Privacy-Preserving Ontology Publishing: The Case of Quantified ABoxes w.r.t. a Static Cycle-Restricted TBox*

Franz Baader[1], Patrick Koopmann[1], Francesco Kriegel[1],
Adrian Nuradiansyah[1], and Rafael Peñaloza[2]

[1] Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany
firstname.lastname@tu-dresden.de
[2] University of Milano-Bicocca, Milano, Italy
rafael.penaloza@unimib.it

Abstract. We review our recent work on how to compute optimal repairs, optimal compliant anonymizations, and optimal safe anonymizations of ABoxes containing possibly anonymized individuals. The results can be used both to remove erroneous consequences from a knowledge base and to hide secret information before publication of the knowledge base, while keeping as much as possible of the original information.

Keywords: Repair · Compliance · Safety · Privacy-preserving ontology publishing · Optimality · Complexity · Practical algorithm

1 Introduction

In contrast to most of the work in description logic (DL), which is about how to derive consequences of a DL knowledge base (KB) efficiently, this paper is about how to get rid of consequences. The reason for this wish can, on the one hand, be that a certain consequence is incorrect, and thus one wants to repair the KB to get rid of this error. On the other hand, one may want to remove a consequence since it is considered to be private information that is not supposed to be made public. In both cases, the new KB should not introduce new consequences (i.e., it should be entailed by the original one), and it should be optimal in the sense that a minimal amount of consequences is removed (i.e., it should be as close as possible to the original one w.r.t. the entailment relation).

Though both scenarios share the wish to remove consequences, there are some differences. On the technical side, in the context of repairs one usually considers a single consequence or a finite set of consequences of the form $C(a)$, i.e., one wants to get rid of instance relationships for specific individuals [3].[3] The resulting KB is then called a repair of the original one. In the context of

* funded by DFG in project number 430150274 and TRR 248 (cpec, grant 389792660).

[3] We restrict the attention here to derived instance relationships, though repairs for subsumption relationships have also been considered in the literature [7].

privacy, one usually considers a policy $\mathcal{P}$, consisting of one or a finite number of concepts, and wants to get rid of all consequences of the form $C(a)$ for $C \in \mathcal{P}$ and a a named individual [6, 8, 11, 12]. The resulting KB is then said to be a $\mathcal{P}$-compliant anonymization of the original one.

On the intentional side, achieving compliance is not always sufficient to guarantee privacy [6, 11, 12]. In fact, an attacker may already have some knowledge, which does not imply the secret, but which together with a published compliant anonymization may be used to derive the secret information. Thus, in the context of privacy, one is interested in computing anonymization that are safe in the sense that, even if extended with an arbitrary compliant KB, they do not imply $C(a)$ for $C \in \mathcal{P}$ and a a named individual.

In the general setting of a DL KB consisting of a TBox and an ABox, optimal repairs (optimal compliant/safe anonymizations) need not exist [3, 7]. There are two ways to overcome this problem. On the one hand, one can weaken the notion of optimality and restrict the attention to repairs (anonymizations) that can be obtained from the original KB by applying certain repair (anonymization) steps. This approach is, e.g., followed in [11, 12] in the setting of privacy and in [7,13,16,22] for the repair scenario. Classical repair approaches that completely remove axioms rather than just weakening them also fall under this category [10, 15, 18–21].

On the other hand, one can stick with the the quest for optimality, and restrict the considered KBs such that optimality can be achieved. Our first work in this direction [5, 9] considered compliance and safety in the very restricted setting of an $\mathcal{EL}$ instance store [14], i.e., where there is no TBox and the ABox does not contain role assertions. In the first paper, the attacker's knowledge is considered to be a set of $\mathcal{EL}$ concept assertions (an $\mathcal{EL}$ instance store) whereas in the second also other DLs are used to represent the attacker's knowledge. In [8] we extended the results of [5] to ABoxes with role assertions (and still no TBox), but restricted the attention to compliance for $\mathcal{EL}$ policies. In [6] we investigated safety in the same setting, but had to restrict the policies to ones consisting of a single $\mathcal{EL}$ concept (singleton policies). Finally, in [3] we extended the results of [8] in two directions, but formulate the new results in the (more general) context of repairs rather than compliance. On the one hand, we add a TBox, which must however be cycle-restricted. On the other hand, we develop a more practical algorithm for computing optimal repairs.

This paper summarizes the results obtained in our previous publications [3,6, 8], but presents them uniformly in the setting of privacy. In addition, it extends the results of [6] by developing a more practical algorithm for computing optimal safe anonymizations. Finally, we show that using TBoxes one can express general policies by singleton policies.

2 Preliminaries

$\mathcal{EL}$ concepts and TBoxes. We assume basic knowledge about DLs [2]. Specifically, we consider the DL $\mathcal{EL}$, defined over a fixed *signature* Σ, which is the disjoint union of the countably infinite sets Σ_O, Σ_C, and Σ_R of *object names*, *concept names*, and *role names*. $\mathcal{EL}$ concepts are built using the concept constructors $\top$, $\sqcap$ and $\exists$. We treat conjunctions as sets, that is, they do not contain duplicates and the order is irrelevant. $\mathcal{EL}$ TBoxes, in the following just called TBoxes, are defined as usual as sets of concept inclusions (CIs) $C \sqsubseteq D$. We use the notation $C \sqsubseteq^\mathcal{T} D$ (alternatively $\mathcal{T} \models C \sqsubseteq D$) to denote that $C \sqsubseteq D$ holds in all models of $\mathcal{T}$. A TBox is called *cycle-restricted* if there is no non-empty sequence of role names $r_1, \ldots, r_k$ and no $\mathcal{EL}$ concept C such that $C \sqsubseteq^\mathcal{T} \exists r_1. \cdots \exists r_k.C$. Cycle-restrictedness of a given TBox can be decided in polynomial time [1].

An *atom* is of the form A or $\exists r.C$, where $A \in \Sigma_\mathsf{C}$ and $r \in \Sigma_\mathsf{R}$. Every $\mathcal{EL}$ concept C is a conjunction of atoms (with $\top$ as empty conjunction), called the *top-level conjunction* of C. We denote the set of atoms occurring in it as $\mathsf{Conj}(C)$. Given a TBox $\mathcal{T}$ and a set $\mathcal{C}$ of concepts, we use $\mathsf{Sub}(\mathcal{T},\mathcal{C})$ to denote the set of concepts occurring in $\mathcal{T}$ and $\mathcal{C}$ (as elements or subconcepts), $\mathsf{Atoms}(\mathcal{T},\mathcal{C})$ to denote the set of atoms occurring in $\mathcal{T}$ and $\mathcal{C}$, and similarly for $\mathsf{Sub}(\mathcal{C})$ and $\mathsf{Atoms}(\mathcal{C})$ for the concepts and atoms occurring in $\mathcal{C}$. Given two sets of $\mathcal{EL}$ concepts $\mathcal{K}$ and $\mathcal{L}$, we say that $\mathcal{K}$ *is covered by* $\mathcal{L}$ (written $\mathcal{K} \leq \mathcal{L}$) if, for every $C \in \mathcal{K}$, there is $D \in \mathcal{L}$ s.t. $C \sqsubseteq^\emptyset D$.

Quantified ABoxes. We use a generalisation of ABoxes called *quantified ABoxes* (*qABoxes*) to adequately represent anonymous individuals as in OWL and *nulls* common in database systems, which play a central role in anonymization [12]. To illustrate, consider the ABox $\{r(a,b), A(a), B(b)\}$, and assume we want to hide the fact that b is an instance of B. Quantified ABoxes allow us to achieve this in a better way than by just deleting the fact $B(b)$, namely by additionally adding an anonymous copy of b, resulting in the quantified ABox $\exists\{x\}.\{r(a,b), A(a), r(a,x), A(x), B(x)\}$, for which a is still an instance of $\exists r.B$.

