

Logical Foundations of Linked Data Anonymisation

Bernardo Cuenca Grau

Egor V. Kostylev

Department of Computer Science

University of Oxford

Oxford, United Kingdom.

BERNARDO.CUENCA.GRAU@CS.OX.AC.UK

EGOR.KOSTYLEV@CS.OX.AC.UK

Abstract

The widespread adoption of the Linked Data paradigm has been driven by the increasing demand for information exchange between organisations, as well as by regulations in domains such as health care and governance that require certain data to be published. In this setting, sensitive information is at high risk of disclosure since published data can be often seamlessly linked with arbitrary external data sources.

In this paper we lay the logical foundations of anonymisation in the context of Linked Data. We consider anonymisations of RDF graphs (and, more generally, relational datasets with labelled nulls) and define notions of policy-compliant and linkage-safe anonymisations. Policy compliance ensures that an anonymised dataset does not reveal any sensitive information as specified by a policy query. Linkage safety ensures that an anonymised dataset remains compliant even if it is linked to (possibly unknown) external datasets available on the Web, thus providing provable protection guarantees against data linkage attacks. We establish the computational complexity of the underpinning decision problems both under the open-world semantics inherent to RDF and under the assumption that an attacker has complete, closed-world knowledge over some parts of the original data.

1. Introduction

One of the key advantages of the Linked Data paradigm (Bizer, Heath, & Berners-Lee, 2009) is the ability to seamlessly publish and connect uniquely identified data objects on the Web, thus facilitating information sharing and large-scale data analysis. Linked Data is based on the Resource Description Framework (RDF) data model (Manola & Miller, 2004) and the standard RDF query language SPARQL (Harris & Seaborne, 2013).

The widespread adoption of Linked Data has been driven by the increasing demand for information exchange between organisations, as well as by regulations in domains such as health care and governance that require certain data to be made available. Data publishing, however, can lead to the disclosure of sensitive information and hence to the violation of individual privacy—a risk that is exacerbated whenever published data can be linked with external data sources.

Privacy-preserving data publishing (PPDP) refers to the problem of protecting individual privacy while at the same time ensuring that published data remains practically useful (Fung, Wang, Chen, & Yu, 2010). In PPDP there is an emphasis in the publication of actual data. This is in contrast to the less stringent requirements of certain applications where it suffices to publish the results of data analysis (e.g., statistics about groups of individuals, or association rules) instead of the data itself; in such cases, methods such as differential privacy (Dwork, 2008) become extremely useful.

The most prominent form of PPDP is *anonymisation*, where explicit individual identifiers and the values of certain sensitive attributes are obfuscated. Early approaches to database anonymisation involved the removal of just the identifiers of record owners. Sweeney (2002), however, demonstrated the threats posed by information linkage when they disclosed confidential medical records by linking a medical database where patient names and Social Security Numbers had been anonymised with a public voter list containing postcode, gender, and age information. As a result, PPDP has become an increasingly important problem in recent years and several anonymisation techniques have been proposed in the context of relational databases (Fung et al., 2010).

Our goal in this paper is to lay the theoretical foundations for PPDP in the context of Linked Data, with a focus on the semantic requirements that an anonymised RDF graph should satisfy before being released to Web, as well as on the computational complexity of checking whether such requirements are fulfilled. Clearly, these are fundamental steps towards the development of optimised anonymisation algorithms suitable for applications.

In Section 3 we introduce our anonymisation framework, where we assume that an anonymised RDF graph G (or, more generally, a relational dataset with labelled null values) is obtained from the original graph (or relational dataset) G_0 by replacing some occurrences of IRIs (constants) in triples with blank nodes (null values). The sensitive information in G_0 that we aim to protect against disclosure is represented by a conjunctive SPARQL query p , which we refer to as a *policy*. An essential requirement in this setting is that none of the sensitive answers to p hold in the anonymised graph G , in which case we say, in Section 3.2, that G is *policy-compliant*. Although policy compliance ensures that the sensitive information is protected when G is considered in isolation, it provides no guarantee against disclosure once G is released to the Web and can be freely linked with arbitrary external data. To address this limitation we formulate in Section 3.3 an additional *linkage safety* requirement, which ensures that G can be released with provable protection guarantees against linkage attacks. In Section 3.4 we consider the natural situation where an attacker may have complete, *closed-world* information about certain parts of the original graph G_0 and use that information to disclose answers to the policy query; to address such potential vulnerability, we propose general variants of policy compliance and linkage safety that lead to more stringent requirements. In addition to satisfying suitable compliance and safety requirements, we would also like the anonymised graph G to preserve as much information from the original graph G_0 as possible, thus ensuring that the published data remains practically useful. To this end, we introduce in Section 3.5 a notion of cost of an anonymisation and argue that anonymisations with higher cost are those that are semantically “less informative”; as a result, we suggest that one should aim at computing anonymisations with minimal cost. Finally, in Section 3.6 we formulate the decision problems underpinning our notions of policy compliance, linkage safety, and cost minimality in both their open-world and general, closed-world, variants. On the one hand, we introduce a policy compliance (respectively, linkage safety) *checking* problem, where the goal is to verify whether a given anonymisation satisfies the relevant requirements; on the other hand, we also introduce a *cost minimisation problems* for compliance and safety, where the goal is to check whether there exists an anonymisation satisfying the relevant privacy requirements and having at most a given maximum associated cost.

In Section 4 we study the computational complexity of the decision problems associated to policy compliance checking and cost minimisation, and establish tight bounds for all variants of these problems. In particular, we establish completeness for various levels of polynomial hierarchy, as well as NP-completeness in data complexity for all the studied variants with the exception of two tractable cases.

In Section 5 we turn our attention to linkage safety and study the complexity of the associated checking and cost minimisation problems. We show that, under the open-world assumption, all these problems can be solved within the second and third levels of the polynomial hierarchy; furthermore, concerning data complexity, safety checking is a tractable problem (solvable in AC^0), whereas cost minimisation is NP-complete and hence computationally more demanding. In contrast, if we allow the input graph to contain closed-world information, all the aforementioned problems become CONEXPTIME-complete in combined complexity and complete for the second level of the polynomial hierarchy (safety checking) or the third level (cost minimisation) in data complexity.

To the best of our knowledge, ours is the first approach to Linked Data anonymisation providing provable logic-based guarantees against disclosure of declaratively-specified sensitive information. Our results in this paper constitute fundamental initial steps towards the development and implementation of algorithms suitable for applications. Indeed, although most of the decision problems that we study here are inherently intractable, anonymisation in data publishing constitutes in most cases as “offline” process that is only performed once for each data release. Finally, our technical results also establish interesting connections between anonymisation and existing problems in database and graph theory, such as the *critical tuple problem* in relational databases (Miklau & Suciu, 2007) and the *node and edge deletion problems* in graphs (Lewis & Yannakakis, 1980; Yannakakis, 1978).

This paper is an extension of a prior conference publication (Cuenca Grau & Kostylev, 2016). The most significant addition is the definition (in Section 3.6) and comprehensive study (in Sections 4.2 and 5.2) of the decision problems associated to cost minimisation. Another significant addition to our conference paper is the study of *strict suppressors*, which capture a very natural class of anonymisations where all occurrences of the same constant are mapped to the same null. Finally, we have also corrected some erroneous technical claims in the conference paper and strengthened a number of complexity bounds; each of these contributions are indicated at the appropriate places in the relevant sections. We refer the reader to Section 6 for a more detailed general discussion.

2. Preliminaries

We adopt standard notions in function-free first-order logic with equality. We also adopt the *unique name assumption* (UNA), which precludes different constants in formulae from being mapped to the same domain element in an interpretation. Although most of our technical results hold (with the same proof) if the UNA is dropped, there are a few exceptions where our proofs critically depend on the adoption of the UNA; these are discussed later on.

2.1 Datasets with Labelled Nulls

Let `Const` and `Null` be pairwise disjoint, countably infinite, sets of *constants* and (*labelled*) *nulls*, respectively. Assuming a relational vocabulary (i.e., a set of predicates with their

respective arities), a *dataset* is a finite set of atoms with predicates from this vocabulary and arguments from $\mathbf{Const} \cup \mathbf{Null}$. A dataset \mathcal{D} is *ground* if it contains no nulls.

When talking about logical entailment we view a dataset \mathcal{D} as a sentence $\exists \bar{\mathbf{b}}. \bigwedge_{\alpha \in \mathcal{D}} \alpha$, where $\bar{\mathbf{b}}$ are the nulls occurring in \mathcal{D} ; in particular, a ground dataset corresponds to a conjunction of ground atoms. According to this interpretation, renamings of nulls preserve logical equivalence; hence, we consider datasets modulo such renamings and assume that datasets \mathcal{D}_1 and \mathcal{D}_2 have disjoint sets of nulls in use when taking the union $\mathcal{D}_1 \cup \mathcal{D}_2$.

As usual, given datasets \mathcal{D}_1 and \mathcal{D}_2 , a *homomorphism* from \mathcal{D}_1 to \mathcal{D}_2 is a mapping $h : \mathbf{Const} \cup \mathbf{Null} \rightarrow \mathbf{Const} \cup \mathbf{Null}$ such that $h(c) = c$ for each $c \in \mathbf{Const}$ and $h(\mathcal{D}_1) \subseteq \mathcal{D}_2$, where $h(\mathcal{D}_1)$ is the result of applying h to the nulls and constants in all the atoms in \mathcal{D}_1 .

Logical entailment of datasets can be characterised in terms of existence of a homomorphism: $\mathcal{D}_1 \models \mathcal{D}_2$ if and only if there is a homomorphism from \mathcal{D}_2 to \mathcal{D}_1 .

