\tau_R(a) = \{R \mid \exists b : R(a, b) \in \mathcal{A}\}$. The* (combined) type *of a w.r.t. $\mathcal{A}$ is a pair (τ_C, τ_R), where $\tau_C(a)$ and $\tau_R(a)$ are the concept and role type of a w.r.t. $\mathcal{A}$, respectively.*

Example 1. Let $\mathcal{A} = \{A(a), A(b), R(a, b)\}$. Then $\tau_C(a) = \tau_C(b) = \{A\}$, $\tau_R(a) = \{R\}$, $\tau_R(b) = \{R^-\}$, $\tau(a) = \tau_1 = \langle\{A\}, \{R\}\rangle$, and $\tau(b) = \tau_2 = \langle\{A\}, \{R^-\}\rangle$.

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

Definition 3 (Abstraction). *Let $\mathcal{A}$ be an ABox and $\tau = \langle\tau_C, \tau_R\rangle$ a type. The* abstraction for τ *is an ABox $\mathcal{B}_\tau = \{C(v_\tau) \mid C \in \tau_C\} \cup \{R(v_\tau, w_\tau^R) \mid R \in \tau_R\}$, where v_τ and w_τ^R are distinguished* abstract individuals *for the type τ. The* abstraction *of $\mathcal{A}$ is $\mathcal{B} = \bigcup_{a\in\mathsf{ind}(\mathcal{A})} \mathcal{B}_{\tau(a)}$, where $\tau(a)$ is the type for a w.r.t. $\mathcal{A}$.*

Example 2. The abstraction for $\mathcal{A}$ in Example 1 is $\mathcal{B} = \mathcal{B}_{\tau(a)} \cup \mathcal{B}_{\tau(b)}$, where $\mathcal{B}_{\tau(a)} = \mathcal{B}_{\tau_1} = \{A(v_{\tau_1}), R(v_{\tau_1}, w_{\tau_1}^R)\}$, $\mathcal{B}_{\tau(b)} = \mathcal{B}_{\tau_2} = \{A(v_{\tau_2}), R^-(v_{\tau_2}, w_{\tau_2}^{R^-})\}$.

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

$$h(v_\tau) \in \{a \in \mathsf{ind}(\mathcal{A}) \mid \tau(a) = \tau\}, \tag{1}$$

$$h(w_\tau^R) \in \{b \in \mathsf{ind}(\mathcal{A}) \mid R(h(v_\tau), b) \in \mathcal{A}\}, \tag{2}$$

is a homomorphism from $\mathcal{B}$ to $\mathcal{A}$, which allows us to transfer entailments from the abstraction back to the original ABox (Lemma 1). We formalize these transferred entailments using Definition 4.

Definition 4. *Let $\mathcal{B}$ be the abstraction of an ABox $\mathcal{A}$ (according to Definition 3), $\mathcal{B}^\infty$ be the (concept-)materialization of $\mathcal{B} \cup \mathcal{T}$, and $\Delta\mathcal{B} = \mathcal{B}^\infty \setminus \mathcal{B}$. The* update of $\mathcal{A}$ using $\Delta\mathcal{B}$ *is the smallest set of assertions $\Delta\mathcal{A}$ such that:*

$$C(v_{\tau(a)}) \in \Delta\mathcal{B} \quad\Rightarrow\quad C(a) \in \Delta\mathcal{A}, \tag{3}$$

$$R(a, b) \in \mathcal{A},\ C(w_{\tau(a)}^R) \in \Delta\mathcal{B}, \quad\Rightarrow\quad C(b) \in \Delta\mathcal{A}. \tag{4}$$

$$\mathbf{R}_{\sqsubseteq}\ \frac{A_1(a) \quad \cdots \quad A_n(a)}{B(a)} : \mathcal{T} \models A_0 \sqcap \ldots \sqcap A_n \sqsubseteq B$$

$$\mathbf{R}_{\forall}\ \frac{A_1(a) \quad \cdots \quad A_n(a) \quad R(a,b)}{B(b)} : \mathcal{T} \models A_0 \sqcap \ldots \sqcap A_n \sqsubseteq \forall S.B \text{ and } \mathcal{T} \models R \sqsubseteq S$$

Fig. 1. Materialization rules for Horn $\mathcal{ALCHI}$ ontologies; A and B are atomic concepts; R, S are roles; a, b are individuals

Using the definitions above, the *Abstraction Refinement method* for reasoning over a Horn $\mathcal{ALCHI}$ ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ can be summarized as follows:

AR1 Build the abstraction $\mathcal{B}$ of $\mathcal{A}$ using Definition 3

AR2 Determine the update $\Delta\mathcal{A}$ of $\mathcal{A}$ according to Definition 4, using a reasoner on $\mathcal{B} \cup \mathcal{T}$ and extend $\mathcal{A}$ with $\Delta\mathcal{A}$.

AR3 Repeat from Step **AR1** until no new entailments can be added to $\mathcal{A}$.

Steps **AR1** and **AR2** are repeated until the procedure reaches a fix-point. The method is sound, complete, and terminating for Horn $\mathcal{ALCHI}$ and, with some extensions, even for Horn $\mathcal{SHOIF}$ [5].

3 Combining Abstraction Refinement with DRed

For computing the materialization of a Horn $\mathcal{ALCHI}$ ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, we compute the closure of the ABox assertions using a slightly modified version of the materialization rules given by Glimm et al. [5] for Horn $\mathcal{SHOIF}$ restricted to Horn $\mathcal{ALCHI}$ as shown in Figure 1. Premises are given above the horizontal line and the conclusions below. Side conditions are given after the colon and restrict the expressions to which the rules are applicable. For example, rule $\mathbf{R}_{\sqsubseteq}$ produces one inference for each individual a and concepts $A_1, \ldots, A_n, B$ such that $\mathcal{T} \models A_0 \sqcap \ldots \sqcap A_n \sqsubseteq B$ with the premise $\{A_1(a), \ldots, A_n(a)\}$ and the conclusion $B(a)$. Note that the axioms in the TBox $\mathcal{T}$ are only used in side conditions and never as premises of the rules, which allows us to focus on ABox reasoning and to leave TBox reasoning to a suitable reasoner. Lemma 2 can be shown very similarly (although in a much simpler way) to the proof used by Glimm et al. [5].