Essentially, qABoxes are syntactic variants of conjunctive queries. Formally, a qABox is of the form $\exists X.\mathcal{A}$, where X is a finite subset of Σ_O, the elements of which are called *variables*, and $\mathcal{A}$ is the *matrix*, a finite set of concept assertions $A(u)$ where $u \in \Sigma_\mathsf{O}$ and $A \in \Sigma_\mathsf{C}$, and of role assertions $r(u,v)$ where $u, v \in \Sigma_\mathsf{O}$ and $r \in \Sigma_\mathsf{R}$. Without loss of generality, we assume different qABoxes to use disjoint sets of variables. A non-variable object name in $\exists X.\mathcal{A}$ is called an *individual name*, and the set of all these names is denoted as $\Sigma_\mathsf{I}(\exists X.\mathcal{A})$. We further set $\Sigma_\mathsf{O}(\exists X.\mathcal{A}) \coloneqq \Sigma_\mathsf{I}(\exists X.\mathcal{A}) \cup X$. Traditional DL ABoxes are qABoxes where $X = \emptyset$; we then write $\mathcal{A}$ instead of $\exists\emptyset.\mathcal{A}$. The matrix of a qABox is such a traditional ABox. An interpretation $\mathcal{I}$ is a *model* of a qABox $\exists X.\mathcal{A}$ if there is an interpretation $\mathcal{J}$ such that $\Delta^\mathcal{I} = \Delta^\mathcal{J}$, the interpretation functions $\cdot^\mathcal{I}$ and $\cdot^\mathcal{J}$ coincide on $\Sigma \setminus X$, and $u^\mathcal{J} \in A^\mathcal{J}$ for each $A(u) \in \mathcal{A}$ as well as $(u^\mathcal{J}, v^\mathcal{J}) \in r^\mathcal{J}$ for each $r(u,v) \in \mathcal{A}$.

Let $\mathcal{T}$ be a TBox and $\exists X.\mathcal{A}$, $\exists Y.\mathcal{B}$ two qABoxes. We write $\exists X.\mathcal{A} \models^\mathcal{T} \exists Y.\mathcal{B}$ to express that every model of $\mathcal{T}$ and $\exists X.\mathcal{A}$ is also a model of $\exists Y.\mathcal{B}$, in which case we say $\exists Y.\mathcal{B}$ *is entailed by* $\exists X.\mathcal{A}$ *w.r.t.* $\mathcal{T}$. Entailment of traditional ABoxes from a qABox can be decided in polynomial time, while entailment between qABoxes is NP-complete.

3 Computing Optimal Compliant Anonymizations

A *policy* is a finite set of $\mathcal{EL}$ concepts. Intuitively, a policy says that one should not be able to derive that any of the individuals of a qABox belongs to a concept in the policy. To make a given qABox compliant to a policy, we compute an anonymization of it, which is a compliant qABox entailed by it. Intuitively, such an anonymization is optimal if it does not remove more information than necessary.

Definition 1. *Let $\mathcal{T}$ be a TBox, $\mathcal{P}$ be a policy, and $\exists X.\mathcal{A}$, $\exists Y.\mathcal{B}$ be qABoxes.*

1. *$\exists X.\mathcal{A}$ is* compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$ *if, for each $a \in \Sigma_\mathsf{I}(\exists X.\mathcal{A})$ and $C \in \mathcal{P}$, $\exists X.\mathcal{A} \not\models^\mathcal{T} C(a)$,*
2. *$\exists Y.\mathcal{B}$ is a* $\mathcal{P}$-compliant anonymization of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$ *if $\exists X.\mathcal{A} \models^\mathcal{T} \exists Y.\mathcal{B}$ and $\exists Y.\mathcal{B}$ is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$;*
3. *$\exists Y.\mathcal{B}$ is an* optimal $\mathcal{P}$-compliant anonymization of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$ *if additionally $\exists Z.\mathcal{C} \models^\mathcal{T} \exists Y.\mathcal{B}$ implies $\exists Y.\mathcal{B} \models^\mathcal{T} \exists Z.\mathcal{C}$ for every $\mathcal{P}$-compliant anonymization $\exists Z.\mathcal{C}$ of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$.*

Since, in $\mathcal{EL}$, entailment of concept assertions (viewed as singleton ABoxes) is in P, we can decide compliance in polynomial time. More interesting is the question of how to compute a (preferably optimal) anonymization for a given qABox. This problem is investigated in [8] for the case without TBox, and in [3] for the case with TBoxes. These works also consider a weaker version of entailment, called *IQ-entailment*, for the case where we are only interested in instance queries, and [3] considers a generalisation of anonymizations called *ABox repairs*, where instead of a policy, a set of assertions is given that should not be entailed. For brevity, we focus here on the version of anonymizations defined above.

It turns out that, to guarantee existence of optimal anonymizations, we must restrict ourselves to cycle-restricted TBoxes. Consider the traditional ABox $\{A(a)\}$, the TBox $\{A \sqsubseteq \exists r.A,\ \ \exists r.A \sqsubseteq A\}$, and the policy $\{A\}$. An optimal anonymization would have to entail any qABox of the form $\exists\{x_0, \ldots, x_n\}.\{r(a, x_0), r(x_i, x_{i+1}) \mid 0 \leq i \leq n-1\}$ for $n \geq 0$, which is not possible for a qABox entailed by $\{A(a)\}$ w.r.t. $\mathcal{T}$. As shown in [3], this problem can be avoided by considering IQ-entailment, which we do not discuss here.

Next, we present a class of anonymizations called canonical anonymizations, which cover all optimal anonymizations. They are given by a rather elegant direct definition, but may be hard to compute in practice. We then present an optimized approach that computes smaller representations of them.

3.1 Canonical Compliant Anonymizations

If the TBox $\mathcal{T}$ is cycle-restricted, it is possible to compute (in exponential time) its *saturation*, i.e., a qABox $\mathsf{sat}^\mathcal{T}(\exists X.\mathcal{A})$ such that for every qABox $\exists Y.\mathcal{B}$, $\exists X.\mathcal{A} \models^\mathcal{T} \exists Y.\mathcal{B}$ iff $\mathsf{sat}^\mathcal{T}(\exists X.\mathcal{A}) \models^\emptyset \exists Y.\mathcal{B}$. The saturation integrates into the qABox all relevant information that can be inferred using the TBox, so that entailments can be decided without use of the TBox (see [3] for how to compute $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$).

In our approach, we first compute the saturation, and then perform the actual anonymization based on *repair types* and *compliance seed functions*. For convenience, we fix in the following the TBox $\mathcal{T}$, policy $\mathcal{P}$ and qABox $\exists X.\mathcal{A}$ given as input, and abbreviate $\Sigma_{\mathsf{I}}(\exists X.\mathcal{A})$ as Σ_{I}. A repair type specifies for a given object which entailments are to be removed by the anonymization.

Definition 2. *Let $\exists Y.\mathcal{B} \coloneqq \mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ and $u \in \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B})$. A* repair type *for u is a subset $\mathcal{K}$ of $\mathsf{Atoms}(\mathcal{P},\mathcal{T})$ that satisfies the following:*

1. *$\mathcal{B} \models^{\emptyset} C(u)$ for each atom $C \in \mathcal{K}$,*
2. *if C, D are distinct atoms in $\mathcal{K}$, then $C \not\sqsubseteq^{\emptyset} D$,*
3. *$\mathcal{K}$ is* premise-saturated *w.r.t. $\mathcal{T}$, i.e., for all $C \in \mathsf{Sub}(\mathcal{P},\mathcal{T})$ s.t. $\mathcal{B} \models^{\emptyset} C(u)$ and $C \sqsubseteq^{\mathcal{T}} D$ for some $D \in \mathcal{K}$, there is $E \in \mathcal{K}$ such that $C \sqsubseteq^{\emptyset} E$.*

Condition 1 makes sure the concepts in the repair type are indeed entailed for the given individual. Condition 2 avoids redundancies, and Condition 3 ensures that removing the corresponding assertions is effective also in presence of the TBox. The compliance seed function now assigns to every named individual a repair type based on the given policy.

Definition 3. *A* compliance seed function *is a function s that maps each individual name $b \in \Sigma_{\mathsf{I}}$ to a repair type $s(b)$ for b such that, if $C \in \mathcal{P}$ and $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A}) \models^{\emptyset} C(b)$, then there is $D \in s(b)$ such that $C \sqsubseteq^{\emptyset} D$.*

Each compliance seed function induces a compliant anonymization defined next. Intuitively, for concept names $A \in s(a)$, we simply remove the concept assertion $A(a)$ from $\mathcal{A}$. For atoms of the form $\exists r.C \in s(a)$, we need to modify the role successors of a such that $\exists r.C(a)$ is no longer entailed. To avoid losing more information than necessary, we do not just remove assertions from the objects in $\mathcal{A}$, but also create copies of objects by introducing new variables, which are based on the set of repair types for each object name.