2.2 Queries and Query Answering

A *conjunctive query* (CQ) with *answer variables* \bar{x} and *existential variables* \bar{y} is a formula of the form $\exists \bar{y}. \varphi(\bar{x}, \bar{y})$, where the *body* $\varphi(\bar{x}, \bar{y})$ is a conjunction of atoms with each argument either a constant from \mathbf{Const} or a variable from $\bar{x} \cup \bar{y}$. A CQ is *quantifier-free* if it has no existential variables; it is *Boolean* if it has no answer variables; and it is *atomic* if it consists of a single atom. The size $|q|$ of a CQ q is the number of atoms it contains.

Analogously to datasets, a *homomorphism* from the body $\varphi(\bar{x}, \bar{y})$ of a CQ to a dataset \mathcal{D} is a mapping $g : \mathbf{Const} \cup \bar{x} \cup \bar{y} \rightarrow \mathbf{Const} \cup \mathbf{Null}$ such that $g(c) = c$ for each $c \in \mathbf{Const}$ and $g(\varphi(\bar{x}, \bar{y})) \subseteq \mathcal{D}$. A tuple of constants \bar{c} from \mathbf{Const} is an *answer* to a CQ $q(\bar{x})$ over a dataset \mathcal{D} if $\mathcal{D} \models q(\bar{c})$, where $q(\bar{c})$ is the Boolean CQ obtained from $q(\bar{x})$ by replacing \bar{x} with the corresponding constants in \bar{c} . Equivalently, \bar{c} is an answer to $q(\bar{x})$ if there exists a homomorphism g from the body $\varphi(\bar{x}, \bar{y})$ of $q(\bar{x})$ to \mathcal{D} such that $g(\bar{x}) = \bar{c}$.

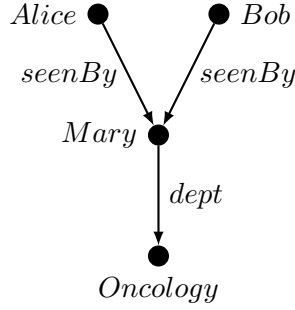
2.3 RDF and SPARQL

All our technical results are stated for general datasets with nulls; however, our work is motivated by RDF and Linked Data, so we next define RDF graphs and describe their correspondence to datasets with nulls.

Let \mathbf{I} , \mathbf{L} , and \mathbf{B} be countably infinite pairwise disjoint sets of *IRIs*, *literals*, and *blank nodes*, respectively. An (*RDF*) *triple* is an element (s, p, o) of the set $(\mathbf{I} \cup \mathbf{B}) \times \mathbf{I} \times (\mathbf{I} \cup \mathbf{L} \cup \mathbf{B})$, where s is referred to as the *subject*, p as the *predicate*, and o as the *object*. An *RDF graph* is a finite set of triples.

RDF comes with a Tarski-style model theory (Hayes, 2004), according to which every RDF graph G can be seen as a dataset \mathcal{D}_G over one ternary predicate **Triple** that consists of atoms $\mathbf{Triple}(s, p, o)$ for each triple (s, p, o) in G , where IRIs \mathbf{I} and literals \mathbf{L} play role of constants, and blank nodes \mathbf{B} play role of nulls that are local to the graph in which they occur (Hogan, Arenas, Mallea, & Polleres, 2014). RDF is equipped with a *merge* operation $G_1 + G_2$ that first renames apart blank nodes in G_1 and G_2 and then constructs the set-theoretic union of their triples; this corresponds precisely to the union of their associated datasets \mathcal{D}_{G_1} and \mathcal{D}_{G_2} (under the assumption in the previous section).

CQs correspond to the core of the W3C standard query language SPARQL as follows. Given variables \mathbf{X} , a basic *SPARQL query* q is of the form **SELECT** \bar{x} **WHERE** P ,


 Figure 1: Example RDF graph G_0

where \bar{x} is a tuple of distinct variables in \mathbf{X} and P is a set of *triple patterns* (s, p, o) with $s, o \in \mathbf{I} \cup \mathbf{L} \cup \mathbf{X}$ and $p \in \mathbf{I} \cup \mathbf{X}$ such that all variables in \bar{x} appear in P . Each such query directly corresponds to a CQ $q(\bar{x}) = \exists \bar{y}. \bigwedge_{(s,p,o) \in P} \text{Triple}(s, p, o)$, with \bar{y} the variables in P that are not in \bar{x} .

2.4 Complexity Classes

We use standard definitions of the basic time complexity classes such as P, NP, coNP, NEXPTIME and CONEXPTIME. We also consider the class AC^0 used in circuit complexity, which encompasses all families of circuits of constant depth and polynomial size with unlimited fan-in AND and OR gates (Papadimitriou, 1994). For complexity classes C and C' , we denote by $C^{C'}$ the class of decision problems that can be solved by a Turing machine running in C and using an oracle for decision problems in C' . The *polynomial hierarchy* is then defined inductively as follows:

$$\Sigma_0^p = \Pi_0^p = \text{P}, \quad \Sigma_{k+1}^p = \text{NP}^{\Sigma_k^p}, \quad \text{and} \quad \Pi_{k+1}^p = \text{coNP}^{\Sigma_k^p}.$$

Finally, we also consider the class DP, which contains each language that is the intersection of a language in NP and a language in coNP. Class DP contains the union of NP and coNP, and is contained in both Σ_2^p and Π_2^p .

3. Logical Framework for PPDP

In this section we present our framework for PPDP and anonymisation in the context of Linked Data. For the sake of generality, our definitions are formulated in terms of datasets with nulls; their application to RDF graphs is immediate by their first-order representation. We follow the same convention in the following sections, where complexity results for the decision problems introduced in this section are presented; all our complexity lower bounds can be adapted to hold for a vocabulary with a single ternary predicate and hence apply to the RDF case. Our motivating examples will be given for RDF graphs.

3.1 Anonymising Linked Data

To illustrate the intuitions behind our approach, let us consider as a simple running example the RDF graph G_0 consisting of the following triples, which represent patient data:

$$\tau_1 = (Alice, seenBy, Mary), \quad \tau_2 = (Bob, seenBy, Mary), \quad \tau_3 = (Mary, dept, Oncology),$$

where all elements of the triples are IRIs (i.e., constants). Graph G_0 is depicted in Figure 1, where, for illustrative purposes, subjects and objects are depicted as black dots and predicates are represented as labelled edges.

We would like to publish an anonymised version of G_0 while ensuring that the names of patients who have seen an oncologist will not be disclosed, in which case we will say that the anonymisation is *policy-compliant* (or simply *compliant*). Additionally, we would like the anonymisation to remain compliant even if linked with external RDF graphs available on the Web and which cannot be assumed to be known in advance; in this case, we will say that the anonymisation is *linkage-safe* (or simply *safe*) and will be able to ensure that it can be published on the Web with provable protection guarantees against linkage attacks. Finally, in both cases, we would want the anonymisation to be *minimal* in the sense that it preserves as much information from G_0 as possible while remaining compliant or safe, thus ensuring that the data stays useful in practice to the extent possible.

We assume that the anonymised graph G is obtained from G_0 by replacing specific occurrences of IRIs in triples with blank nodes (i.e., nulls). For instance, such graph could be obtained by replacing *Alice* in triple τ_1 , *Bob* in triple τ_2 and *Mary* in all three triples with distinct blank nodes b_1 , b_2 and b_3 , respectively, thus obtaining the graph G^1 depicted in Figure 2a (where blank nodes are shown as white dots). Semantically, this implies that the anonymised graph G is a weakening of the original graph G_0 , in the sense that \mathcal{D}_G is homomorphically embeddable into \mathcal{D}_{G_0} and hence $\mathcal{D}_{G_0} \models \mathcal{D}_G$.

Following the mainstream approach in PPDP for databases (e.g., see (Meyerson & Williams, 2004)) we formalise anonymisation in terms of *suppressor functions*, which map occurrences of terms in datasets to null values. In contrast to the standard definition, however, we use labelled nulls rather than unlabelled ones.

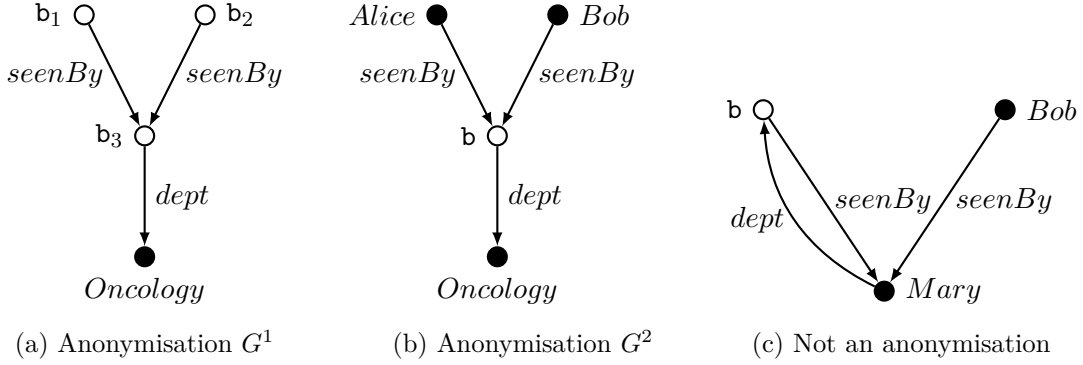
Definition 1. A position s in a dataset \mathcal{D} is a pair $\langle \alpha, j \rangle$ for α an n -ary atom in \mathcal{D} and j a number satisfying $1 \leq j \leq n$. The value $val(s)$ of position s is the j -th argument (constant or null) of α . Given a dataset \mathcal{D}_0 , a \mathcal{D}_0 -suppressor is a function f that maps all positions in \mathcal{D}_0 to $\mathbf{Const} \cup \mathbf{Null}$ such that, for all positions s and s' in \mathcal{D}_0 ,

1. if $f(s) \in \mathbf{Const}$, then $val(s) = f(s)$; and
2. if $f(s) = f(s')$, then $val(s) = val(s')$.