Lemma 2. *The rules in Figure 1 are sound and complete for the concept-materialization of a normalized Horn $\mathcal{ALCHI}$ ontology.*

3.1 The Delete/Rederive Algorithm

While one can examine a variety of different knowledge base changes in the DL setting, we choose to adapt the database-based view, in which only known ABox facts can be added or deleted. This view already provides enough expressivity for a variety of use cases, e.g. stream reasoning [2]. Additionally, as additions of ABox facts can be handled by simply 'continuing' materialization using the new facts due to the monotonicity of DL reasoning, we also focus on deletions in the remainder. More precisely, we assume the following setting and notation in the remainder of the paper: Given a normalized $\mathcal{ALCHI}$ ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the materialization $\mathcal{A}^\infty$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$, and a set of assertions $\mathcal{A}^- \subseteq \mathcal{A}$ to be deleted, we want to determine the materialization of $\mathcal{A} \setminus \mathcal{A}^-$ w.r.t. $\mathcal{T}$ using $\mathcal{A}^\infty$. Doing so requires the identification and removal of assertions in $\mathcal{A}^\infty$ no longer derivable from $\mathcal{A} \setminus \mathcal{A}^-$, which is a rather difficult task. The Delete/Rederive (DRed) algorithm [7, 11] therefore initially overestimates the necessary deletions and then determines facts still derivable from $\mathcal{A} \setminus \mathcal{A}^-$. This overestimation is obtained by continuously overdeleting facts, which could be derived from already overdeleted facts. We formalize this behavior by defining a form of restricted rule applications.

Definition 5 (Restricted Derivations). *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ and an ABox $\mathcal{A}'$, an axiom α* directly follows from $\mathcal{A} \cup \mathcal{T}$ under restriction $\mathcal{A}'$*, if either $\alpha \in \mathcal{A}'$ or if a rule from Figure 1 can be applied to $\mathcal{A}$ w.r.t $\mathcal{T}$ to derive α and at least one of the premises is in $\mathcal{A} \cap \mathcal{A}'$. If α* directly follows from $\mathcal{A} \cup \mathcal{T}$ under restriction $\mathcal{A}'$*, we write $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{A}'} \alpha$.*

For the presentation of the DRed algorithm, we follow the presentation style of Motik et al. [10], but we adapt it to a consequence-based calculus. This avoids complications with ensuring termination of a Datalog program in the presence of function symbols, which are required to translate existential quantifiers. The algorithm further uses several auxiliary sets: the set $\mathcal{D}_{\text{all}}$ accumulates facts that might need to be removed from $\mathcal{A}^\infty$ due to the removal of $\mathcal{A}^-$, the set $\mathcal{D}_{\text{new}}$ gathers facts derived in the current iteration of rule application, and the set $\mathcal{D}_{\text{prev}}$ contains facts derived in the previous iteration of rule application. The DRed algorithm determines $\mathcal{A}^\infty_{\text{new}}$ as the (concept-)materialization of $(\mathcal{A} \setminus \mathcal{A}^-)$ w.r.t. $\mathcal{T}$ as follows:

DR1 Set $\mathcal{A} = \mathcal{A} \setminus \mathcal{A}^-$, set $\mathcal{D}_{\text{new}} = \mathcal{A}^-$, and set $\mathcal{D}_{\text{all}} = \emptyset$.

DR2 Set $\mathcal{D}_{\text{prev}} = \mathcal{D}_{\text{new}} \setminus \mathcal{D}_{\text{all}}$. If $\mathcal{D}_{\text{prev}} = \emptyset$, then go to step **DR4**;

DR3 Set $\mathcal{D}_{\text{all}} = \mathcal{D}_{\text{all}} \cup \mathcal{D}_{\text{prev}}$, set $\mathcal{D}_{\text{new}}$ to the set of all assertions α with $\mathcal{A}^\infty \vdash_{\mathcal{D}_{\text{prev}}} \alpha$. Repeat step **DR2**.
After this step, $\mathcal{D}_{\text{all}}$ contains all facts that might need to be deleted, so steps **DR2**–**DR3** are called *overdeletion*.

DR4 Set $\mathcal{A}^\infty_{\text{new}} = \mathcal{A}^\infty \setminus \mathcal{D}_{\text{all}}$, and set $\mathcal{D}_{\text{new}} = \mathcal{D}_{\text{all}} \cap \mathcal{A}$.

DR5 Evaluate an extension of the rules in Figure 1 over $\mathcal{A}^\infty_{\text{new}}$ w.r.t. $\mathcal{T}$ such that each premise additionally contains its conclusion, but restricted to match assertions in $\mathcal{D}_{\text{all}}$ and such that the conclusion is added to $\mathcal{D}_{\text{new}}$.
This *rederivation* step adds to $\mathcal{D}_{\text{new}}$ facts with an alternate derivation in the updated $\mathcal{A}^\infty_{\text{new}}$. The additional premise restricted to matches in $\mathcal{D}_{\text{all}}$ restricts rederivation only to facts from steps **DR2**–**DR3**.