Definition 4. *Given a compliance seed function s, we define the* canonical compliant anonymization $\mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$ induced by *s as the qABox $\exists Y.\mathcal{B}$ where:*

1. *The set Y consists of the variables $y_{u,\mathcal{K}}$ s.t. u is an object name in $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ and $\mathcal{K}$ is a repair type for u, except for the case where u is an individual name and $\mathcal{K} = s(u)$. In the latter case, we keep the individual name u, but use $y_{u,s(u)}$ as a synonym for u in the definition of $\mathcal{B}$ below.*
2. *The matrix $\mathcal{B}$ consists of the following assertions:*
   (a) *$A(y_{u,\mathcal{K}})$ if $A(u)$ occurs in $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ and $A \notin \mathcal{K}$, and*
   (b) *$r(y_{u,\mathcal{K}}, y_{v,\mathcal{L}})$ if $r(u,v)$ occurs in $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ and for each $\exists r.C \in \mathcal{K}$ s.t. the matrix of $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ entails $C(v)$, there exists $D \in \mathcal{L}$ s.t. $C \sqsubseteq^{\emptyset} D$.*

Every qABox $\mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$ induced by a seed function s is a compliant optimization of $\exists X.\mathcal{A}$, but it need not be optimal. However, every optimal compliant anonymization is induced (up to equivalence) by some seed function. Thus, we can compute all optimal compliant anonymizations (modulo equivalence) by computing all canonical compliant anonymizations and then removing the non-optimal ones. The latter requires testing entailment between quantified ABoxes.

Theorem 5 ([3]). *There is a deterministic, exponential time algorithm with access to an* NP *oracle that computes the set of all optimal compliant anonymizations of* $\exists X.\mathcal{A}$ *for* $\mathcal{P}$ *w.r.t.* $\mathcal{T}$.

3.2 Optimality Using Minimal Seed Functions

The NP oracle in Theorem 5 is needed for the NP-complete entailment test, which is applied to exponentially large qABoxes. If it is sufficient to compute some, rather than all, optimal compliant anonymizations, we can dispense with the NP oracle and instead utilize a (polynomial time decidable) partial order on seed functions [8]. For two compliance seed functions s and t, we say that *s is covered by t* (written $s \leq t$) if $s(a)$ is covered by $t(a)$ for every $a \in \Sigma_{\mathsf{I}}$, i.e., for every C in $s(a)$ there is D is $t(a)$ s.t. $C \sqsubseteq^{\emptyset} D$.

Proposition 6 ([8]). *If* $\mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, s) \models^{\mathcal{T}} \mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, t)$ *for two compliance seed functions* s *and* t*, then* $s \leq t$.

This was shown in [8] for the case without a TBox, but the proof can easily be extended to the case considered here.

The proposition implies that each minimal seed function induces an optimal anonymization. Since there is always at least one minimal seed function and since $\leq$ can be decided in polynomial time, we can draw the following conclusion.

Theorem 7 ([8]). *A non-empty set of optimal compliant anonymizations of* $\exists X.\mathcal{A}$ *for* $\mathcal{P}$ *w.r.t.* $\mathcal{T}$ *can be computed in exponential time.*

3.3 Smaller Optimal Compliant Anonymizations

Since the number of variables introduced in a canonical compliant anonymization is always exponential in the size of the TBox and the policy,[4] computing even one of them in practice quickly becomes infeasible. The exponential blow-up is in general not avoidable, already for the very limited case without TBox and where the qABox corresponds to an $\mathcal{EL}$ instance store [5]. However, in many practical cases, we can compute a compliant anonymization that is significantly smaller than the canonical compliant anonymization, but logically equivalent to it. The idea is to avoid introducing unnecessary variables by starting with the individual names and unmodified single copies of all object names, and then incrementally

[4] However, canonical anonymizations can be computed in polynomial time w.r.t. data complexity, i.e., if only the size of the qABox counts (TBox and policy fixed).

determining which variables of the canonical anonymization need to be included, where in each step we only look at the immediate role-successors of each object name and the requirements expressed in the associated repair type.

To be more precise, let s be a repair seed function and $\exists Y.\mathcal{B} \coloneqq \mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$. According to Definition 4, we have $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}}) \in \mathcal{B}$ iff $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$ contains the role assertion $r(t,u)$ and the repair type $\mathcal{L}$ covers

$$\mathsf{Succ}(\mathcal{K}, r, u) \coloneqq \{\, C \mid \exists r.C \in \mathcal{K} \text{ and the matrix of } \mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A}) \text{ entails } C(u) \,\}.$$

Our procedure produces a sequence $Y_0, Y_1, \ldots, Y_m$ of subsets Y_i of Y such that $\exists Y.\mathcal{B}$ is equivalent to $\exists Y_m.\mathcal{B}_m$, where $\mathcal{B}_m$ is the subset of $\mathcal{B}$ that uses only objects from $\Sigma_{\mathsf{I}} \cup Y_m$. We start with the set

$$Y_0 \coloneqq \{\, y_{t,\emptyset} \mid t \text{ is an object name occurring in } \mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A}) \,\}.$$

The subsequent sets are obtained by exhaustively applying the following rule:

Compliant Anonymization Rule.
If (i) $y_{t,\mathcal{K}},\ y_{u,\mathcal{L}} \in \Sigma_{\mathsf{I}} \cup Y_i$, (ii) $r(t,u)$ occurs in $\mathsf{sat}^{\mathcal{T}}(\exists X.\mathcal{A})$, (iii) $\mathcal{L}$ does not cover $\mathsf{Succ}(\mathcal{K}, r, u)$, (iv) there is a covering-minimal repair type $\mathcal{M}$ for u that covers $\mathcal{L} \cup \mathsf{Succ}(\mathcal{K}, r, u)$, and (v) $y_{u,\mathcal{M}} \notin \Sigma_{\mathsf{I}} \cup Y_i$,
then set $Y_{i+1} \coloneqq Y_i \cup \{y_{u,\mathcal{M}}\}$.

Since each rule application adds a variable, the exhaustive application of rules must terminate after finitely many steps with a set $Y_m \subseteq Y$ of variables. We call $\exists Y_m.\mathcal{B}_m$ the *optimized compliant anonymization* of $\exists X.\mathcal{A}$ w.r.t. $\mathcal{T}$ induced by the seed function s.

Theorem 8 ([3]). *For each compliance seed function s, the optimized compliant anonymization induced by s is equivalent to $\mathsf{ca}^{\mathcal{T}}(\exists X.\mathcal{A}, s)$.*

To compute $\mathcal{B}_m$ we need not compute the larger matrix $\mathcal{B}$ first. Instead, we directly apply the definition of the matrix (Definition 4) to the object names in $\Sigma_{\mathsf{I}} \cup Y_m$. Experiments with an implementation[5] of this procedure (for the more general case of ABox repairs) indicate that applying this optimized procedure reduces the size of the computed compliant anonymizations considerably.

Example 9. To illustrate both kinds of anonymizations, consider an empty TBox, policy $\mathcal{P} \coloneqq \{P\}$ for $P \coloneqq \exists\mathsf{relative}.(\mathsf{Comedian} \sqcap \exists\mathsf{spouse}.\mathsf{Comedian})$, and qABox $\exists X.\mathcal{A} \coloneqq \exists\{x\}.\{\mathsf{relative}(\mathsf{ben}, x), \mathsf{Comedian}(x), \mathsf{spouse}(x, \mathsf{jerry}), \mathsf{Comedian}(\mathsf{jerry})\}$. As seed function, we select s s.t. $s(\mathsf{ben}) = \{P\}$ and $s(\mathsf{jerry}) = \emptyset$. Fig. 1 depicts both the canonical and the optimized compliant anonymization.