Suppressor f determines the dataset

$$f(\mathcal{D}_0) = \{R(f(\langle \alpha, 1 \rangle), \dots, f(\langle \alpha, n \rangle)) \mid \alpha \text{ an atom in } \mathcal{D}_0 \text{ over } n\text{-ary predicate } R\},$$

which we refer to as an anonymisation of \mathcal{D}_0 . Finally, suppressor f is *strict* if it additionally satisfies the converse of the aforementioned condition 2: for all positions s and s' in \mathcal{D}_0 , if $val(s) = val(s')$ then $f(s) = f(s')$.


 Figure 2: Anonymisations of example graph G_0

Intuitively, a suppressor function f either replaces the value in a position of \mathcal{D}_0 by a null or keeps it as it is, while the same null can only replace the same value. In the context of RDF graphs, a suppressor f can be seen as a function anonymising a subset of individual occurrences of elements in triples by blank nodes. Condition 1 in Definition 1 ensures that a constant occurring in a position of \mathcal{D}_0 cannot be mapped by f to a different constant; in turn, condition 2 ensures that no two positions in \mathcal{D}_0 involving different values are mapped into the same null. These two properties are essential to guarantee that an anonymisation of \mathcal{D}_0 is a logical weakening of (and hence entailed by) \mathcal{D}_0 . Next, suppressor f is strict if it treats uniformly all positions with the same value, by either sending them to the same null, or by leaving them unanonymised. In the context of RDF graphs, strict suppressors can be seen as those anonymising nodes in the graph.

For instance, graphs G^1 and G^2 in Figures 2a and 2b, respectively, are anonymisations of our example graph G_0 , where the corresponding suppressors are strict. In contrast, the graph in Figure 2c is not an anonymisation since its corresponding suppressor function maps the (only) occurrences of IRIs *Alice* and *Oncology* in G_0 to the same blank node, *b*, thus violating condition 2 in Definition 1. Note also that, intuitively, anonymisation G^2 seems to be “better” than G^1 , in the sense that it preserves more information; we will formalise this intuition in Section 3.5.

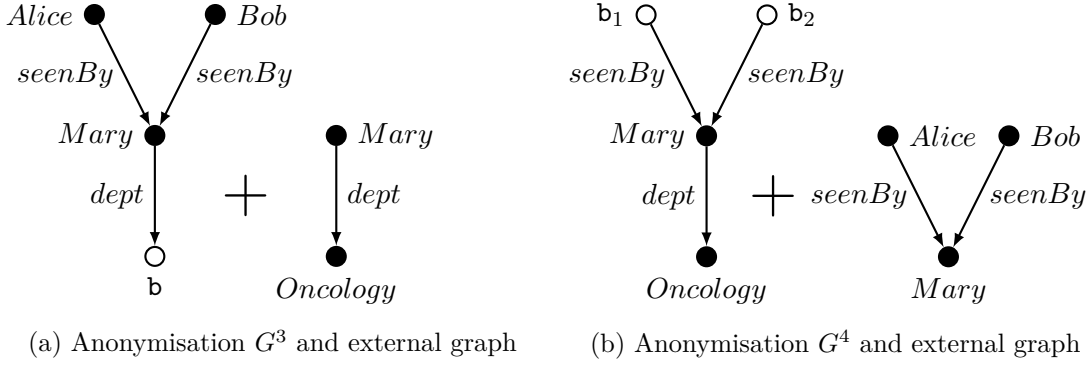
3.2 Formalising Sensitive Information: Policies and Compliance

The sensitive information that we aim to protect against disclosure can be naturally represented as a conjunctive query, which we call a *policy*. For instance, the requirement to protect the list of patients seen by an oncologist can be represented by the following SPARQL query, which has *Alice* and *Bob* as answers over G_0 :

SELECT x WHERE $\{(x, \text{seenBy}, y), (y, \text{dept}, \text{Oncology})\}$.

A suppressor function for a dataset together with a policy constitute a *PPDP instance*, as defined next.

Definition 2. A PPDP instance is a triple (\mathcal{D}_0, f, p) , with \mathcal{D}_0 a dataset, f a \mathcal{D}_0 -suppressor, and p a CQ, called the policy. An instance (\mathcal{D}_0, f, p) is ground if \mathcal{D}_0 is ground. It is an


 Figure 3: Unsafe anonymisations of graph G_0

RDF PPDP instance, if $\mathcal{D}_0 = \mathcal{D}_{G_0}$ for some RDF graph G_0 and the policy p corresponds to a SPARQL query (i.e., the vocabulary of \mathcal{D}_0 and p has a single ternary predicate **Triple**).

To protect the sensitive information in our example graph G_0 , an essential requirement is that the evaluation of the policy over the anonymised graph does not reveal any of the sensitive answers *Alice* and *Bob*. For instance, a strict suppressor that replaces all occurrences of *Mary* in G_0 by a single blank node, thus generating graph G^2 in Figure 2b, would violate the policy since both sensitive answers follow from the resulting anonymisation. In contrast, by replacing *Alice*, *Bob* and *Mary* with blank nodes, as in graph G^1 in Figure 2a, we can ensure that no sensitive answer is disclosed.

Definition 3. A dataset \mathcal{D} complies with a policy p if $\mathcal{D} \not\models p(\bar{c})$ for each tuple \bar{c} of constants. A PPDP instance (\mathcal{D}_0, f, p) is policy-compliant (or, just compliant) if $f(\mathcal{D}_0)$ complies with p .

3.3 Protecting Anonymised Data Against Linkage Attacks

Compliance ensures that sensitive information remains protected when the anonymised data is considered in isolation. It provides, however, no guarantee against disclosure once the anonymised data is released to the Web and can be linked with arbitrary external datasets.

For example, consider a strict suppressor that replaces *Oncology* in our example graph G_0 with a blank node, thus generating graph G^3 on the left-hand side of Figure 3a. Although the anonymised graph G^3 is compliant, the sensitive information can be recovered by linking the anonymised graph with one representing the relationship between doctors and their departments (but saying nothing about patients), as depicted also in Figure 3a. We would equally run into trouble if we followed the natural approach of replacing *Alice* and *Bob* with blank nodes since, as depicted in Figure 3b, the resulting anonymisation G^4 could be linked with a graph capturing the relationship between patients and the doctors they saw (but saying nothing about departments).

Therefore, to provide a sensible level of protection against linking attacks we should ensure that the policy is not compromised even if the anonymisation can be freely linked with other graphs. Obviously, anonymising a graph only makes sense under the assumption that the sensitive information cannot be obtained from external sources only (otherwise, even

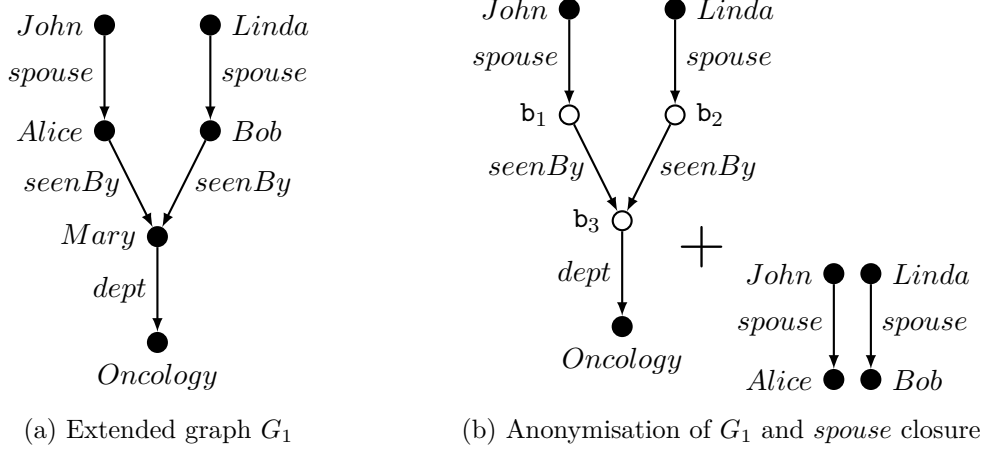


Figure 4: Safety with closed-world information

publishing the empty graph would be problematic); hence, only external graphs complying with the policy are of interest.

Definition 4. A PPDP instance (\mathcal{D}_0, f, p) is linkage-safe (or just safe) if, for every dataset \mathcal{D}' complying with p , the union dataset $f(\mathcal{D}_0) \cup \mathcal{D}'$ also complies with p .

We can ensure safety by replacing all occurrences of *Alice*, *Bob* and *Mary* in G_0 with blank nodes b_1 , b_2 and b_3 , respectively, thus obtaining G^1 in Figure 2a. Intuitively, graph G^1 is safe because its blank nodes cannot be “accessed” by any external graph G' ; indeed, Definition 4 considers the merge of graphs where blank nodes are first renamed apart before constructing the set-theoretic union. As a result, if the merged graph violates the policy, then the external graph G' alone must introduce *Alice* (or *Bob*) and their connection with the oncology department and hence also violate the policy.

3.4 Policy Compliance and Linkage Safety with Closed-World Information

Definition 4 fits well with the first-order logic semantics of RDF, which is inherently open-world. There are situations, however, when a dataset contains relations for which a smart attacker could easily gather *complete* information about. Consider graph G_1 depicted in Figure 4a extending G_0 with the following triples:

$$\tau_4 = (John, spouse, Alice), \quad \tau_5 = (Linda, spouse, Bob).$$

We can satisfy the requirements of Definition 4 by replacing *Alice*, *Bob* and *Mary* with blank nodes, as depicted in Figure 4b. However, an attacker having access to a marriage registry database may have complete information about the *spouse* relation, in which case they can exploit τ_4 and τ_5 to re-identify b_1 with *Alice* and b_2 with *Bob*.