DR6 Set $\mathcal{D}_{\text{prev}} = \mathcal{D}_{\text{new}} \setminus \mathcal{A}^\infty_{\text{new}}$. If $\mathcal{D}_{\text{prev}} = \emptyset$, then terminate;

DR7 Set $\mathcal{A}^\infty_{\text{new}} = \mathcal{A}^\infty_{\text{new}} \cup \mathcal{D}_{\text{prev}}$, set $\mathcal{D}_{\text{new}}$ to the set of all assertions α with $\mathcal{A}^\infty_{\text{new}} \vdash_{\mathcal{D}_{\text{prev}}} \alpha$. Repeat step **DR6**. Steps **DR6**–**DR7** are called *(re)insertion*.

3.2 Incremental Materialization via Abstraction Refinement

We adopt DRed in the general abstraction refinement way: We construct, for each of the different phases, suitable abstractions of the ABox, on which we perform the overdeletion, rederivation and the reinsertion phase. Interleaved refinement steps (in case of overdeletion and reinsertion repeatedly, until a fixpoint is reached) transfer results back to the original ABox and yield an adapted abstraction.

While reasoning using standard abstraction refinement requires only knowledge about asserted concepts and roles, overdeleting and rederiving on abstractions additionally requires knowledge about the set of assertions that are to be deleted. To keep track of such assertions, we extend the definitions of types and abstractions to bi-types ans bi-abstractions.

Definition 6 (Bi-Type). *Given ABoxes $\mathcal{A}_1, \mathcal{A}_2$, the* bi-type *of an individual $a \in \mathsf{ind}(\mathcal{A}_1 \cup \mathcal{A}_2)$ w.r.t. $(\mathcal{A}_1, \mathcal{A}_2)$ is a quadruple $(\tau^1_C(a), \tau^1_R(a), \tau^2_C(a), \tau^2_R(a))$, where $(\tau^1_C(a), \tau^1_R(a))$ is the combined type of a w.r.t. $\mathcal{A}_1$ and $(\tau^2_C(a), \tau^2_R(a))$ is the combined type of a w.r.t. $\mathcal{A}_2$.*

Definition 7 (Bi-Abstraction). *Given two ABoxes $\mathcal{A}_1$, $\mathcal{A}_2$ and a bi-type $\tau = (\tau^1_C, \tau^1_R, \tau^2_C, \tau^2_R)$ w.r.t. $(\mathcal{A}_1, \mathcal{A}_2)$, the* bi-abstraction for *τ is an ABox $\mathcal{B}^1_\tau \cup \mathcal{B}^2_\tau$, where $\mathcal{B}^1_\tau = \{C(v_\tau) \mid C \in \tau^1_C\} \cup \{R(v_\tau, w^R_\tau) \mid R \in \tau^1_R\}$, $\mathcal{B}^2_\tau = \{C(v_\tau) \mid C \in \tau^2_C\} \cup \{R(v_\tau, w^R_\tau) \mid R \in \tau^2_R\}$, and v_τ and w^R_τ are distinguished* abstract individuals *for the bi-type τ. The* bi-abstraction *of $(\mathcal{A}_1, \mathcal{A}_2)$ is*
$\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2, \qquad \mathcal{B}^1 = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathcal{B}^1_{\tau(a)}, \qquad \mathcal{B}^2 = \bigcup_{a \in \mathsf{ind}(\mathcal{A})} \mathcal{B}^2_{\tau(a)},$
where $\tau(a)$ is the bi-type for a w.r.t. $(\mathcal{A}_1, \mathcal{A}_2)$ and $\mathcal{B}^1_{\tau(a)} \cup \mathcal{B}^2_{\tau(a)}$ is the bi-abstraction for $\tau(a)$.

The following example highlights, how bi-abstractions also differentiate types based on their (over-)deleted assertions, while still aggregating 'similar' cases.

Example 3. For $\mathcal{A} = \{A(a_1), R(a_1, b), A(a_2), R(a_2, b), A(a_3), R(a_3, b)\}$ and $\mathcal{A}^- = \{A(a_1), A(a_3)\}$, the combined type of a_1, a_2, and a_3 w.r.t. $\mathcal{A}$ is $(\{A\}, \{R\})$. To distinguish a_1 and a_3 from a_2, we consider the bi-types w.r.t. $(\mathcal{A} \setminus \mathcal{A}^-, \mathcal{A}^-)$ and obtain $\tau(a_1) = \tau(a_3) = \tau_1 = (\emptyset, \{R\}, \{A\}, \emptyset)$ and $\tau(a_2) = \tau_2 = (\{A\}, \{R\}, \emptyset, \emptyset)$; furthermore, $\tau(b) = \tau_3 = (\emptyset, \{R^-\}, \emptyset, \emptyset)$.

In our procedure, we use (similar to the DRed algorithm), auxiliary sets $\mathcal{D}_{\text{all}}$, $\mathcal{E}_{\text{all}}$, $\mathcal{D}_{\text{new}}$, $\mathcal{E}_{\text{new}}$, and $\mathcal{E}_{\text{prev}}$, where $\mathcal{E}_{\text{all}}$, $\mathcal{E}_{\text{new}}$ and $\mathcal{E}_{\text{prev}}$ represent sets used within abstractions. The Abstraction Refinement Delete and Rederive algorithm (ARDred) determines $\mathcal{A}^\infty_{\text{new}}$ as the (concept-)materialization of $\mathcal{A} \setminus \mathcal{A}^-$ w.r.t. $\mathcal{T}$ as follows:

ARD1 Set $\mathcal{D}_{\text{new}} = \mathcal{A}^-$ and set $\mathcal{D}_{\text{all}} = \emptyset$.
ARD2 If $\mathcal{D}_{\text{new}} \setminus \mathcal{D}_{\text{all}} = \emptyset$, then go to step **ARD4**.
ARD3 Set $\mathcal{D}_{\text{all}} = \mathcal{D}_{\text{all}} \cup \mathcal{D}_{\text{new}}$, set $\mathcal{D}_{\text{new}} = \emptyset$. Build the *bi-abstraction* $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ w.r.t. $(\mathcal{A}^\infty \setminus \mathcal{D}_{\text{all}}, \mathcal{D}_{\text{all}})$.
OD1 Set $\mathcal{E}_{\text{new}} = \mathcal{B}^2$ and set $\mathcal{E}_{\text{all}} = \emptyset$.