4 Safety of Quantified ABoxes, Mainly Without TBox

To guarantee privacy, policy compliance is not always sufficient since an attacker may have additional knowledge that, by itself, does not reveal the secret,

[5] `https://github.com/de-tu-dresden-inf-lat/abox-repairs-wrt-static-tbox`

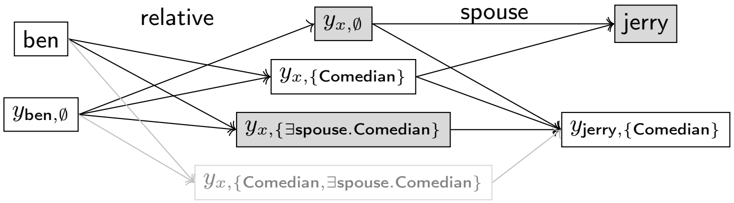
relative
spouse
ben
$y_{x,\emptyset}$
jerry
$y_{x,\{\mathsf{Comedian}\}}$
$y_{\mathsf{ben},\emptyset}$
$y_{x,\{\exists\mathsf{spouse}.\mathsf{Comedian}\}}$
$y_{\mathsf{jerry},\{\mathsf{Comedian}\}}$
$y_{x,\{\mathsf{Comedian},\exists\mathsf{spouse}.\mathsf{Comedian}\}}$

Fig. 1: Canonical anonymization (all nodes) and the subset that is the optimized anonymization (non-shadowed nodes). Gray nodes denote instances of Comedian.

but which, together with the to be published compliant information, would violate the privacy policy. This is captured by the notion of *safety*: a qABox $\exists X.\mathcal{A}$ is *safe* for a given policy $\mathcal{P}$ if for every $\mathcal{P}$-compliant $\exists Y.\mathcal{B}$, the union $\exists(X \cup Y).(\mathcal{A} \cup \mathcal{B})$ is compliant with $\mathcal{P}$ as well. For instance, the canonical compliant anonymization shown in Figure 1 is not safe since we could add the compliant qABox $\exists\{y\}.\{\mathsf{relative}(\mathsf{ben}, y), \mathsf{Comedian}(y), \mathsf{spouse}(y, \mathsf{jerry})\}$. In the resulting qABox, Ben is an instance of the policy concept P.

In [6], we give a characterization for safety of qABoxes for *singleton policies*,[6] which are of the form $\{P\}$ for an $\mathcal{EL}$ concept P. Specifically, safety for $\{P\}$ is violated if (1) $A(a) \in \mathcal{A}$ for some individual name a, and $A \in \mathsf{Atoms}(P)$, or (2) $r(a, u) \in \mathcal{A}$ and $\exists r.D \in \mathsf{Atoms}(P)$ such that a part of the concept D can be found in $\exists X.\mathcal{A}$ at the specific object u—in both cases we can construct attacking compliant qABoxes as certificates for non-safety. The second condition is captured by the notion of *partial homomorphisms* (cf. Definition 3.6 in [6]). Intuitively, a partial homomorphism from a concept D to a qABox is "almost" a homomorphism,[7] but which only maps all those nodes of the syntax tree of D that are between the root and a "cut." Figure 2 shows an example: the "cut" is depicted as the green line. These two conditions are not only necessary but also

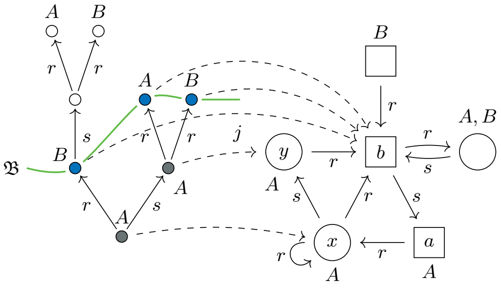

Fig. 2: A partial homomorphism $(j, \mathfrak{B})$

[6] Characterizing safety for general policies is an open problem.
[7] Homomorphisms come into play since they characterize the instance problem in $\mathcal{EL}$.

sufficient for safety.

Proposition 10 ([6]). *$\exists X.\mathcal{A}$ is safe for $\{P\}$ iff, for each individual name a, the following holds: (1) if $A \in \mathsf{Atoms}(\{P\})$, then $A(a) \notin \mathcal{A}$ and (2) if $r(a,u) \in \mathcal{A}$ and $\exists r.D \in \mathsf{Atoms}(\{P\})$, then there is no partial homomorphism from D to $\exists X.\mathcal{A}$ at u.*

Since the existence of a partial homomorphism can be decided in polynomial time [6], we obtain the following complexity result.

Theorem 11. *Safety of qABox w.r.t. singleton $\mathcal{EL}$ policies is in P.*

4.1 Canonical Safe Anonymizations

If a qABox turns out not to be safe, we again want to compute an anonymization that is safe and that preserves as much information from the original qABox as possible. We say that a qABox $\exists Y.\mathcal{B}$ is a *$\{P\}$-safe anonymization* of $\exists X.\mathcal{A}$ if $\exists X.\mathcal{A} \models \exists Y.\mathcal{B}$ and $\exists Y.\mathcal{B}$ is safe for $\{P\}$. Such an anonymization is *optimal* if there is no $\{P\}$-safe anonymization $\exists Z.\mathcal{C}$ of $\exists X.\mathcal{A}$ that lies strictly between $\exists X.\mathcal{A}$ and $\exists Y.\mathcal{B}$ w.r.t. the entailment order. In [6], we presented an approach for computing a unique *optimal safe anonymization* in exponential time. The approach computes a qABox called *canonical safe anonymization* that entails each $\{P\}$-safe anonymization of $\exists X.\mathcal{A}$.

Definition 12. *The* canonical safe anonymization $\mathsf{sa}(\exists X.\mathcal{A}, \{P\})$ *of $\exists X.\mathcal{A}$ w.r.t. $\{P\}$ is defined as the qABox $\exists Y.\mathcal{B}$ such that*

1. *the set Y consists of the variables $y_{t,\mathcal{K}}$ where t is an object name occurring in $\exists X.\mathcal{A}$ and $\mathcal{K}$ is a subset of $\mathsf{Atoms}(\{P\})$ that does not contain $\sqsubseteq_\emptyset$-comparable atoms, and*
2. *the matrix $\mathcal{B}$ consists of the following assertions:*
   (a) $A(y_{t,\mathcal{K}})$ if $A(t)$ occurs in $\mathcal{A}$ and $A \notin \mathcal{K}$,
   (b) $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ provided $r(t,u) \in \mathcal{A}$ and, for each $\exists r.C \in \mathcal{K}$, there is $D \in \mathcal{L}$ with $C \sqsubseteq_\emptyset D$,
   (c) $r(y_{t,\mathcal{K}}, b)$ if $r(t,b)$ occurs in $\mathcal{A}$ and there is no $\exists r.C \in \mathcal{K}$.
   In these conditions, the first object name $y_{t,\mathcal{K}}$ may also stand for an individual name a, which is then treated like the variable $y_{a,\mathsf{Max}(\mathsf{Atoms}(\{P\}))}$, where $\mathsf{Max}(\mathcal{K})$ collects the subsumption-maximal elements of $\mathcal{K}$ modulo equivalence.

As in the case of compliance, the canonical safe anonymizations introduce an exponential number of copies for each object in the input, which may make a computation infeasible in practice.

4.2 Making It Smaller Again

Similar to the case of compliant anonymizations, we can reduce the number of variables in the safe anonymization by creating copies only when needed. According to Definition 12, $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}}) \in \mathcal{B}$ iff $r(t,u) \in \mathcal{A}$ and $\mathcal{L}$ covers

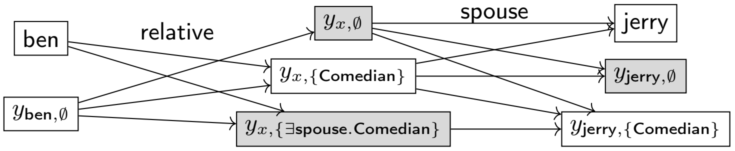
ben
relative
$y_{x,\emptyset}$
spouse
jerry
$y_{x,\{\mathsf{Comedian}\}}$
$y_{\mathsf{jerry},\emptyset}$
$y_{\mathsf{ben},\emptyset}$
$y_{x,\{\exists\mathsf{spouse}.\mathsf{Comedian}\}}$
$y_{\mathsf{jerry},\{\mathsf{Comedian}\}}$

Fig. 3: Optimized safe anonymization of the example ABox. Gray nodes denote instances of Comedian.

$\mathsf{Succ}(\mathcal{K}, r) := \{C \mid \exists r.C \in \mathcal{K}\}$. To compute the optimized safe anonymization, we again produce a sequence $Y_0, \ldots, Y_m$ of subsets of Y. Starting with the set $Y_0 := \{y_{t,\emptyset} \mid t \in \Sigma_\mathsf{O}(\exists X.\mathcal{A})\}$, and applying the following two rules exhaustively.