Such re-identification, however, is only possible under the additional assumption that the marriage registry is complete—that is, no unrecorded marriage can exist. This is in contrast to the open-world setting, where linking the anonymised graph with one having

triples τ_4 and τ_5 provides no certain information about the identities of blank nodes \mathbf{b}_1 and \mathbf{b}_2 .¹

We can formally represent such closed-world information by means of a CQ corresponding to the SPARQL query `SELECT x, y WHERE ($x, spouse, y$)` together with its answers $(John, Alice)$ and $(Linda, Bob)$ over the original graph G_1 . To ensure that these answers capture all possible triples over the *spouse* predicate, thus “closing” the *spouse* property, we adapt the standard approach pioneered by Reiter (1992), where a database is seen as a first-order theory with equality axiomatising predicate closure.

Definition 5. A closure $[q, Ans]$ is pair of a CQ $q(\bar{x}) = \exists \bar{y}. \varphi(\bar{x}, \bar{y})$ and a set Ans of tuples of constants with the same size as \bar{x} . In the context of logic, we see $[q, Ans]$ as the set of the following logical sentences:

- $q(\bar{c})$ for each $\bar{c} \in Ans$; and
- $\forall \bar{x}. \forall \bar{y}. (\varphi(\bar{x}, \bar{y}) \rightarrow \bigvee_{\bar{c} \in Ans} \bar{x} = \bar{c})$, where $\bar{x} = \bar{c}$ stands for $\bigwedge_{1 \leq i \leq |\bar{x}|} x_i = c_i$ with x_i and c_i being the i -th components of \bar{x} and \bar{c} , respectively.

In our example, the closure fixes the triples involving the *spouse* predicate and hence, together with the anonymisation, the attacker can derive that \mathbf{b}_1 is *Alice* and \mathbf{b}_2 is *Bob*.

We can incorporate the notion of closure in our framework by generalising policy compliance and linkage safety as given next. We will allow for sets of closures, rather than individual closures as in our prior conference publication (Cuenca Grau & Kostylev, 2016); this is a natural generalisation since the information considered to be complete is likely to involve rather distinct parts of the published graph.

Definition 6. A dataset \mathcal{D} complies with a policy p with respect to a set of closures \mathcal{C} if $\mathcal{D} \cup \mathcal{C} \not\models p(\bar{c})$ for each tuple \bar{c} of constants. A PPDP instance (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} if the anonymisation $f(\mathcal{D}_0)$ complies with p with respect to \mathcal{C} .

Note that, since we adopt the UNA, a dataset can contradict a closure, in which case this dataset does not comply with any policy with respect to the closure.

By this definition, the anonymisation of G_1 in Figure 4b is not compliant with our running example policy with respect to the closure of the *spouse* property.

Definition 7. A PPDP instance (\mathcal{D}_0, f, p) is safe with respect to a set of closures \mathcal{C} if, for each \mathcal{D}' complying with p with respect to \mathcal{C} , the union $f(\mathcal{D}_0) \cup \mathcal{D}'$ also complies with p with respect to \mathcal{C} .

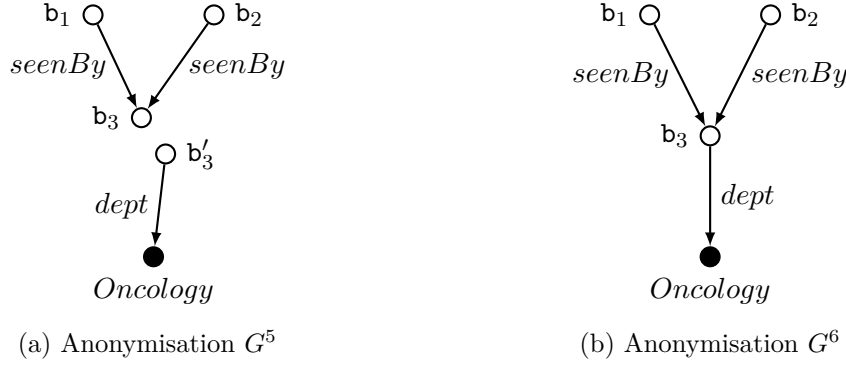
These notions generalise their open-world counterparts: to capture Definitions 3 and 4 it suffices to consider the empty set of closures.

3.5 Maximising Data Availability

There is an intrinsic trade-off between privacy preservation and availability of information. Thus, a key challenge is to ensure that the published linked datasets are protected against disclosure of sensitive information while remaining practically useful.

We next introduce an order on anonymisations, which corresponds to different levels of information availability.

1. Note that the spouse relation is not axiomatised as functional, in which case the identity of the blank nodes would be revealed even in the open-world case.


 Figure 5: Anonymisations of G_0 with different levels of information availability

Consider our example G_0 and the (safe) anonymisation G^5 depicted in Figure 5a. This is, however, not the most informative safe anonymisation derivable from G_0 . For instance, we could identify b_3 with b'_3 to obtain G^6 , depicted in Figure 5b, without putting the policy at risk; this yields additional information since we can now conclude that some patient saw an oncologist. To determine which anonymisations are more informative, we introduce an order between suppressor functions. Intuitively, a suppressor f_1 is at most as informative as a suppressor f_2 if f_2 can be obtained from f_1 by replacing some nulls by constants and, as we did in our example, by identifying some pairs of nulls.

Definition 8. *Given a dataset \mathcal{D}_0 and \mathcal{D}_0 -suppressors f_1 and f_2 , we say that f_1 is at most as informative as f_2 (and that f_2 is at least as informative as f_1), written $f_1 \preceq f_2$, if the following conditions hold for all positions s and s' in \mathcal{D}_0 :*

1. *if $f_1(s) \in \mathbf{Const}$ then $f_2(s) = f_1(s)$, and*
2. *if $f_1(s) = f_1(s')$ then $f_2(s) = f_2(s')$.*

We write $f_1 \prec f_2$ if $f_1 \preceq f_2$, but $f_2 \preceq f_1$ does not hold.

For strict suppressors, condition 2 in the aforementioned definition is redundant.

It is immediate to see that the relation \preceq is a preorder—that is, it is reflexive and transitive. However, suppressors f_1 and f_2 are equivalent according to this preorder—that is, both $f_1 \preceq f_2$ and $f_2 \preceq f_1$ hold—only if f_1 and f_2 are the same up to renaming of nulls. This is not a meaningful distinction in our formalism, and hence in what follows we consider such equivalent f_1 and f_2 the same; then, the preorder \preceq induces a partial order on (the equivalence classes of) suppressors, which we also denote by \preceq .

Order \preceq reflects our understanding of when one suppressor is more informative than another in a simple syntactic way. However, sometimes even if two suppressors are incomparable by \preceq , we may prefer one to another: this happens, for example, when the second suppressor anonymises two constants c_1 and c_2 , while the first suppressor anonymises just one constant that is different from both c_1 and c_2 . To this end, we should define an integer cost of suppressors such that more informative ones have lower cost—that is, $f_1 \prec f_2$ should imply that the cost of f_1 is greater than the cost of f_2 . Looking at Figure 5, a first idea would be to define the cost as the number of different nulls in the range of the suppressor.

Alas, this definition does not satisfy the required property: the suppressor corresponding to G^5 in Figure 5a is strictly less informative than the suppressor corresponding to the dataset obtained from G^6 by replacing one of the occurrences of \mathbf{b}_3 by *Mary*, but both of them have 4 different nulls in the image. In this case, it seems that counting the number of occurrences of nulls in the images is more appropriate. This, however, undesirably makes G^5 and G^6 to have the same cost. Therefore, to deal with both cases, we define the cost as the sum of the numbers of different nulls and null occurrences.

Definition 9. *Given a dataset \mathcal{D}_0 and a \mathcal{D}_0 -suppressor f , the cost of f is the sum of the number of different nulls in the image of f and the number of positions s of \mathcal{D}_0 such that $f(s)$ is a null.*

It is trivial to check that, under this definition, the cost satisfies the intended property.

Proposition 1. *Given a dataset \mathcal{D}_0 and two \mathcal{D}_0 -suppressors f_1 and f_2 with costs ℓ_1 and ℓ_2 , respectively, if $f_1 \prec f_2$ then $\ell_1 > \ell_2$.*

The cost is thus a natural measure which we should aim to minimise when looking for a policy-compliant or linkage-safe anonymisation.

3.6 Reasoning Problems

We are now ready to formulate the decision problems underpinning our notions of policy compliance, linkage safety, and cost minimality.

Definition 10. *Given as input a PPDP instance (\mathcal{D}_0, f, p) and a set of closures \mathcal{C} , we define the following decision problems, corresponding to Definitions 6 and 7:*

- COMPLIANCE: *Is (\mathcal{D}_0, f, p) compliant with respect to \mathcal{C} ?*
- SAFETY: *Is (\mathcal{D}_0, f, p) safe with respect to \mathcal{C} ?*

Given as input a dataset \mathcal{D}_0 , a policy p , a set of closures \mathcal{C} , and a positive integer $\ell \geq 0$, we also define the following decision problems, extending the previous ones with the cost from Definition 9:

- MIN-COMPLIANCE: *Does there exist a \mathcal{D}_0 -suppressor f of cost at most ℓ such that the PPDP instance (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} ?*
- MIN-SAFETY: *Does there exist a \mathcal{D}_0 -suppressor f of cost at most ℓ such that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} ?*

The strict versions MIN-COMPLIANCE_s and MIN-SAFETY_s of problems MIN-COMPLIANCE and MIN-SAFETY, respectively, are obtained by additionally requiring the suppressor f to be strict. The open-world versions X^{ow} of any of the aforementioned problems X are obtained by setting the closure \mathcal{C} as empty.

In addition to the combined complexity of these problems, their data complexity is the complexity when the policy and each of the queries q_i in the input closure $\mathcal{C} = \bigcup_i [q_i, \text{Ans}_i]$ are considered to be fixed.