OD2 Set $\mathcal{E}_{\text{prev}} = \mathcal{E}_{\text{new}} \setminus \mathcal{E}_{\text{all}}$. If $\mathcal{E}_{\text{prev}} = \emptyset$, go to step **OD4**.

OD3 Set $\mathcal{E}_{\text{all}} = \mathcal{E}_{\text{all}} \cup \mathcal{E}_{\text{prev}}$, set $\mathcal{E}_{\text{new}}$ to the set of all assertions α with $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{T} \vdash_{\mathcal{E}_{\text{prev}}} \alpha$. Go to step **OD2**.

OD4 Determine $\mathcal{D}_{\text{new}}$ as the update of $\mathcal{D}_{\text{all}}$ according to Definition 4, using $\mathcal{E}_{\text{all}}$ as $\Delta\mathcal{B}$ and the role assertions of $\mathcal{A}^{\infty}$ in line 4 of the definition. Continue with step **ARD2**.

This concludes the *overdeletion* phase of the algorithm.

ARD4 Set $\mathcal{A}^{\infty}_{\text{new}} = \mathcal{A}^{\infty} \setminus \mathcal{D}_{\text{all}}$, set $\mathcal{D}_{\text{new}} = \mathcal{D}_{\text{all}} \cap (\mathcal{A} \setminus \mathcal{A}^{-})$.

ARD5 Let $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ be the *bi-abstraction* w.r.t. $(\mathcal{A}^{\infty}_{\text{new}}, \mathcal{D}_{\text{all}})$. Set $\mathcal{E}_{\text{new}} = \emptyset$ and evaluate the rules in Figure 1 w.r.t. $\mathcal{T}$ over $\mathcal{B}^1$, s.t. conclusions are added to $\mathcal{E}_{\text{new}}$ and $\mathbf{R}_{\sqsubseteq}$ additionally contains its conclusion as a premise restricted to match assertions in $\mathcal{B}^2$. Extend the assertions present in $\mathcal{D}_{\text{new}}$ with the update to $\mathcal{D}_{\text{new}}$ w.r.t. Definition 4, using $\mathcal{E}_{\text{new}}$ as $\Delta\mathcal{B}$ and the role assertions of $\mathcal{A} \setminus \mathcal{A}^{-}$ in line 4 of the Definition. Set $\mathcal{D}_{\text{new}} = \mathcal{D}_{\text{new}} \cap \mathcal{D}_{\text{all}}$.

This step is also referred to as the *rederivation* phase.

ARD6 Set $\mathcal{D}_{\text{prev}} = \mathcal{D}_{\text{new}} \setminus \mathcal{A}^{\infty}_{\text{new}}$. If $\mathcal{D}_{\text{prev}} = \emptyset$ terminate.

ARD7 Set $\mathcal{A}^{\infty}_{\text{new}} = \mathcal{A}^{\infty}_{\text{new}} \cup \mathcal{D}_{\text{new}}$. Construct the *bi-abstraction* $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ w.r.t. $(\mathcal{A}^{\infty}_{\text{new}}, \mathcal{D}_{\text{all}})$.

RD1 Set $\mathcal{E}_{\text{new}} = \mathcal{B}^1 \cap \mathcal{B}^2$, $\mathcal{E}_{\text{all}} = \emptyset$.

RD2 Set $\mathcal{E}_{\text{prev}} = \mathcal{E}_{\text{new}} \setminus \mathcal{E}_{\text{all}}$. If $\mathcal{E}_{\text{prev}} = \emptyset$, go to step **RD4**.

RD3 Set $\mathcal{E}_{\text{all}} = \mathcal{E}_{\text{all}} \cup \mathcal{E}_{\text{new}}$, set $\mathcal{E}_{\text{new}}$ to the set of all assertions α with $\mathcal{B}^1 \cup \mathcal{E}_{\text{all}} \cup \mathcal{T} \vdash_{\mathcal{E}_{\text{prev}}} \alpha$. Continue with step **RD2**.

RD4 Determine $\mathcal{D}_{\text{new}}$ as the update of $\mathcal{A}^{\infty}_{\text{new}}$ according to Definition 4, using $\mathcal{E}_{\text{all}}$ as $\Delta\mathcal{B}$ and the role assertions of $\mathcal{A}^{\infty}_{\text{new}}$ in line 4 of the Definition. Continue with step **ARD6**.

Steps **ARD6**–**RD4** form the *(re-)insertion* phase of the algorithm. Steps **RD1**–**RD3** form the *inner (re-)insertion phase*.

Similar to DRed, we have an overdeletion (steps **ARD1**–**OD4**), a rederivation (steps **ARD4**–**ARD5**) and a reinsertion phase (steps **ARD6**–**RD4**). Similar to abstraction refinement, instead of determining the overdeleted, rederived and reinserted assertions on the original ABox, we determine assertions on suitable bi-abstractions and transfer the obtained results back, in case of overdeletion and reinsertion repeatedly, until a fixpoint is reached.

The use of bi-types and bi-abstractions allows us to transfer the deletion set into $\mathcal{B}$ as $\mathcal{B}^2$. In the overdeletion phase, we (indirectly) extend $\mathcal{B}_2$ to ultimately extend the overdeletion $\mathcal{D}_{\text{all}}$ and construct the bi-abstraction over $(\mathcal{A}^{\infty} \setminus \mathcal{D}_{\text{all}}, \mathcal{D}_{\text{all}})$, such that $\mathcal{B}$ is still the abstraction of the complete materialization (as otherwise, not all assertions could be derived). The rederivation phase uses $\mathcal{B}^2$ to restrict possible derivations of rule $\mathbf{R}_{\sqsubseteq}$. Finally, the reinsertion phases determines rederived assertions using the 'overlap' between $\mathcal{B}^1$ (created from $\mathcal{A}^{\infty}_{\text{new}}$) and $\mathcal{B}^2$, which allows us to reduce the number of possible derivations.