Safe Anonymization Rule 1.
If (i) $y_{t,\mathcal{K}}, y_{u,\mathcal{L}} \in Y_i$, (ii) $r(t, u) \in \mathcal{A}$, (iii) $\mathcal{L}$ does not cover $\mathsf{Succ}(\mathcal{K}, r)$, (iv) $\mathcal{M}$ is a cover-minimal set of atoms covering $\mathcal{L} \cup \mathsf{Succ}(\mathcal{K}, r)$, but (v) $y_{u,\mathcal{M}} \notin Y_i$,
then set $Y_{i+1} := Y_i \cup \{y_{u,\mathcal{M}}\}$
Safe Anonymization Rule 2.
If (i) $a \in \Sigma_\mathsf{I}$ and $y_{u,\mathcal{L}} \in Y_i$, (ii) $r(a, u) \in \mathcal{A}$, (iii) $\mathcal{L}$ does not cover $\mathsf{Succ}(\mathsf{Max}(\mathsf{Atoms}(\{P\})), r)$, (iv) $\mathcal{M}$ is a cover-minimal set of atoms covering $\mathcal{L} \cup \mathsf{Succ}(\mathsf{Max}(\mathsf{Atoms}(\{P\})), r)$, but (v) $y_{u,\mathcal{M}} \notin Y_i$,
then set $Y_{i+1} := Y_i \cup \{y_{u,\mathcal{M}}\}$

After generating the set of variables, we construct the matrix of the optimized safe anonymization based on Definition 12.

Definition 13. *Let $Y_m \subseteq Y$ be the set of all variables obtained by exhaustive applications of Safe Anonymization Rule 1 and Rule 2. The* optimized $\{P\}$-safe anonymization *of $\exists X.\mathcal{A}$ is the qABox $\exists Y_m.\mathcal{B}_m$, where $\mathcal{B}_m$ contains all assertions in the matrix of* $\mathsf{sa}(\exists X.\mathcal{A}, \{P\})$ *involving only object names in $\Sigma_\mathsf{I} \cup Y_m$.*

Example 14. For the policy and qABox in Example 9, the canonical safe anonymization would contain 24 variables, while the optimized safe anonymization is much smaller.[8] Applying the Safe Anonymization Rule 2 to the pair b and $y_{x,\emptyset}$ exhaustively, we obtain the variables $y_{x,\{C\}}$ and $y_{x,\{\exists s.C\}}$, and then applying the Safe Anonymization Rule 1 to the pair $y_{x,\{C,\exists s.C\}}$ and $y_{j,\emptyset}$ generates $y_{j,\{C\}}$. On the resulting set of objects, no rule is applicable, and our procedure terminates. Thus, the optimized safe anonymization contains only 8 objects in total. Using the matrix construction in Definition 12, we obtain the optimal safe anonymization $\exists Y_m.\mathcal{B}_m$ whose matrix is depicted in Figure 3.

$\mathcal{B}_m$ is a subset of the matrix of $\mathsf{sa}(\exists X.\mathcal{A}, \{P\})$, which implies that the former is entailed by the latter. It immediately follows that $\exists Y_m.\mathcal{B}_m$ is a $\{P\}$-safe anonymization of $\exists X.\mathcal{A}$. We can also show the other direction.

[8] To save space and increase legibility, we abbreviate names by their first letters.

Proposition 15. *The optimized $\{P\}$-safe anonymization of $\exists X.\mathcal{A}$ entails $\mathsf{sa}(\exists X.\mathcal{A}, \{P\})$.*

We thus obtain the following theorem, which shows that we can work with the smaller anonymization.

Theorem 16. *Given a qABox $\exists X.\mathcal{A}$ and a singleton policy $\{P\}$, the optimized $\{P\}$-safe anonymization $\exists Y_m.\mathcal{B}_m$ and $\mathsf{sa}(\exists X.\mathcal{A}, \{P\})$ are equivalent.*

4.3 Static $\mathcal{EL}$ TBoxes and General Policies

So far, our methods for testing for and achieving safety can only deal with singleton policies without a TBox. Safety w.r.t. a TBox is defined as follows: the qABox $\exists X.\mathcal{A}$ is *safe for $\mathcal{P}$ w.r.t. $\mathcal{T}$* if for each quantified ABox $\exists Y.\mathcal{B}$ that is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$, the union $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B}$ is also compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$. Interestingly, TBoxes can be used to express general policies by singleton policies.

Proposition 17. *Consider a quantified ABox $\exists X.\mathcal{A}$, an $\mathcal{EL}$ TBox $\mathcal{T}$, and a policy $\mathcal{P}$. Further let A be a fresh concept name not occurring in $\exists X.\mathcal{A}$, in $\mathcal{T}$, or in $\mathcal{P}$, and define the extended TBox $\mathcal{T}_\mathcal{P} \coloneqq \mathcal{T} \cup \{\, P \sqsubseteq A \mid P \in \mathcal{P}\,\}$. Then $\exists X.\mathcal{A}$ is safe for $\mathcal{P}$ w.r.t. $\mathcal{T}$ iff $\exists X.\mathcal{A}$ is safe for $\{A\}$ w.r.t. $\mathcal{T}_\mathcal{P}$.*

By setting $\mathcal{T} \coloneqq \emptyset$ in this proposition, we see that safety for an arbitrary policy $\mathcal{P}$ (but without TBox) can be reduced to safety for the singleton policy $\{A\}$ w.r.t. to a non-empty cycle-restricted TBox. As shown in [6], such a reduction cannot exist without a TBox. Until now, we do not have a characterization of safety akin to Proposition 10 for non-singleton policies without TBox. The proposition shows that dealing with (cycle-restricted) TBoxes, even for singleton policies, is at least as hard as dealing with general policies.

Nevertheless, by using ideas from [11, 12], we can find a coNP decision procedure for safety for a general policy w.r.t. an $\mathcal{EL}$ TBox. This complexity result extends the one given in [6] (Proposition 3.16) for the case without a TBox, and at the same time corrects a typo in the formulation of that proposition.

Proposition 18. *The safety problem for general policies w.r.t. static $\mathcal{EL}$ TBoxes is in coNP.*

5 Conclusions

The work reviewed in this paper shows that, under some restrictions, optimality can indeed be achieved when computing repairs as well as compliant and safe anonymizations. What remains open is the question of how to deal with general policies and/or cycle-restricted TBoxes in the context of safety. For general TBoxes, optimality is not always achievable, but one can of course ask whether the existence of an optimal repair or an optimal compliant/safe anonymization is decidable, and whether one can then compute such an optimal ABox if it exists. Finally, using conjunctive queries rather then $\mathcal{EL}$ concepts is also an interesting topic for future research.