Note that, in the case of COMPLIANCE and SAFETY, the suppressor is given an input; thus, the problem is to check whether the resulting anonymisation satisfies the compliance or safety requirements, respectively. In practice, however, our goal would often be to *compute* a suppressor of minimal cost satisfying the aforementioned semantic requirements; the problems MIN-COMPLIANCE and MIN-SAFETY constitute the decision counterparts to these natural minimisation problems. Note also that we have not specified whether number ℓ is encoded in unary or in binary; this is immaterial to our complexity bounds in this paper, because the maximal cost of a \mathcal{D}_0 -suppressor is linearly bounded by the number of positions in \mathcal{D}_0 . Finally, note that in our prior conference publication (Cuenca Grau & Kostylev, 2016), the data complexity of reasoning problems involving a closure was defined in a slightly different way: in the conference version the whole closure was considered to be fixed, while now we consider fixed only the query part (but not the answers); we believe that this is a more natural definition since the answers clearly depend on the input data.

Our goal in the following sections will be to establish tight bounds on the computational complexity of the decision problems introduced in Definition 10. We achieve this goal, with only a couple of exceptions where our bounds are not tight. All our results are summarised for the reader’s convenience in Tables 1 and 2, with references to the corresponding propositions and theorems. All our lower bounds hold even for RDF PPDP instances, so all the results are applicable to Linked Data as well. Most of our complexity bounds are the same for arbitrary and Boolean policies; however, in several cases they differ, and the bounds for Boolean policies are shown in parentheses. Many of the lower bounds also hold under additional restrictions, which we discuss explicitly in each particular case.

4. Policy Compliance

In this section, we establish tight complexity bounds for the decision problems associated to policy compliance. These results, summarised in Table 1, will be relevant to the study of linkage safety later on. Policy compliance is also an interesting problem in its own right as, in its most general form, it is strongly connected to the problem of query evaluation under open and closed predicates, which has drawn much recent attention in the knowledge representation community (Lutz, Seylan, & Wolter, 2015; Ahmetaj, Ortiz, & Simkus, 2016).

In Section 4.1 we consider the compliance checking problem COMPLIANCE in both its open-world and general variants. Subsequently, in Section 4.2, we study the minimisation problem MIN-COMPLIANCE across the two dimensions specified in Definition 10: on the one hand, we consider open-world vs. general semantics and, on the other hand, whether suppressors are required to be strict or not.

4.1 Checking Policy Compliance

In this section we show that in the open-world case compliance checking is coNP-complete and in AC^0 in data complexity, while in general the problem is Π_3^P -complete (Π_2^P -complete if the policies are Boolean or have a bounded number of answer variables), and NP-complete in data complexity.

It is immediate to observe that the open-world version of compliance is essentially the complement of the standard Boolean CQ evaluation problem in databases, which is known to be NP-complete and in AC^0 in data complexity (Abiteboul, Hull, & Vianu, 1995). The

	Open-world		General	
	Combined complexity	Data complexity	Combined complexity	Data complexity
COMPLIANCE	coNP-c. [Pr. 2]	in AC^0 [Pr. 2]	Π_3^p -c. (Σ_2^p -c.) [Th. 1]	NP-c. [Th. 1]
MIN-COMPLIANCE	Σ_2^p -c. [Th. 2]	NP-c. [Th. 2]	Σ_4^p -c. (Σ_2^p -c.) [Th. 2]	NP-c. [Th. 2]
MIN-COMPLIANCE _S	Σ_2^p -c. (coNP-c.) [Th. 2]	NP-c. (in AC^0) [Th. 2]	Σ_4^p -c. (Σ_2^p -c.) [Th. 2]	NP-c. [Th. 2]

Table 1: Complexity of policy compliance problems with references to the relevant propositions and theorems (‘c.’ stands for ‘complete’, bounds in parentheses apply for settings with Boolean policies when different from the general case)

	Open-world		General	
	Combined complexity	Data complexity	Combined complexity	Data complexity
SAFETY	DP-hard, in Π_2^p [Th. 3]	in AC^0 [Th. 3]	coNEXPTIME-c. [Th. 3]	Π_2^p -c. [Th. 3]
MIN-SAFETY	Σ_2^p -hard, in Σ_3^p [Th. 4]	NP-c. [Th. 4]	coNEXPTIME-c. [Th. 4]	Σ_3^p -c. [Th. 4]
MIN-SAFETY _S	Σ_2^p -hard, in Σ_3^p [Th. 4]	NP-c. [Th. 4]	coNEXPTIME-c. [Th. 4]	Σ_3^p -c. [Th. 4]

Table 2: Complexity of linkage safety problems with references to corresponding propositions and theorems (‘c.’ stands for ‘complete’)

lower bound in the following proposition slightly strengthens that in our prior work (Cuenca Grau & Kostylev, 2016), where no additional restrictions were imposed.

Proposition 2. *COMPLIANCE^{ow} is coNP-complete and in AC^0 in data complexity. The coNP lower bound holds already for ground RDF PPDP instances, strict suppressors, and Boolean policies.*

Proof. For the coNP lower bound, consider an input dataset \mathcal{D} and Boolean CQ query q to the CQ evaluation problem over the vocabulary with a single ternary predicate *Triple*. Let also $\mathcal{D}_0 = \mathcal{D}$, let $p = q$, and let f be the identity suppressor—that is, the strict \mathcal{D}_0 -suppressor such that $f(\mathcal{D}_0) = \mathcal{D}_0$. By definition, (\mathcal{D}_0, f, p) is compliant if and only if $\mathcal{D} \not\models p$.

For membership in coNP and AC^0 , observe that (\mathcal{D}_0, f, p) is compliant if and only if there is no homomorphism from $p(\bar{x})$ to $f(\mathcal{D}_0)$ mapping \bar{x} to constants. \square

In contrast to the open-world case, we next show that COMPLIANCE in its general form is Π_3^p -complete and Σ_2^p -complete if we restrict ourselves to Boolean policies (or policies with a bounded number of answer variables); moreover, the problem is NP-complete in data

complexity in both cases. In fact, we will show that all three lower bounds hold even for ground RDF PPDP instances, strict suppressors, and atomic closure CQs. In particular, the data complexity lower bound implies that checking compliance of an anonymised RDF graph containing both open and closed predicates is an intractable problem with respect to the size of the input graph and closed relations.

To obtain the upper bounds in the general case, it suffices to observe that a PPDP instance (\mathcal{D}_0, f, p) is compliant with respect to a set of closures \mathcal{C} if and only if for each tuple of constants \bar{c} there exists a model of the logical theory $f(\mathcal{D}_0) \cup \mathcal{C}$ over which $p(\bar{c})$ evaluates to false; furthermore, if such a model exists, then there is one of polynomial size.

Lemma 1. *A PPDP instance (\mathcal{D}_0, f, p) with Boolean p is compliant with respect to a set of closures \mathcal{C} if and only if there exists an interpretation I of size (i.e., the number of tuples in the interpretations of all predicates) at most*

$$|\mathcal{D}_0| + \sum_{[q, Ans] \in \mathcal{C}} |Ans| \times |q|,$$

such that $I \models f(\mathcal{D}_0) \cup \mathcal{C}$ but $I \not\models p$.

Proof. Assume that an interpretation I satisfying the conditions in the lemma exists. Then, $f(\mathcal{D}_0) \cup \mathcal{C} \not\models p$ and, by Definition 6, we have that (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} .

Conversely, if the instance (\mathcal{D}_0, f, p) is compliant, then there must exist a model I' of $f(\mathcal{D}_0) \cup \mathcal{C}$ such that p does not hold in I' . We now observe that $f(\mathcal{D}_0) \cup \mathcal{C}$ is the conjunction of the following sentences:

1. Boolean CQ $\exists \bar{b}. \bigwedge_{\alpha \in f(\mathcal{D}_0)} \alpha$, where \bar{b} are all the nulls in $f(\mathcal{D}_0)$;
2. Boolean CQs $\bigwedge_{\bar{c} \in Ans} \exists \bar{y}. \varphi(\bar{c}, \bar{y})$ for each $[q, Ans] \in \mathcal{C}$ with $q(\bar{x}) = \exists \bar{y}. \varphi(\bar{x}, \bar{y})$; and
3. $\forall \bar{x}. \forall \bar{y}. (\varphi(\bar{x}, \bar{y}) \rightarrow \bigvee_{\bar{c} \in Ans} \bar{x} = \bar{c})$ also for each $[q, Ans] \in \mathcal{C}$ with $q(\bar{x}) = \exists \bar{y}. \varphi(\bar{x}, \bar{y})$.

Let I be any sub-interpretation of I' containing a homomorphic image of each of the CQs in groups 1 and 2 into I . In addition to satisfying all these CQs, I also satisfies sentences in group 3, since so does I' ; furthermore, since it is a sub-interpretation of I' , p does not hold in I as well. Finally, I also satisfies the required size bounds. \square

As a consequence of Lemma 1, we can decide COMPLIANCE by guessing, for each candidate tuple of constants, a polynomial-size interpretation and then calling an NP oracle to check whether the interpretation satisfies all the sentences in groups 1–3 in the proof of the Lemma, while at the same time not satisfying the policy. In the general case, the number of candidate tuples is exponential, so we need to go through them non-deterministically. However, if the policy has bounded number of answer variables (e.g., it is Boolean), then we can check them one by one in polynomial time. The given Π_3^P bound (which we will prove to be tight in Lemma 3) corrects the Σ_2^P bound in our prior work (Cuenca Grau & Kostylev, 2016), which is applicable only if the number of answer variables in the policy is considered fixed.

Lemma 2. *COMPLIANCE is in Π_3^P and in Σ_2^P if the policy has a bounded number of answer variables. The problem is in NP in data complexity.*

Proof. Consider the algorithm that guesses a tuple of constants mentioned in the dataset of size equal to the number of answer variables of the policy, then guesses, for this tuple, an interpretation satisfying the size bounds in Lemma 1, and then checks whether this interpretation satisfies the sentences in all three groups in the lemma as well as the negation of the policy with answer variables substituted by the constants. This algorithm correctly decides COMPLIANCE in Π_3^p , because checking the sentences in groups 1 and 2 in can be done in NP, while checking sentences in group 3 and checking the negation of the policy can be done in CONP. If the number of answer variables in the policy is bounded, then the first guess is not necessary, because we can go through all tuples of constants deterministically one by one; therefore, the algorithm has Σ_2^p complexity. Finally, if the CQs in the policy and the closures are fixed, then a homomorphism witnessing the sentence in group 1 can be also guessed together with the interpretation and then checked in polynomial time along with all the other sentences including the negation p , so the overall algorithm gives us an NP upper bound for data complexity. \square

We start our study of complexity lower bounds for COMPLIANCE in its most general form by providing matching lower bounds to the Π_3^p and Σ_2^p upper bounds in Lemma 2. The lower bounds hold already in the context of RDF, and even under the assumption that the closure CQs are fixed, quantifier-free, and atomic.