Note that there is a slight conceptual difference in the overdeletion part of DRed (which always operates on a complete (concept-)materialization of an ABox) and the inner overdeletion (steps **OD1**–**OD4**), as the latter operates on $\mathcal{B}$ and not the (concept-)materialization of $\mathcal{B}$. Thus, the inner overdeletion of ARDred also extends $\mathcal{E}_{\text{all}}$ with derivable assertions not in $\mathcal{B}$.

Sound- and completeness of the algorithm strongly depends on the overdeletion phase (steps **ARD1–OD4**) for soundness (i.e. removing all axioms, which are no longer derivable) and the reinsertion phase (steps **ARD6–RD4**) for completeness (i.e. rederiving all axioms that are still derivable).

First, we want to fix the notion of overdeletion for the following discussions.

Definition 8 (Overdeletion). *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be an ontology, $\mathcal{A}^- \subseteq \mathcal{A}$ a set of assertions to be deleted and $\mathcal{A}^\infty$ the (concept-)materialization of $\mathcal{A}$ w.r.t. $\mathcal{T}$. A set $\mathcal{D}_{all} \subseteq \mathcal{A}^\infty$ is an* overdeletion *of $\mathcal{A}^\infty$ w.r.t. $\mathcal{A}^-$, if, for all $\alpha \in \mathcal{D}_{all}$, there is a minimal ABox justification $J \subseteq \mathcal{A}$ for α w.r.t. $\mathcal{T}$, s.t. $J \cap \mathcal{A}^- \neq \emptyset$.*

As soundness of the algorithm is related to completeness of the overdeletion phase of the algorithm, we can in the following focus on two completeness proofs. To do so, we first consider some properties of the restricted derivation relation of Definition 5. We can then show that determining a complete closure for restricted derivations in an abstraction refinement way ensures a similar closure over the original ABoxes. Finally, we use the obtained results to show completeness w.r.t. the base task, which is, respectively, overdeletion and rederivation.

Lemma 3 (Properties of Direct Restricted Derivations). *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be an ontology and $\mathcal{A}_1, \mathcal{A}_2$ ABoxes with $\mathcal{A}_1 \subseteq \mathcal{A}_2$, $\mathsf{ind}(\mathcal{A}_1) \subseteq \mathsf{ind}(\mathcal{A}_2) \subseteq \mathsf{ind}(\mathcal{A})$ and $\mathcal{B}$ an ABox, s.t. there is a homomorphism $h : \mathsf{ind}(\mathcal{A}) \to \mathsf{ind}(\mathcal{B})$ from $\mathcal{A}$ to $\mathcal{B}$. Then the following properties hold for any assertion α:*

$$\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{A}_1} \alpha \text{ implies } \mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{A}_2} \alpha. \qquad \textit{(Monotonicity)} \tag{5}$$

$$\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{A}_1} \alpha \text{ implies } \mathcal{B} \cup \mathcal{T} \vdash_{h(\mathcal{A}_1 \cap \mathcal{A})} h(\alpha). \tag{6}$$

Proof (Sketch). Both properties can easily be verified using the assumptions, Definitions 1 and 5 and the rules in Figure 1. □

Using Definition 5, we can describe the fixpoint constructed by the overdeletion and reinsertion phases as follows: For $\mathcal{B}$ the bi-abstraction w.r.t. $(\mathcal{A}^\infty \setminus \mathcal{D}_{\text{all}}, \mathcal{D}_{\text{all}})$ (Step **ARD3**), the inner overdeletion phase extends $\mathcal{B}^2$ with assertions α to determine a closure over $\mathcal{B}^1 \cup \mathcal{B}^2 \cup \mathcal{T} \vdash_{\mathcal{B}^2} \alpha$. Similarly, in the inner reinsertion phase, for $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ the bi-abstraction w.r.t. $(\mathcal{A}^\infty_{\text{new}}, \mathcal{D}_{\text{all}})$ (Step **ARD7**), $\mathcal{B}^1$ is extended with assertions α to construct a closure over $\mathcal{B}^1 \cup \mathcal{T} \vdash_{\mathcal{B}^1 \cap \mathcal{B}^2} \alpha$.

In both cases, the results of constructing the closure are transferred back to the original ABoxes using homomorphisms and updates. We proceed to show (roughly similar to the completeness proof of the original abstraction refinement approach [4]), that if no new assertions are transferred back to the original ABoxes, a corresponding fixpoint on those original ABoxes ($\mathcal{D}_{\text{all}}$ in case of the overdeletion phase, $\mathcal{A}^\infty_{\text{new}}$ in case of the reinsertion phase) has been reached.