References

1. Baader, F., Borgwardt, S., Morawska, B.: Extending unification in $\mathcal{EL}$ towards general TBoxes. In: Proc. of the 13th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2012). pp. 568–572. AAAI Press/The MIT Press (2012)
2. Baader, F., Horrocks, I., Lutz, C., Sattler, U.: An Introduction to Description Logic. Cambridge University Press (2017)
3. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Computing optimal repairs of quantified ABoxes w.r.t. static $\mathcal{EL}$ TBoxes. In: Proceedings of the 28th International Conference on Automated Deduction (CADE-28), July 11–16, 2021, Virtual Event, United States (2021), to appear.
4. Baader, F., Koopmann, P., Kriegel, F., Nuradiansyah, A.: Computing optimal repairs of quantified ABoxes w.r.t. static $\mathcal{EL}$ TBoxes (extended version). LTCS-Report 21-01, Chair of Automata Theory, Institute of Theoretical Computer Science, Technische Universität Dresden, Dresden, Germany (2021)
5. Baader, F., Kriegel, F., Nuradiansyah, A.: Privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores. In: Calimeri, F., Leone, N., Manna, M. (eds.) Logics in Artificial Intelligence - 16th European Conference, JELIA 2019, Rende, Italy, May 7-11, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11468, pp. 323–338. Springer (2019)
6. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Safety of quantified ABoxes w.r.t. singleton $\mathcal{EL}$ policies. In: Proceedings of the 36th ACM/SIGAPP Symposium on Applied Computing (SAC 2021) (2021). https://doi.org/https://doi.org/10.1145/3412841.3441961
7. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Making repairs in description logics more gentle. In: Thielscher, M., Toni, F., Wolter, F. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018. pp. 319–328. AAAI Press (2018)
8. Baader, F., Kriegel, F., Nuradiansyah, A., Peñaloza, R.: Computing compliant anonymisations of quantified ABoxes w.r.t. $\mathcal{EL}$ policies. In: Pan, J.Z., Tamma, V.A.M., d'Amato, C., Janowicz, K., Fu, B., Polleres, A., Seneviratne, O., Kagal, L. (eds.) The Semantic Web - ISWC 2020 - 19th International Semantic Web Conference, Athens, Greece, November 2-6, 2020, Proceedings, Part I. Lecture Notes in Computer Science, vol. 12506, pp. 3–20. Springer (2020)
9. Baader, F., Nuradiansyah, A.: Mixing description logics in privacy-preserving ontology publishing. In: Benzmüller, C., Stuckenschmidt, H. (eds.) KI 2019: Advances in Artificial Intelligence - 42nd German Conference on AI, Kassel, Germany, September 23-26, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11793, pp. 87–100. Springer (2019). https://doi.org/10.1007/978-3-030-30179-8_7, `https://doi.org/10.1007/978-3-030-30179-8_7`
10. Baader, F., Suntisrivaraporn, B.: Debugging SNOMED CT using axiom pinpointing in the description logic $\mathcal{EL}^+$. In: Proceedings of the International Conference on Representing and Sharing Knowledge Using SNOMED (KR-MED'08). Phoenix, Arizona (2008)
11. Grau, B.C., Kostylev, E.V.: Logical foundations of privacy-preserving publishing of linked data. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 943–949. AAAI Press (2016)

12. Grau, B.C., Kostylev, E.V.: Logical foundations of linked data anonymisation. J. Artif. Intell. Res. **64**, 253–314 (2019)
13. Horridge, M., Parsia, B., Sattler, U.: Laconic and precise justifications in OWL. In: Sheth, A.P., Staab, S., Dean, M., Paolucci, M., Maynard, D., Finin, T.W., Thirunarayan, K. (eds.) The Semantic Web - ISWC 2008, 7th International Semantic Web Conference, ISWC 2008, Karlsruhe, Germany, October 26-30, 2008. Proceedings. Lecture Notes in Computer Science, vol. 5318, pp. 323–338. Springer (2008)
14. Horrocks, I., Li, L., Turi, D., Bechhofer, S.: The instance store: DL reasoning with large numbers of individuals. In: Haarslev, V., Möller, R. (eds.) Proceedings of the 2004 International Workshop on Description Logics (DL2004), Whistler, British Columbia, Canada, June 6-8, 2004. CEUR Workshop Proceedings, vol. 104. CEUR-WS.org (2004)
15. Kalyanpur, A., Parsia, B., Horridge, M., Sirin, E.: Finding all justifications of OWL DL entailments. In: Proc. of ISWC'07. Lecture Notes in Computer Science, vol. 4825, pp. 267–280. Springer-Verlag (2007)
16. Lam, J.S.C., Sleeman, D.H., Pan, J.Z., Vasconcelos, W.W.: A fine-grained approach to resolving unsatisfiable ontologies. J. Data Semant. **10**, 62–95 (2008)
17. Lutz, C., Wolter, F.: Deciding inseparability and conservative extensions in the description logic $\mathcal{EL}$. J. Symb. Comput. **45**(2), 194–228 (2010)
18. Meyer, T., Lee, K., Booth, R., Pan, J.Z.: Finding maximally satisfiable terminologies for the description logic $\mathcal{ALC}$. In: Proc. of the 21st Nat. Conf. on Artificial Intelligence (AAAI 2006). AAAI Press/The MIT Press (2006)
19. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging OWL ontologies. In: Ellis, A., Hagino, T. (eds.) Proc. of the 14th International Conference on World Wide Web (WWW'05). pp. 633–640. ACM (2005)
20. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies. In: Gottlob, G., Walsh, T. (eds.) Proc. of the 18th Int. Joint Conf. on Artificial Intelligence (IJCAI 2003). pp. 355–362. Morgan Kaufmann, Los Altos, Acapulco, Mexico (2003)
21. Schlobach, S., Huang, Z., Cornet, R., Harmelen, F.: Debugging incoherent terminologies. J. Automated Reasoning **39**(3), 317–349 (2007)
22. Troquard, N., Confalonieri, R., Galliani, P., Peñaloza, R., Porello, D., Kutz, O.: Repairing ontologies via axiom weakening. In: McIlraith, S.A., Weinberger, K.Q. (eds.) Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018. pp. 1981–1988. AAAI Press (2018)

A Proofs

Proof of Proposition 6

Proof. The proof is similar to the one of Proposition 19 in [8], but uses Lemma XII in [4]. □

Proof of Theorem 7

Proof. The statement can be proved analogously to Theorem 20 in [8]. □

Proof of Proposition 17

As in the preconditions of Proposition 17, consider a quantified ABox $\exists X.\mathcal{A}$, let $\mathcal{T}$ be an $\mathcal{EL}$ TBox, and assume that $\mathcal{P}$ is a policy. Further let A be a fresh concept name not occurring in $\exists X.\mathcal{A}$, in $\mathcal{T}$, or in $\mathcal{P}$, and define the extended TBox

$$\mathcal{T}_{\mathcal{P}} \coloneqq \mathcal{T} \cup \{\, P \sqsubseteq A \mid P \in \mathcal{P}\,\}.$$

Before we can prove Proposition 17, we need two auxiliary lemmas.

Lemma 19. *Let C be an $\mathcal{EL}$ concept description where $\emptyset \not\models C \sqsubseteq A$, i.e., in which A does not occur in the top-level conjunction. If $\mathcal{T}_{\mathcal{P}} \models C \sqsubseteq A$, then there is some $P \in \mathcal{P}$ such that $\mathcal{T} \models C \sqsubseteq P$.*

Proof. The proof is by contraposition and thus assume that $\mathcal{T} \not\models C \sqsubseteq P$ for each $P \in \mathcal{P}$. We are going to contruct a model $\mathcal{I}$ of $\mathcal{T}_{\mathcal{P}}$ in which the concept inclusion $C \sqsubseteq A$ is not valid. We initialize the model $\mathcal{I}$ as the canonical model of C w.r.t. $\mathcal{T}$ (according to Definition 11 in [17]), where we denote the root by x_C. Then, we have that $x_C \in C^{\mathcal{I}}$, but $\mathcal{I}$ might not be a model of $\mathcal{T}_{\mathcal{P}}$ (in particular, not of the concept inclusions in $\mathcal{T}_{\mathcal{P}} \setminus \mathcal{T}$). To resolve the latter, we simply add x to $A^{\mathcal{I}}$ for each element $x \in \Delta^{\mathcal{I}}$ where $x \in P^{\mathcal{I}}$. Since $x_C \in C^{\mathcal{I}}$ and $\mathcal{T} \not\models C \sqsubseteq P$ for each $P \in \mathcal{P}$, we infer with Lemma 13 in [17] that $x_C \notin P^{\mathcal{I}}$. Since furthermore A does not occur in the top-level conjunction of C, we conclude that $x_C \notin A^{\mathcal{I}}$. □

Lemma 20. *$\exists X.\mathcal{A}$ is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$ if and only if $\exists X.\mathcal{A}$ is compliant with $\{A\}$ w.r.t. $\mathcal{T}_{\mathcal{P}}$.*

Proof. For showing the if direction, let $\exists X.\mathcal{A} \models_{\mathcal{T}} P(a)$ for some $P \in \mathcal{P}$ and $a \in \Sigma_{\mathsf{I}}$. Since $\mathcal{T}_{\mathcal{P}}$ contains $P \sqsubseteq A$, it follows that $\exists X.\mathcal{A} \models_{\mathcal{T}_{\mathcal{P}}} A(a)$.