Lemma 3. *COMPLIANCE is Π_3^p -hard for ground RDF PPDP instances, strict suppressors, and fixed, quantifier-free and atomic CQs in closures. It is Σ_2^p -hard if additionally policies are restricted to be Boolean.*

Proof. We first justify the Π_3^p lower bound by means of a reduction of $\forall\exists\forall 3\text{SAT}$ —a canonical Π_3^p -complete problem. Consider a quantified Boolean formula $\phi = \forall \bar{s}. \exists \bar{u}. \forall \bar{v}. \neg \psi(\bar{s}, \bar{u}, \bar{v})$ where $\psi(\bar{s}, \bar{u}, \bar{v})$ is a propositional formula in 3CNF over tuples \bar{s} , \bar{u} and \bar{v} of distinct variables. Here, ψ is a conjunction of clauses, and each clause is a disjunction of three literals—that is, variables in $\bar{s} \cup \bar{u} \cup \bar{v}$ or their negations. Then, each clause has at most 7 satisfying Boolean assignments of its three variables, and the overall formula ϕ is valid if and only if for all assignments of the universally quantified variables \bar{s} there exists an assignment of the existentially quantified variables \bar{u} such that for all assignments of variables \bar{v} there exists a clause such that the projection of the resulting assignment to the three variables of this clause is not among the 7 satisfying assignments of the clause.

We construct a PPDP instance (\mathcal{D}_0, f, p) and a set of closures \mathcal{C} , both restricted as required, such that the instance is compliant with respect to the closures if and only if ϕ is valid. The construction uses a vocabulary containing a unary predicate U , a unary predicate Cl_γ for each clause γ in ψ , and binary predicates Arg_1 , Arg_2 , Arg_3 and $U\text{Values}$. Later on, we describe how our construction can be easily adapted to the RDF case, where the vocabulary consists of a single Triple predicate. We will also see that if we set \bar{s} to be empty then the reduction naturally works as a proof of Σ_2^p -hardness of the problem with Boolean policies.

We first define dataset \mathcal{D}_0 and \mathcal{D}_0 -suppressor f . To this end, consider first the anonymisation $f(\mathcal{D}_0)$ consisting of the following facts, where we use constants c_γ^π for every clause γ in ψ and each (of at most 7) assignment π of the variables of γ that satisfies γ , constants c_w^{false} and c_w^{true} for each $w \in \bar{s} \cup \bar{v}$, and nulls b_u^{false} and b_u^{true} for each $u \in \bar{u}$:

- $\text{Cl}_\gamma(c_\gamma^\pi), \text{Arg}_1(c_\gamma^\pi, t_{w_1}^{\pi(w_1)}), \dots, \text{Arg}_3(c_\gamma^\pi, t_{w_3}^{\pi(w_3)})$ for each clause γ over variables w_1, \dots, w_3 and each assignment π of w_1, \dots, w_3 satisfying γ , where $t_w^{\pi(w)}$ is $\mathbf{b}_w^{\pi(w)}$ if $w \in \bar{u}$ and $c_w^{\pi(w)}$ if $w \in \bar{s} \cup \bar{v}$;
- $\text{UValues}(\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}})$ for every $u \in \bar{u}$.

We assume an arbitrary, but fixed enumeration of the (occurrences of the) literals in each clause, which imposes an enumeration of the (occurrences of the) variables in the clause.

Let dataset \mathcal{D}_0 be the same as anonymisation $f(\mathcal{D}_0)$ except that, for each $u \in \bar{u}$, nulls $\mathbf{b}_u^{\text{false}}$ and $\mathbf{b}_u^{\text{true}}$ are replaced with constants c_u and c'_u , respectively. Then, f is the strict \mathcal{D}_0 -suppressor that sends all positions with these constants to the corresponding nulls.

The policy p is the CQ with the following atoms, where the x_s , for $s \in \bar{s}$, are answer variables and all other arguments are existential variables:

- $\text{Cl}_\gamma(x_\gamma), \text{Arg}_1(x_\gamma, x_{w_1}), \dots, \text{Arg}_3(x_\gamma, x_{w_3})$ for each γ in ψ over variables w_1, \dots, w_3 ;
- $\text{U}(x_u)$ for each $u \in \bar{u}$.

Finally, let \mathcal{C} consist of the following two closures with atomic quantifier-free CQs:

- $[\text{UValues}(x, y), \{(c_u, c'_u), (c'_u, c_u) \mid u \in \bar{u}\}],$ and
- $[\text{U}(x), \{c_u \mid u \in \bar{u}\}].$

We now show that formula ϕ is valid if and only if the anonymisation $f(\mathcal{D}_0)$ complies with p with respect to \mathcal{C} . Intuitively, compliance holds when for each \bar{c} consisting of c_s^{false} or c_s^{true} for every $s \in \bar{s}$ there is a model of $f(\mathcal{D}_0) \cup \mathcal{C}$ in which $p(\bar{c})$ does not hold. In search for such a model we need to go through all possible identifications of each pair of nulls $(\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}})$ with $u \in \bar{u}$ to either $(c_{u'}, c'_{u'})$ or $(c'_{u'}, c_{u'})$ for some u' (same as or different from u), and this corresponds to the assignment of u to **true** or **false**. A homomorphism from $p(\bar{c})$ to such a model corresponds to a completion of this assignment to \bar{v} that turns ψ to **true**, because each x_u must be sent to the corresponding $c_{u'}$, which is in U contrary to $c'_{u'}$. Next we formally prove this intuition correct.

Assume that ϕ holds—that is, for every truth assignment σ of variables \bar{s} there exists an extension of σ to variables \bar{u} such that for any further extension of σ to variables \bar{v} the resulting overall assignment turns ψ to **false**. We need to prove that for every tuple of constants \bar{c} there is a model I of $f(\mathcal{D}_0) \cup \mathcal{C}$ that is not a model of $p(\bar{c})$. Consider an arbitrary tuple \bar{c} of the same size as \bar{s} . If there exists $s \in \bar{s}$ such the constant c_s in \bar{c} corresponding to x_s is neither c_s^{false} nor c_s^{true} , then the claim is trivial. Indeed, we can take as I the Herbrand interpretation of \mathcal{D}_0 extended by the atoms enforced by \mathcal{C} —that is, atoms $\text{UValues}(c_u, c'_u)$, $\text{UValues}(c'_u, c_u)$, and $\text{U}(c_u)$ for each $u \in \bar{u}$ (the *Herbrand interpretation* of a ground dataset is the interpretation that interprets all constants by themselves and all predicates exactly according to the atoms in this dataset). Then, $I \not\models p(\bar{c})$ because c_s is not connected by Arg_i in I to an element in Cl_γ for a clause γ with s in the i -th literal. Consider now the case when \bar{c} has c_s^{false} or c_s^{true} in the position corresponding to each x_s with $s \in \bar{s}$. Let σ be the assignment of \bar{s} corresponding to \bar{c} —that is, such that $\sigma(s) = \text{false}$ if and only if \bar{c} has c_s^{false} in the corresponding position. By our assumption, there exists an extension of σ

to \bar{u} such that for any further extension of σ to variables \bar{v} the resulting overall assignment turns ψ to **false**. Consider such an extension to \bar{u} and the Herbrand interpretation I of the ground dataset obtained from the anonymisation $f(\mathcal{D}_0)$ by

- adding all the atoms enforced by \mathcal{C} as above, and
- replacing, for each $u \in \bar{u}$, the pair $\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}}$ by the pair c_u, c'_u if $\sigma(u)$ is **false** and by the pair c'_u, c_u otherwise.

Interpretation I is a model of $f(\mathcal{D}_0) \cup \mathcal{C}$ by construction. We claim that it is not a model of $p(\bar{c})$. Indeed, if $I \models p(\bar{c})$, then there exists a homomorphism from $p(\bar{c})$ to I , and the only possibility is that each x_γ is sent to one of c_γ^π with π agreeing with σ on the used variables, each x_u to c_u , and each x_v to either c_v^{false} and c_v^{true} , and the combination of all these π extends σ to \bar{v} in such a way that ψ evaluates to **true**, which is not possible by assumption. Therefore, I is a model of $f(\mathcal{D}_0) \cup \mathcal{C}$ and not a model of $p(\bar{c})$. Since the choice of \bar{c} was arbitrary, we conclude that anonymisation $f(\mathcal{D}_0)$ complies with p with respect to \mathcal{C} .