Theorem 1 (Overdeletion Fixpoint Theorem). *Let $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$ be a normalized Horn $\mathcal{ALCHI}$ ontology, $\mathcal{A}^\infty$ the (concept-)materialization of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and $\mathcal{A}^- \subseteq \mathcal{A}$ a set of axioms to be deleted. Let further $\mathcal{D}_{all}$ be an ABox with $\mathcal{A}^- \subseteq \mathcal{D}_{all} \subseteq \mathcal{A}^\infty$ and let $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ be the bi-abstraction w.r.t. $(\mathcal{A}^\infty \setminus \mathcal{D}_{all}, \mathcal{D}_{all})$.*

Then it holds for all α with $\mathcal{A}^\infty \cup \mathcal{T} \vdash_{\mathcal{D}_{all}} \alpha$, that $\alpha \in \mathcal{D}_{all}$, if for every bi-type $\tau = (\tau_C^1, \tau_R^1, \tau_C^2, \tau_R^2)$ of an individual $a \in \mathsf{ind}(\mathcal{A})$, every atomic concept $A \in \mathsf{con}(\mathcal{O})$ and every role $R \in \mathsf{rol}(\mathcal{O})$, we have

1. *$\mathcal{B}^1 \cup \mathcal{B}^2 \cup \mathcal{T} \vdash_{\mathcal{B}^2} A(v_\tau)$ implies $A(a) \in \mathcal{D}_{all}$.*
2. *$\mathcal{B}^1 \cup \mathcal{B}^2 \cup \mathcal{T} \vdash_{\mathcal{B}^2} A(w_\tau^R)$ and $R(a,b) \in \mathcal{A}$ implies $A(b) \in \mathcal{D}_{all}$.*

Proof (Sketch). Extend $\mathcal{B} = \mathcal{B}^1 \cup \mathcal{B}^2$ with new role assertions to $\mathcal{B}' = \mathcal{B}^{1'} \cup \mathcal{B}^{2'}$ as follows: If $R(a,b) \in \mathcal{A}^\infty \setminus \mathcal{D}_{\text{all}}$, extend $\mathcal{B}^1$ with $R(v_{\tau(a)}, v_{\tau(b)})$. If $R(a,b) \in \mathcal{A}^-$, extend $\mathcal{B}^2$ with $R(v_{\tau(a)}, v_{\tau(b)})$. Note that there is a homomorphism from $\mathcal{A}^\infty$ to $\mathcal{B}'$ and thus, according to Property 6 of Lemma 3, $\mathcal{B}' \cup \mathcal{T} \vdash_{h(\mathcal{D}_{\text{all}})} \alpha$, if $\mathcal{A} \cup \mathcal{T} \vdash_{\mathcal{D}_{\text{all}}} \alpha$. We then show that $h(\mathcal{D}_{\text{all}}) \subseteq \mathcal{B}^{2'}$ and, therefore, $\mathcal{B}' \cup \mathcal{T} \vdash_{\mathcal{B}^{2'}} \alpha$. Thus, we are left to show, that the new role assertions provide no new derivations, as then the assumed properties assure the desired fixpoint. $\square$

We briefly sketch the idea behind the proof, that $\mathcal{B}'$ has the same atomic concept assertions as direct derivations under restriction $\mathcal{B}^{2'}$ as $\mathcal{B}$ under restriction $\mathcal{B}^2$. It suffices to consider four individuals v_1, v_2, w_1, w_2 of the bi-abstraction, s.t. $v_1 = v_{\tau(a)}, w_1 = w_{\tau(a)}^R, v_2 = v_{\tau(b)}, w_2 = w_{\tau(b)}^R$ and $R(a,b) \in \mathcal{A}^\infty$. By considering the construction of $\mathcal{B}'$ and $\mathcal{B}$ and the rules in Figure 1 together with the conditions of restricted derivations, we can easily show that introducing the new role assertion $R(v_1, v_2)$ does not result in new restricted entailments.

We can argue similar to the above, that $\mathcal{A}^\infty_{\text{new}}$ is determined as the closure w.r.t. $\mathcal{A}^\infty_{\text{new}} \cup \mathcal{T} \vdash_{\mathcal{A}^\infty_{\text{new}} \cap \mathcal{D}_{\text{all}}} \alpha$ by the reinsertion phase, which concludes the completeness proofs of the overdeletion and rederivation phases w.r.t. restricted derivations. It is left to show, that first of all, the results above are related to the task at hand (removing all axioms no longer entailed by $\mathcal{A} \setminus \mathcal{A}^-$ and rederiving all axioms still entailed by $\mathcal{A} \setminus \mathcal{A}^-$) and second, that the full algorithm is complete and sound w.r.t. the base task, which is, incrementally determining the changed materialization. We proceed to first examine the overdeletion phase based on the previous results and extend this examination into the soundness proof for ARDred. We then examine the reinsertion phase and its preceding operations to ultimately deliver the completeness proof for ARDred.

For the overdeletion phase, we use the bi-abstraction w.r.t. $(\mathcal{A}^\infty \setminus \mathcal{D}_{\text{all}}, \mathcal{D}_{\text{all}})$ and construct $\mathcal{D}_{\text{all}}$, such that $\mathcal{A}^\infty \cup \mathcal{T} \vdash_{\mathcal{D}_{\text{all}}} \alpha$ implies $\alpha \in \mathcal{D}_{\text{all}}$. In particular, initially $\mathcal{D}_{\text{all}} = \mathcal{A}^-$. Consider the following Lemma.

Lemma 4. *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the (concept-)materialization $\mathcal{A}^\infty$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and a set of axioms to be deleted $\mathcal{A}^- \subseteq \mathcal{A}$. Let $\mathcal{D}_{all}$ be a set s.t. $\mathcal{A}^- \subseteq \mathcal{D}_{all}$ and, if, for any concept or role assertion α, $\mathcal{A}^\infty \cup \mathcal{T} \vdash_{\mathcal{D}_{all}} \alpha$, then $\alpha \in \mathcal{D}_{all}$. Then $\mathcal{D}_{all}$ is an overdeletion of $\mathcal{A}^\infty$ w.r.t. $\mathcal{A}^-$ according to Definition 8.*

Proof (Sketch). Assuming the contrary (i.e. $\alpha \notin \mathcal{D}_{\text{all}}$), we argue by sound- and completeness of the rules in Figure 1, that α can be derived via a number of rule applications from a minimal ABox justification J (as in Def. 8) and thus there is a number of premises, of which at least one also has a minimal ABox justification $J' \subseteq J$ with $J' \cap \mathcal{A}^- \neq \emptyset$. By doing so repeatedly, we ultimately determine some assertion β, s.t. $\mathcal{A}^\infty \cup \mathcal{T} \vdash_{\mathcal{D}_{\text{all}}} \beta$ holds, which is a contradiction. $\square$

Thus, the overdeletion phase of the ARDred algorithm in steps **ARD1–OD4** produces a correct overdeletion $\mathcal{D}_{\text{all}}$ of $\mathcal{A}^{\infty}$ w.r.t. $\mathcal{A}^{-}$. Additionally, we remove this overdeletion from the materialization in step **ARD4**. Thus, it remains to be shown that no unsound assertions are reinserted afterwards.