Regarding the only-if direction, assume that $\exists X.\mathcal{A} \models_{\mathcal{T}_{\mathcal{P}}} A(a)$ for some $a \in \Sigma_{\mathsf{I}}$. According to Lemma 22 in [17], there is some $\mathcal{EL}$ concept description C such that $\exists X.\mathcal{A} \models C(a)$ and $\mathcal{T}_{\mathcal{P}} \models C \sqsubseteq A$. Note that, since A does not occur in $\exists X.\mathcal{A}$, it cannot occur in C either. The above Lemma 19 yields a policy concept $P \in \mathcal{P}$ such that $\mathcal{T} \models C \sqsubseteq P$, and thus we conclude that $\exists X.\mathcal{A} \models_{\mathcal{T}} P(a)$. □

Proof (of Proposition 17). We start with the if direction and therefore assume that $\exists Y.\mathcal{B}$ is a qABox which is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$ and such that $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models_{\mathcal{T}} P(a)$ for some $P \in \mathcal{P}$ and some $a \in \Sigma_{\mathsf{I}}$. By means of Lemma 20 we infer that $\exists Y.\mathcal{B}$ must be compliant with $\{A\}$ w.r.t. $\mathcal{T}_{\mathcal{P}}$. It is furthermore easy to see that $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models_{\mathcal{T}_{\mathcal{P}}} A(a)$.

For the only-if direction let $\exists Y.\mathcal{B}$ be a qABox that is compliant with $\{A\}$ w.r.t. $\mathcal{T}_{\mathcal{P}}$ and such that $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models_{\mathcal{T}_{\mathcal{P}}} A(a)$ for some individual name a. According to Lemma 20, $\exists Y.\mathcal{B}$ is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$. From $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models_{\mathcal{T}_{\mathcal{P}}} A(a)$ we infer by means of Lemma 22 in [17] that there exists some $\mathcal{EL}$ concept description C where $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models C(a)$ and $\mathcal{T}_{\mathcal{P}} \models C \sqsubseteq A$. Now note that C cannot contain A in its top-level conjunction since A does not occur in $\exists X.\mathcal{A}$ and, due to compliance, $\exists Y.\mathcal{B}$ cannot contain $A(a)$. The above Lemma 19 yields that there is some policy concept $P \in \mathcal{P}$ such that $\mathcal{T} \models C \sqsubseteq P$. We conclude that $\exists X.\mathcal{A} \cup \exists Y.\mathcal{B} \models_{\mathcal{T}} P(a)$. □

Proof of Proposition 15

Proof. The proof is basically similar to the one used in Proposition 13 of [4]. We create a sequence of mappings $h_0, h_1, \ldots, h_n$, where each $h_i : \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B}) \to \Sigma_{\mathsf{O}}(\exists Y_m.\mathcal{B}_m)$ and it will be shown later that the last mapping h_n is a homomorphism from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$. We first define the mapping h_0, where $h_0(a) := a$ for all $a \in \Sigma_{\mathsf{I}}$ and $h_0(y_{t,\mathcal{K}}) := y_{t,\emptyset}$ for all $y_{t,\mathcal{K}} \in Y$.

The following invariants are satisfied by each mapping h_i:

Invariant 1 if $h_i(a) = y$, then $y = a$
Invariant 2 if $h_i(y_{t,\mathcal{K}}) = y_{u,\mathcal{K}'}$, then $t = u$ and $\mathcal{K}' \leq \mathcal{K}$
Invariant 3 if $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$ and $h_{i+1}(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_{i+1}}$, then $\mathcal{K}_i \leq \mathcal{K}_{i+1}$

The mapping h_0 obviously satisfies the first and the second invariant.

Given a mapping h_i, we call a role assertion $r(y, y_{u,\mathcal{L}})$ in $\mathcal{B}$ a *defect* if its image $r(h_i(y), h_i(y_{u,\mathcal{L}}))$ over h_i does not exist in $\mathcal{B}$. In other words, it is a defect

- if $y = a \in \Sigma_{\mathsf{I}}$, $h_i(y_{u,\mathcal{L}}) = y_{u,\mathcal{L}_i}$, $h_i(a) = a$, and $\mathcal{L}_i$ does not cover $\mathsf{Succ}(\mathsf{Max}(\mathsf{Atoms}(\{P\})), r)$ or
- if $y = y_{t,\mathcal{K}}$, $h_i(y_{u,\mathcal{L}}) = y_{u,\mathcal{L}_i}$, $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$, and $\mathcal{L}_i$ does not cover $\mathsf{Succ}(\mathcal{K}_i, r)$.

Intuitively, such role assertions are defects because their image over h_i violates one of the six conditions written in Definition 12, in particular Conditions 2b and 2b. One might ask why there are no defects of the form $r(a, b)$ or $r(y_{t,\mathcal{K}}, b)$. This can be answered as follows.

- If there is an assertion of the first form in $\mathcal{B}$ and assume that it is a defect such that $r(h_i(a), h_i(b)) \notin \mathcal{B}$ because $\exists r.C \in \mathsf{Atoms}(\{P\})$, then this is obviously a contradiction since $h_i(a) = a, h_i(b) = b$, and by Definition 12, the occurrence of $r(a, b) \in \mathcal{B}$ means that $r(a, b) \in \mathcal{A}$ and there is no $\exists r.C \in \mathsf{Atoms}(\{P\})$.

- Furthermore, suppose that there is an assertion of the second form and assume that it is a defect such that $r(h_i(y_{t,\mathcal{K}}), h_i(b)) \notin \mathcal{B}$ because $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$ and $\exists r.C \in \mathcal{K}_i$. However, this assumption cannot be true since $r(y_{t,\mathcal{K}}, b) \in \mathcal{B}$ means that there is no $\exists r.C \in \mathcal{K}$, and due to Invariant 2, we have $\mathcal{K}_i \leq \mathcal{K}$, which implies that $\exists r.C$ cannot also be in $\mathcal{K}_i$.

In the following, we show how the occurrence of a defect triggers a creation of a mapping h_{i+1} that extends h_i. In particular, we assume that h_i is the mapping we have created so far and there is a defect $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ in $\mathcal{B}$ such that $h_i(y_{u,\mathcal{L}}) = y_{u,\mathcal{L}_i}$, $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$, but $\mathcal{L}_i$ does not cover $\mathsf{Succ}(\mathcal{K}_i, r)$.

- First of all, we need to show that $\mathcal{L}$ covers $\mathsf{Succ}(\mathcal{K}_i, r)$. Suppose that there is an existential restriction $\exists r.C \in \mathcal{K}_i$ and we further have $C \in \mathsf{Succ}(\mathcal{K}_i, r)$. Due to Invariant 2, we have $\mathcal{K}_i \leq \mathcal{K}$, and thus there is an atom $\exists r.D \in \mathcal{K}$ such that D subsumes C. Since $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ occurs in the matrix $\mathcal{B}$ of the canonical safe anonymization, by Definition 12, we know that $\mathcal{L}$ covers $\mathsf{Succ}(\mathcal{K}, r)$. Then, it implies that $\mathcal{L}$ contains some atom that subsumes D, and this atom consequently also subsumes C. We conclude that $\mathcal{L}$ covers $\mathsf{Succ}(\mathcal{K}_i, r)$.
- Due to Invariant 2, we can further infer that $\mathcal{L}_i \leq \mathcal{L}$, which implies that $\mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r) \leq \mathcal{L}$.
- Since $\mathcal{L}_i$ does not cover $\mathsf{Succ}(\mathcal{K}_i, r)$, we have $\mathcal{L}_i < \mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r)$.
- By applying **Safe Anonymization Rule 2**, we take a $\leq$-minimal set $\mathcal{L}_{i+1}$ such that $\mathcal{L}_i \cup \mathsf{Succ}(\mathcal{K}_i, r) \leq \mathcal{L}_{i+1} \leq \mathcal{L}$. We define $h_{i+1} := h_i$, except $h_{i+1}(y_{u,\mathcal{L}}) := y_{u,\mathcal{L}_{i+1}}$. Note that $\mathcal{L}_i < \mathcal{L}_{i+1}$, which means that h_i and h_{i+1} are not equal. It is obvious to see that the three invariants are satisfied.
- Furthermore, both **Safe Anonymization Rule 1** and **Safe Anonymization Rule 2** are never applicable to Y_m, and thus we have $y_{u,\mathcal{L}_{i+1}} \in Y_m \subseteq \Sigma_{\mathsf{O}}(\exists Y_m.\mathcal{B}_m)$. This consequently justifies that the mapping h_{i+1} has its range in $\Sigma_{\mathsf{O}}(\exists Y_m.\mathcal{B}_m)$.