Assume now that anonymisation $f(\mathcal{D}_0)$ complies with p with respect to \mathcal{C} —that is, for each \bar{c} there exists a model of $f(\mathcal{D}_0) \cup \mathcal{C}$ that does not satisfy $p(\bar{c})$. Consider an arbitrary assignment σ of \bar{s} and the corresponding tuple of constants \bar{c} —that is, the tuple that has c_s^{false} in the position corresponding to each x_s with $\sigma(s) = \text{false}$ and c_s^{true} in the position corresponding to each x_s with $\sigma(s) = \text{true}$. Let I be a model of $f(\mathcal{D}_0) \cup \mathcal{C}$ that does not satisfy $p(\bar{c})$, which exists by the assumption. Since I is a model of $f(\mathcal{D}_0)$, there is a homomorphism h from $f(\mathcal{D}_0)$ to I , and, since I is a model of \mathcal{C} , the pair $h(\mathbf{b}_u^{\text{false}}), h(\mathbf{b}_u^{\text{true}})$, for every $u \in \bar{u}$, is either the interpretations of $c_{u'}, c'_{u'}$ or of $c'_{u'}, c_{u'}$, for some $u' \in \bar{u}$. Consider an extension of assignment σ to \bar{u} that sends each u to **false**, if $h(\mathbf{b}_u^{\text{false}}), h(\mathbf{b}_u^{\text{true}})$ are the interpretations of $c_{u'}, c'_{u'}$, and to **true** otherwise. We claim that there is no further extension of σ to \bar{v} turning ψ to **true**—that is, we claim that ϕ is valid. Indeed, if it is not the case, then we can construct a homomorphism from $p(\bar{c})$ to I contradicting the fact that I is not a model of $p(\bar{c})$: such a homomorphism would send

- for each clause γ , variable x_γ to the interpretation of c_γ^π , where π is the restriction of σ to the variables of γ ,
- for each $u \in \bar{u}$, variable x_u to $h(\mathbf{b}_u^{\sigma(u)})$ (which is either the interpretation of $c_{u'}$ or of $c'_{u'}$, as described above), and
- for each $v \in \bar{v}$, variable x_v to the interpretation of $c_v^{\sigma(v)}$.

Consequently, ϕ is valid, as required.

To conclude the Π_3^p hardness proof, we adapt our reduction to RDF, where the vocabulary contains a single ternary predicate **Triple**. Since only unary and binary predicates are involved in the construction, the adaptation is trivial: unary atoms $\mathbf{A}(t)$ translate to atoms $\text{Triple}(t, \text{type}, \mathbf{A})$, where *type* is a fresh constant global for the translation, and binary atoms $\mathbf{P}(t_1, t_2)$ translate to $\text{Triple}(t_1, \mathbf{P}, t_2)$. Here predicates \mathbf{A} and \mathbf{P} are seen as constants.

Finally, note that if \bar{s} is empty, then we reduce a canonical Σ_2^p -complete problem $\exists\forall 3\text{SAT}$. Moreover, in this case the constructed policy p becomes Boolean, and hence we obtain a proof for the second claim of the theorem. \square

Next, we establish NP-hardness of COMPLIANCE in data complexity by providing a reduction of 3-COLOURABILITY, a well-known NP-complete graph problem. Our NP lower bound strengthens that in our prior work, where we do not impose any restrictions on instances, suppressors, policies, and closures.

Lemma 4. *COMPLIANCE is NP-hard in data complexity for ground RDF PPDP instances, strict suppressors, Boolean policies, and quantifier-free and atomic CQs in closures.*

Proof. We provide a reduction of 3-COLOURABILITY. As in the proof of the Lemma 3, our reduction uses only unary and binary predicates, and can thus be adapted to the RDF case in the same way as we did in that proof.

Let \mathcal{G} be an input undirected graph to 3-COLOURABILITY with nodes \mathcal{V} and edges \mathcal{E} . Without loss of generality, we assume that \mathcal{G} is connected. We construct a PPDP instance (\mathcal{D}_0, f, p) and a set of closures \mathcal{C} satisfying the requirements such that neither p nor the queries in \mathcal{C} depend on \mathcal{G} , and (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} if and only if \mathcal{G} is 3-colourable. In the construction, we use two unary predicates \mathbf{U} and \mathbf{V} , and a binary predicate \mathbf{Edge} .

Dataset \mathcal{D}_0 uses constants c_v for all $v \in \mathcal{V}$ and consists of

- atoms $\mathbf{Edge}(c_{v_1}, c_{v_2})$ and $\mathbf{Edge}(c_{v_2}, c_{v_1})$ for each edge $\{v_1, v_2\}$ in \mathcal{E} , and
- atom $\mathbf{V}(c_v)$ for each $v \in \mathcal{V}$.

Suppressor f is a strict \mathcal{D}_0 -suppressor that, for each $v \in \mathcal{V}$, sends each position involving constant c_v to a null \mathbf{b}_v that is uniquely associated with v . Policy p is defined as the Boolean CQ $\exists x. \mathbf{U}(x) \wedge \mathbf{V}(x)$; clearly, p does not depend on \mathcal{G} , as required. Finally, the set \mathcal{C} consists of the following closures with existential-free atomic CQs that do not depend on \mathcal{G} :

- $[\mathbf{Edge}(x, y), \mathbf{Ans}]$, where \mathbf{Ans} consists of
 - the pairs (c_{v_1}, c_{v_2}) and (c_{v_2}, c_{v_1}) for each edge $\{v_1, v_2\}$ in \mathcal{E} , and
 - the pairs $(d^r, d^g), (d^g, d^r), (d^g, d^b), (d^b, d^g), (d^b, d^r),$ and (d^r, d^b) , where $d^r, d^g,$ and d^b are constants representing the three colours; and
- $[\mathbf{U}(x), \{c_v \mid v \in \mathcal{V}\}]$.

We next argue that \mathcal{G} is 3-colourable if and only if (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} .

Assume that \mathcal{G} is 3-colourable. Consider an interpretation I that interprets \mathbf{Edge} by all pairs in \mathbf{Ans} , \mathbf{U} by all c_v , and \mathbf{V} by d^r, d^g and d^b . Interpretation I is a model of $f(\mathcal{D}_0)$, because the function h sending each \mathbf{b}_v to one of d^r, d^g and d^b according to a colouring of \mathcal{G} is a homomorphism from $f(\mathcal{D}_0)$ to I . It is also a model of \mathcal{C} by construction. Finally, it is not a model of p . Therefore, (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} .

Assume now that (\mathcal{D}_0, f, p) is compliant with respect to \mathcal{C} —that is, there is a model I of $f(\mathcal{D}_0) \cup \mathcal{C}$ that is not a model of p . Since I is a model of the anonymisation $f(\mathcal{D}_0)$, there is a homomorphism h from $f(\mathcal{D}_0)$ to I . Since \mathcal{G} is connected, and I is a model of \mathcal{C} and not a model of p , the image of each \mathbf{b}_v under h is the interpretation of one of d^r, d^g and d^b . Therefore, we can construct a 3-colouring of \mathcal{G} by colouring each v to red if $h(\mathbf{b}_v)$ is the interpretation of d^r , to green if it is the interpretation of d^g , and to blue if it is the interpretation of d^b . \square

The following theorem summarises the results of Lemmas 2, 3, and 4.

Theorem 1. *COMPLIANCE is Π_3^p -complete and Σ_2^p -complete for Boolean policies; it is NP-complete in data complexity. The lower bounds hold already for ground RDF PPDP instances, strict suppressors, and fixed, quantifier-free and atomic CQs in closures. The data complexity lower bound holds additionally for Boolean policies.*

To conclude this section, we recall that the conference version of this paper (Cuenca Grau & Kostylev, 2016) considered only policies with bounded number of answer variables. That version also made slightly different assumptions when establishing the Σ_2^p and NP lower bounds for COMPLIANCE. On the one hand, it was assumed that the closure set \mathcal{C} was a singleton but involved a query with existential variables, whereas we assumed \mathcal{C} to consist of two quantifier-free and atomic closures. On the other hand, the lower bounds in our prior work were established for non-strict suppressors f (Cuenca Grau & Kostylev, 2016), whereas we now additionally require the suppressor to be strict. Our new bounds thus tighten those in our conference publication by assuming strict suppressors and quantifier-free atomic closure queries, but at the slight cost of requiring two closures in \mathcal{C} instead of just one.

4.2 Cost Minimisation for Policy Compliance

In this section we establish tight complexity bounds for the two cost minimisation problems MIN-COMPLIANCE and MIN-COMPLIANCE_s associated to policy compliance. We show that in the open-world case both problems are Σ_2^p -complete in general; however, if we restrict ourselves to Boolean policies, then MIN-COMPLIANCE has the same complexity, while MIN-COMPLIANCE_s is CONP-complete. In the case with closed-world information the problems become Σ_4^p -complete; for Boolean policies they are Σ_2^p -complete. In data complexity, all versions of both problems are NP-complete, except MIN-COMPLIANCE_s^{ow} for Boolean policies, which is in AC⁰. Therefore, in most of the cases the cost minimisation problems for compliance are one level higher in the polynomial hierarchy than the usual compliance checking problems.

The upper bounds are easily obtained by means of simple guess-and-check algorithms, which are straightforward variants of those described in Section 4.1 for compliance checking.

Lemma 5. *MIN-COMPLIANCE and MIN-COMPLIANCE_s are in Σ_4^p and in Σ_2^p if the policy is Boolean. The problems are in NP in data complexity. Additionally, if the policy is Boolean then MIN-COMPLIANCE_s^{ow} is in CONP and in AC⁰ in data complexity.*

Proof. For the first three bounds, we can use the same algorithms as in the proof of Lemma 2, with the exception that in the first step we additionally need to guess a suppressor (arbitrary in case of MIN-COMPLIANCE and strict in case of MIN-COMPLIANCE_s) within the required cost. This suffices because, as we already discussed in Section 3.6, the sizes of all the suppressors are linearly bounded by the size of their corresponding datasets.

For the last two bounds, note that if the policy is Boolean and constant-free, then no strict suppressor can change the answer to the policy—that is, the answers to the policy on the original dataset and on the resulting anonymisation are the same. So, in this case MIN-COMPLIANCE_s^{ow} can be done by checking whether the policy has the positive answer

on the original dataset and complementing the result, which is possible to do in CONP and in AC^0 in data complexity by standard database techniques (Abiteboul et al., 1995). If the Boolean policy mentions constants, then we first need to check that there are no constants in the policy such that the strict suppressor anonymising this constant alone has the cost within the limit; if such a constant exists, then $\text{MIN-COMPLIANCE}_{\text{S}}^{\text{ow}}$ holds, otherwise we perform the same procedure as for the constant-free case. This preprocessing step can be done in polynomial time and in AC^0 in data complexity. \square

Next, we provide matching Σ_4^p and Σ_2^p lower bounds for the combined complexity of the non-strict version of the problem. The Σ_2^p lower bound holds already under the open-world assumption; furthermore, it also applies already to ground RDF PPDP instances and Boolean policies. Recall that, under those assumptions, the compliance checking problem COMPLIANCE is of lower complexity (CONP -complete as established in Proposition 2). Note also that the Σ_4^p bound is claimed only for non-ground PPDP instances; in this paper, we leave open the exact complexity of the problem for the settings where instances are ground, which we believe is an interesting problem for future work.