Lemma 5. *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, a set of assertions to be deleted $\mathcal{A}^{-} \subseteq \mathcal{A}$ and the set $\mathcal{A}^{\infty}_{new}$ of the ARDred algorithm after step **ARD4**. If, during the execution of the ARDred algorithm, an axiom α is added to $\mathcal{A}^{\infty}_{new}$ after step **ARD4**, then $(\mathcal{A} \setminus \mathcal{A}^{-}) \cup \mathcal{T} \models \alpha$.*

Proof (Sketch). We argue that all newly added assertions are derived using sound rules on sound abstractions, but with additional restrictions. As the restrictions only remove derivations, all added assertions are sound. □

Soundness follows from the previous considerations concerning overdeletion.

Corollary 1 (Soundness of ARDred). *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the (concept-)materialization $\mathcal{A}^{\infty}$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and a set of axioms to be deleted $\mathcal{A}^{-} \subseteq \mathcal{A}$. Let $\mathcal{A}^{\infty}_{new}$ be the materialization obtained by executing the ARDred algorithm for $\mathcal{O}$, $\mathcal{A}^{\infty}$ and $\mathcal{A}^{-}$. Then $\alpha \in \mathcal{A}^{\infty}_{new}$ implies $(\mathcal{A} \setminus \mathcal{A}^{-}) \cup \mathcal{T} \models \alpha$.*

For the completeness of the reinsertion part of the algorithm, we need to additionally consider the initial contents of the sets $\mathcal{A}^{\infty}_{\text{new}}$ and $\mathcal{D}_{\text{all}}$, as they are used in the closure construction. As shown before, $\mathcal{D}_{\text{all}}$ contains the overdeletion of $\mathcal{A}^{\infty}$ w.r.t. $\mathcal{A}^{-}$. For $\mathcal{A}^{\infty}_{\text{new}}$, from step **ARD4** and step **ARD7**, we see that $\mathcal{A} \setminus \mathcal{A}^{-} \subseteq \mathcal{A}^{\infty}_{\text{new}}$. Further, we show, that step **ARD5** extends $\mathcal{A}^{\infty}_{\text{new}}$ with at least all assertions, which can be directly rederived from $\mathcal{A}^{\infty} \setminus \mathcal{D}_{\text{all}}$.

Lemma 6. *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the (concept-)materialization $\mathcal{A}^{\infty}$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and a set of axioms to be deleted $\mathcal{A}^{-} \subseteq \mathcal{A}$. Let further $\mathcal{D}_{all}$ be the overdeletion of $\mathcal{A}^{\infty}$ w.r.t. $\mathcal{A}^{-}$ and $\mathcal{A}^{\infty}_{new} = \mathcal{A}^{\infty} \setminus \mathcal{D}_{all}$. Then $\mathcal{D}_{new}$ is extended by step **ARD5** with all axioms $\alpha \notin \mathcal{A}^{\infty}_{new}$ obtained by applying one rule from Figure 1 to $\mathcal{A}^{\infty} \setminus \mathcal{A}^{\infty}_{new}$.*

Proof (Sketch). We obtain the desired result by examining all ways, in which a rule from Figure 1 can derive an assertion in one step and use the necessary preconditions to determine which assertions must be part of $\mathcal{B}^1$ and $\mathcal{B}^2$ as constructed by the algorithm in step **ARD5**. We then use this examination to show that all necessary assertions will also be derived on the abstraction (and be part of the generated update). □

Using these preconditions together with the established closure properties of the reinsertion phase of the algorithm, we can now formulate the following theorem, which rather directly entails the completeness of the ARDred algorithm.

Theorem 2. *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the (concept-)materialization $\mathcal{A}^{\infty}$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$, a set of axioms to be deleted $\mathcal{A}^{-} \subseteq \mathcal{A}$ and the overdeletion $\mathcal{D}_{all}$ of $\mathcal{A}^{\infty}$ w.r.t. $\mathcal{A}^{-}$. Let further $\mathcal{A}^{\infty}_{new}$ be the smallest set, s.t. $\mathcal{A}^{\infty} \setminus \mathcal{D}_{all} \subseteq \mathcal{A}^{\infty}_{new}$,*

$\mathcal{A}\setminus\mathcal{A}^- \subseteq \mathcal{A}^\infty_{new}$ and, if any concept assertion α can be derived from $\mathcal{A}^\infty \setminus \mathcal{D}_{all}$ by directly applying one of the rules from Figure 1 w.r.t. $\mathcal{T}$, then $\alpha \in \mathcal{A}^\infty_{new}$. Further extend $\mathcal{A}^\infty_{new}$ to the smallest set, s.t. if $\mathcal{A}^\infty_{new} \cup \mathcal{T} \vdash_{\mathcal{A}^\infty_{new}\cap\mathcal{D}_{all}} \alpha$ for any concept assertion α, then $\alpha \in \mathcal{A}^\infty_{new}$. Then, for any concept assertion α, $(\mathcal{A}\setminus\mathcal{A}^-)\cup\mathcal{T} \models \alpha$ implies $\alpha \in \mathcal{A}^\infty_{new}$.