What we have shown in the last two paragraphs actually is for the case where a defect is of the form of a role assertion $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ such that its image over h_i does not exist in $\exists Y.\mathcal{B}$. For the other defects that are of the form $r(a, y_{u,\mathcal{L}})$, we can treat them analogously using arguments similar as above, except now we replace $y_{t,\mathcal{K}}$ with individual a and $\mathsf{Succ}(\mathcal{K}, r)$with $\mathsf{Succ}(\mathsf{Max}(\mathsf{Atoms}(\{P\})), r)$.

We show that the construction of this mapping indeed terminates. We define an ordering relation $\leq_h$ on the mappings as follows: $h_i \leq_h h_j$ if for each $y_{t,\mathcal{K}} \in \Sigma_{\mathsf{O}}(\exists Y.\mathcal{B})$, we have $\mathcal{K}_i \leq \mathcal{K}_j$, where $h_i(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_i}$ and $h_j(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_j}$. It is easy to see that this relation $\leq_h$ is a partial order since the covers relation $\leq$ is also a partial order on sets of $\mathcal{EL}$ concepts. We further see from the construction above that for each two consecutive mappings h_i, h_{i+1}, both of them are not equal, which implies that $h_i <_h h_{i+1}$, and thus the sequence of these mappings is strictly increasing until it ends up with a mapping h_n that is free of defects. Furthermore, both $\exists Y.\mathcal{B}$ and $\exists Y_m.\mathcal{B}_m$ are finite, which implies that there are only finitely many types of mappings that map elements of $\exists Y.\mathcal{B}$ to some element in $\exists Y_m.\mathcal{B}_m$. These arguments finally conclude that the construction of this mapping always terminates.

Last but not least, we show that the last mapping h_n is indeed a homomorphism from $\exists Y.\mathcal{B}$ to $\exists Y_m.\mathcal{B}_m$. Due to Invariant 1, we can easily infer that $h_n(a) = a$. For the concept assertions $A(a) \in \mathcal{B}$, we know that $h_n(a) = a, A(a) \in \mathcal{A}$, and there is no $A \in \mathsf{Atoms}(\{P\})$. Using the matrix construction in Definition 12, this implies that $A(a)$ stays in $\mathcal{B}_m$. Now, consider a concept assertion $A(y_{t,\mathcal{K}}) \in \mathcal{B}$ and $h_n(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_n}$. By Definition 12, there is no A in $\mathcal{K}$. Due to Invariant 2, we have $\mathcal{K}_n \leq \mathcal{K}$, and thus $A \notin \mathcal{K}_n$. We finally infer that $A(y_{t,\mathcal{K}_n}) \in \mathcal{B}_m$.

Now, consider role assertions occurring in $\mathcal{B}$. Since role assertions of the form $r(a, b)$ and $r(y_{t,\mathcal{K}}, b)$ can never be a defect, we can restrict our attention only to role assertions of the form $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}})$ and $r(a, y_{u,\mathcal{L}})$.

- Consider role assertions of the form $r(y_{t,\mathcal{K}}, y_{u,\mathcal{L}}) \in \mathcal{B}$. We further assume that $h_n(y_{t,\mathcal{K}}) = y_{t,\mathcal{K}_n}$ and $h_n(y_{u,\mathcal{L}}) = y_{u,\mathcal{L}_n}$. By Definition 12, we know that $r(t, u) \in \mathcal{A}$. Since h_n is already free of defects, we can infer that $\mathcal{L}_n$ covers $\mathsf{Succ}(\mathcal{K}_n, r)$. By Definition 12 and 13, it implies that $r(y_{t,\mathcal{K}_n}, y_{u,\mathcal{L}_n}) \in \mathcal{B}_m$.
- Consider role assertions of the form $r(a, y_{u,\mathcal{L}}) \in \mathcal{B}$. We further assume that $h_n(y_{u,\mathcal{L}}) = y_{u,\mathcal{L}_n}$. By Definition 12, we know that $r(t, u) \in \mathcal{A}$. Since h_n is already free of defects, we can infer that $\mathcal{L}_n$ covers $\mathsf{Succ}(\mathsf{Max}(\mathsf{Atoms}(\{P\})), r)$. By Definition 12 and 13, it implies that $r(a, y_{u,\mathcal{L}_n}) \in \mathcal{B}_m$.

The last three paragraphs above finally conclude that the optimized $\{P\}$-safe anonymization of $\exists X.\mathcal{A}$ for $\{P\}$ entails the canonical safe anonymization of $\exists X.\mathcal{A}$ w.r.t. $\{P\}$. □

Proof of Proposition 18

Recall that the safety problem for general policies w.r.t. static $\mathcal{EL}$ TBoxes asks if a given qABox $\exists X.\mathcal{A}$ is safe for a policy $\mathcal{P} = \{P_1, \ldots, P_n\}$, where $n \geq 1$, w.r.t. some TBox $\mathcal{T}$. To have a decision procedure for this problem, we first need the notion of IQ- saturation, which is a weaker version of chase and is based on the IQ-entailment relation described in Section 3.2 of [3]. Formally, a qABox $\exists X.\mathcal{A}_1$ IQ-entails a qABox $\exists Y.\mathcal{A}_2$ w.r.t. a TBox $\mathcal{T}'$ if every concept assertion $C(a)$ entailed w.r.t. $\mathcal{T}'$ by the latter is also entailed w.r.t. $\mathcal{T}'$ by the former.

In contrast to chase, IQ-saturation always terminates w.r.t. arbitrary $\mathcal{EL}$ TBoxes. In other words, we do not need to impose any restriction to the TBox. The IQ-saturation rules are given in Figure 2 of [3] and the exhaustive application of them terminates in polynomial time in the size of the given qABox and TBox, which then yields a qABox $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$.

In addition to IQ-saturation, we need a characterization of the instance problem w.r.t. $\mathcal{EL}$ TBoxes that is based on a homomorphism function adjusted from [6]. Let $C(a)$ be an $\mathcal{EL}$ concept assertion. As written in the last paragraph of the section of preliminaries in [6], we can express $C(a)$ as a quantified ABox called the *ABox translation* of $C(a)$. As stated in Proposition IV of [4], $\exists X.\mathcal{A} \models^{\mathcal{T}} C(a)$ iff $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}) \models^{\emptyset} C(a)$. This means that the instance problem w.r.t. $\mathcal{EL}$ TBoxes can be reduced to the problem of instance checking w.r.t. quantified ABoxes and no TBoxes whose procedure can be found below Example 3.5 of [6]. In particular, $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A}) \models^{\emptyset} C(a)$ iff there is a homomorphsim from the ABox translation of $C(a)$ to $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A})$. Note that finding such a homomorphism for this instance problem can be done in polynomial time as shown in [8].

Proof (of Proposition 18). If a qABox $\exists X.\mathcal{A}$ is not safe for a general policy $\mathcal{P}$ w.r.t. some TBox $\mathcal{T}$, then there is a qABox $\exists Z.\mathcal{C}$ that is compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$, but the union $\exists X.\mathcal{A} \cup \exists Z.\mathcal{C}$ entails $P(a)$ for some $P \in \mathcal{P}$ and some $a \in \Sigma_{\mathsf{I}}$. This implies that there is a homomorphism from the ABox translation of $P(a)$ to the IQ-saturation $\mathsf{sat}^{\mathcal{T}}_{\mathsf{IQ}}(\exists X.\mathcal{A} \cup \exists Z.\mathcal{C})$ w.r.t. $\mathcal{T}$. This homomorphism basically does not only map each element of the ABox translation of $P(a)$ to objects of $\exists X.\mathcal{A} \cup \exists Z.\mathcal{C}$, but also to objects that are introduced during the exhaustive application of the IQ-saturation rules. Now, let $\exists Y.\mathcal{B}$ be the qABox obtained from $\exists Z.\mathcal{C}$ by removing objects that are not in the image of the homomorphsim. This implies that the number of object names in $\mathcal{B}$ is polynomially bounded by the maximal size of the concepts in $\mathcal{P}$. Since $\exists Y.\mathcal{B}$ is a subset of $\exists Z.\mathcal{C}$, we know that $\exists Y.\mathcal{B}$ is also compliant with $\mathcal{P}$ w.r.t. $\mathcal{T}$. This finally concludes that such a qABox $\exists Y.\mathcal{B}$ can basically be guessed in nondeterministic polynomial time. □