Proof. As $\alpha \notin \mathcal{A}^\infty_{\text{new}}$ and due to the initial construction of $\mathcal{A}^\infty_{\text{new}}$, there must be some β, s.t. β is, so to speak, the initial culprit, i.e. $\beta \notin \mathcal{A}^\infty_{\text{new}}$ and β can be derived directly from $\mathcal{A}^\infty_{\text{new}}$ using a rule from Figure 1. We show that if there is any such β, then the conditions of the construction of $\mathcal{A}^\infty_{\text{new}}$ are violated. □

Completeness of ARDred directly follows from the previous considerations.

Corollary 2 (Completeness of ARDred). *Given an ontology $\mathcal{O} = \mathcal{A} \cup \mathcal{T}$, the (concept-)materialization $\mathcal{A}^\infty$ of $\mathcal{A}$ w.r.t. $\mathcal{T}$ and a set of axioms to be deleted $\mathcal{A}^- \subseteq \mathcal{A}$. Let $\mathcal{A}^\infty_{new}$ be the materialization obtained by executing the ARDred algorithm for $\mathcal{O}$, $\mathcal{A}^\infty$ and $\mathcal{A}^-$. Then $(\mathcal{A} \setminus \mathcal{A}^-) \cup \mathcal{T} \models \alpha$ implies $\alpha \in \mathcal{A}^\infty_{new}$.*

As both the overdeletion and reinsertion loop of the algorithm terminate after no new assertions could be determined in the previous iteration, we obtain the termination of the algorithm as a direct consequence of the sound- and completeness results. In particular, we terminate in the worst case, when $\mathcal{D}_{\text{all}} = \mathcal{A}^\infty$ (overdeletion) and $\mathcal{A}^\infty_{\text{new}} = \mathcal{A}^\infty$ (reinsertion).

4 Conclusion and Future Work

We have introduced a way to incrementally maintain the (concept-)materialization of an $\mathcal{ALCHI}$ ontology by combining DRed and abstraction refinement and using a consequence-based reasoning procedure. We have further shown that the presented procedure is sound and complete. Benefits of this approach lie in the summarization of similar deletion and reasoning tasks, paving the road for efficient maintenance of materializations of large ABoxes.

While the choosen presentation is appropriate for theoretical examinations, we will in the future focus on optimizations needed for a practical implementation and evaluation of the approach, to verify its practicality. For example, retaining results from previously materialized abstractions can reduce reasoning efforts, as results for unchanged types can be reused. Other optimizations will focus on better handling role deletions to avoid the currently large amount of unnecessary overdeletions (this can be done by either extending abstractions with a new role successor for deleted roles or by reducing role deletion to concept deletions using knowledge about propagated concepts) and on general ways to reduce the number of generated types while retaining sound- and completeness (one could, for example, only use the deletions of the previous abstraction-refinement round to generate bi-abstractions in the overdeletion phase), as a reduction in the number of types automatically entails a reduction in the overall computations. Finally, we want to incorporate approaches which tackle the general deficiencies of DRed, such as the Backward/Forward algorithm [10], which avoids unnecessary overdeletions by considering knowledge still asserted in $\mathcal{A} \setminus \mathcal{A}^-$.

References

1. Baader, F.: The description logic handbook: Theory, implementation and applications. Cambridge university press (2003)
2. Barbieri, D.F., Braga, D., Ceri, S., Della Valle, E., Grossniklaus, M.: Incremental reasoning on streams and rich background knowledge. In: Extended Semantic Web Conference. pp. 1–15. Springer (2010)
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. 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 (2014)
5. Glimm, B., Kazakov, Y., Tran, T.: Ontology materialization by abstraction refinement in horn SHOIF. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. pp. 1114–1120. AAAI Press (2017)
6. Grau, B.C., Halaschek-Wiener, C., Kazakov, Y., Suntisrivaraporn, B.: Incremental classification of description logics ontologies. Journal of Automated Reasoning 44(4), 337–369 (2010)
7. Gupta, A., Mumick, I.S., Subrahmanian, V.S.: Maintaining views incrementally. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data. pp. 157–166. SIGMOD '93, ACM (1993)
8. Kazakov, Y.: Consequence-Driven Reasoning for Horn $\mathcal{SHIQ}$ Ontologies. In: Proc. of the 21st Int. Joint Conf. on Artificial Intelligence, IJCAI 2009. pp. 2040–2045 (2009)
9. Krötzsch, M., Rudolph, S., Hitzler, P.: Complexities of Horn Description Logics. ACM Trans. Comput. Log. 14(1) (2013)
10. Motik, B., Nenov, Y., Piro, R.E.F., Horrocks, I.: Incremental update of datalog materialisation: the backward/forward algorithm. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence. pp. 1560–1568. AAAI Press (2015)
11. Staudt, M., Jarke, M.: Incremental maintenance of externally materialized views. In: Proceedings of the 22th International Conference on Very Large Data Bases. pp. 75–86. VLDB '96, Morgan Kaufmann Publishers Inc. (1996)
12. Steigmiller, A., Glimm, B., Liebig, T.: Completion graph caching for expressive description logics. In: Proceedings of the 28th International Workshop on Description Logics. CEUR Workshop Proceedings, vol. 1350. CEUR-WS.org (2015), http://ceur-ws.org/Vol-1350/paper-35.pdf
13. Suntisrivaraporn, B.: Module extraction and incremental classification: A pragmatic approach for el+ elˆ+ ontologies. The Semantic Web: Research and Applications pp. 230–244 (2008)
14. Volz, R., Staab, S., Motik, B.: Incremental maintenance of materialized ontologies. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE - OTM Confederated International Conferences, CoopIS, DOA, and ODBASE. Lecture Notes in Computer Science, vol. 2888, pp. 707–724. Springer (2003)
15. Volz, R., Staab, S., Motik, B.: Incrementally maintaining materializations of ontologies stored in logic databases. Journal on Data Semantics II pp. 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)