Lemma 6. *The following holds:*

1. $\text{MIN-COMPLIANCE}^{\text{ow}}$ is Σ_2^p -hard for ground RDF PPDP instances and Boolean policies; and
2. MIN-COMPLIANCE is Σ_4^p -hard for RDF PPDP instances, and fixed, quantifier-free and atomic CQs in closures.

Proof. We start with statement 1 and show Σ_2^p -hardness of $\text{MIN-COMPLIANCE}^{\text{ow}}$ by means of a reduction of the canonical Σ_2^p -complete problem $\exists\forall 3\text{SAT}$. Again, we use only unary and binary predicates in the reduction, and the adaptation to the case of RDF is as usual. After this, we will generalise the reduction by using the ideas in the proof of Lemma 3 to show statement 2 on Σ_4^p -hardness of MIN-COMPLIANCE .

Let $\phi = \exists \bar{r}. \forall \bar{v}. \neg \psi(\bar{r}, \bar{v})$, where \bar{r} and \bar{v} are tuples of distinct variables, and ψ is a conjunction of clauses, where each clause is a disjunction of three literals over $\bar{r} \cup \bar{v}$. We construct a dataset \mathcal{D}_0 , a Boolean policy p and an integer ℓ such that there exists a \mathcal{D}_0 -suppressor f of cost at most ℓ with (\mathcal{D}_0, f, p) compliant if and only if ϕ is valid.

Let us first set $\ell = 2 \cdot |\bar{r}|$.

Our construction of \mathcal{D}_0 and p relies on unary predicates R and U , binary predicates Cl_γ for each clause γ in ψ , and binary predicates $\text{Arg}_1, \text{Arg}_2, \text{Arg}_3, \text{Back}_1^{\text{false}}, \text{Back}_1^{\text{true}}$ and Back_2 . We construct \mathcal{D}_0 in two steps: in the first one we construct an intermediate dataset \mathcal{D}'_0 from which \mathcal{D}_0 is obtained in the second one. Let \mathcal{D}'_0 consist of the following atoms, where we use constants $c^{\text{false}}, c^{\text{true}}, c^y, c^z$, constants c^π_γ for every clause γ and each (of at most 7) assignment π of the variables of γ that satisfies γ , constants c_w^{false} and c_w^{true} for each $w \in \bar{r} \cup \bar{v}$, constants d_r^y for every $r \in \bar{r}$, and constants d^z, d^{cl} , and d^{var} :

- $R(c^{\text{false}})$ and $R(c^{\text{true}})$;
- $\text{Back}_1^{\text{false}}(c^{\text{false}}, c^y), \text{Back}_1^{\text{true}}(c^{\text{true}}, c^y)$, and $\text{Back}_2(c^y, c^z)$;

- $\text{Cl}_\gamma(c^\mathbf{z}, c_\gamma^\pi), \text{Arg}_1(c_\gamma^\pi, c_{w_1}^{\pi(w_1)}), \dots, \text{Arg}_3(c_\gamma^\pi, c_{w_3}^{\pi(w_3)})$ for each clause γ in ψ over variables w_1, \dots, w_3 and each assignment π of w_1, \dots, w_3 satisfying γ ;
- $\text{R}(c_r^{\text{false}})$ and $\text{R}(c_r^{\text{true}})$ for each $r \in \bar{r}$;
- $\text{Back}_1^{\text{false}}(c_r^{\text{false}}, d_r^\mathbf{y}), \text{Back}_1^{\text{true}}(c_r^{\text{true}}, d_r^\mathbf{y})$, and $\text{Back}_2(d_r^\mathbf{y}, d^\mathbf{z})$, for each $r \in \bar{r}$;
- $\text{Cl}_\gamma(d^\mathbf{z}, d^{\text{cl}})$, for each clause γ in ψ ; and
- $\text{Arg}_1(d^{\text{cl}}, d^{\text{var}}), \dots, \text{Arg}_3(d^{\text{cl}}, d^{\text{var}})$, and $\text{U}(d^{\text{var}})$.

Next, we obtain \mathcal{D}_0 from \mathcal{D}'_0 by making $2 \cdot |\bar{r}| + 1$ copies of all constants of \mathcal{D}'_0 except c_r^{false} and c_r^{true} for all $r \in \bar{r}$, and all atoms involving these constants—that is, \mathcal{D}_0 uses constants

- $c_1, \dots, c_{2 \cdot |\bar{r}| + 1}$ for each constant c in \mathcal{D}'_0 different from all c_r^{false} and c_r^{true} , and
- c_r^{false} and c_r^{true} for each $r \in \bar{r}$,

and consists of atoms $\text{P}(\bar{c}_1), \dots, \text{P}(\bar{c}_{2 \cdot |\bar{r}| + 1})$ for each atom $\text{P}(\bar{c})$ in \mathcal{D}'_0 , where each \bar{c}_i is obtained from \bar{c} by replacing each c different from all c_r^{false} and c_r^{true} by c_i (therefore, the only atoms that stay intact are $\text{R}(c_r^{\text{false}})$ and $\text{R}(c_r^{\text{true}})$).

Next, the policy p is the Boolean CQ with the following atoms, where all other arguments are existential variables:

- $\text{R}(x^{\text{false}})$ and $\text{R}(x^{\text{true}})$,
- $\text{Back}_1^{\text{false}}(x^{\text{false}}, y), \text{Back}_1^{\text{true}}(x^{\text{true}}, y)$, and $\text{Back}_2(y, z)$,
- $\text{Cl}_\gamma(z, x_\gamma), \text{Arg}_1(x_\gamma, x_{w_1}), \dots, \text{Arg}_3(x_\gamma, x_{w_3})$, for each clause γ in ψ over propositional variables w_1, \dots, w_3 , and
- $\text{R}(x_r)$, for each $r \in \bar{r}$.

We claim that ϕ is valid if and only if there exists a \mathcal{D}_0 -suppressor f of cost at most ℓ such that (\mathcal{D}_0, f, p) is compliant. The intuition is as follows. First, note that if a suppressor f does not anonymise at least one of the positions of $\text{R}(c_r^{\text{false}})$ and $\text{R}(c_r^{\text{true}})$ for every existentially quantified variable r , then the resulting anonymisation is not compliant; indeed, in such case there exists homomorphism from p sending x^{false} and x^{true} to c_r^{false} and c_r^{true} , respectively, for this r , and all other variables to the d constants. Therefore, at least one of these positions should be anonymised for each r , and at most one of them is possible because of the maximal cost ℓ . Hence, each of these anonymisations corresponds to an assignment of variables \bar{r} . Then, a possible homomorphism from p corresponds to an extension of the assignment to \bar{v} satisfying ψ . The correctness of this intuition follows from the correctness of the more general Σ_4^p reduction for statement 2, so we postpone it until the end of this proof.

Next we prove statement 2 of the theorem. In particular, we show Σ_4^p -hardness of MIN-COMPLIANCE by means of a reduction of the canonical Σ_4^p -complete $\exists\forall\exists\forall\text{3SAT}$ problem. Again, we use only unary and binary predicates in the reduction, and the adaptation to the case of RDF is as usual.

Let $\phi = \exists \bar{r}. \forall \bar{s}. \exists \bar{u}. \forall \bar{v}. \neg \psi(\bar{r}, \bar{s}, \bar{u}, \bar{v})$, where \bar{r} , \bar{s} , \bar{u} and \bar{v} are tuples of distinct propositional variables, and ψ is a conjunction of clauses, where each clause is a disjunction of three literals over $\bar{r} \cup \bar{s} \cup \bar{u} \cup \bar{v}$. We need to construct a dataset \mathcal{D}_0 , a policy p , an integer ℓ , and a set of quantifier-free and atomic closures \mathcal{C} with queries not dependent on ϕ such that there exists a \mathcal{D}_0 -suppressor f of cost at most ℓ with (\mathcal{D}_0, f, p) compliant with respect to \mathcal{C} if and only if ϕ is valid. Instead of giving the construction from scratch, we build upon the Σ_2^p -hardness proof for statement 1 of this theorem and concentrate on the additions required for the construction. In fact, these additions are very similar to the construction in the proof of Lemma 3 on Π_3^p -hardness of COMPLIANCE in its general form: propositional variables \bar{s} correspond to answer variables of the policy, while each propositional variable in \bar{u} is encoded by a pair of nulls as in the anonymisation in the proof of Lemma 3.

In comparison to the Σ_2^p -hardness proof of statement 1, the cost should take into account the nulls corresponding to \bar{u} variables. We set $\ell = 2 \cdot |\bar{r}| + (2 \cdot |\bar{r}| + 1) \cdot (6 \cdot |\bar{u}| + N(\psi, \bar{u}))$, where $N(\psi, \bar{u})$ is the number of (occurrences of) literals in ψ with variables in \bar{u} .

Additionally to the predicates introduced in the reduction for statement 1, in the construction of \mathcal{D}_0 and p we will use a binary predicate **UValues**. We construct \mathcal{D}_0 again in two steps. Let \mathcal{D}'_0 be the same as before except that

- \mathcal{D}'_0 additionally uses constants c_s^{false} and c_s^{true} for each $s \in \bar{s}$ and nulls $\mathbf{b}_u^{\text{false}}$ and $\mathbf{b}_u^{\text{true}}$ for each $u \in \bar{u}$,
- instead of the Arg_i atoms in the construction for statement 1, dataset \mathcal{D}'_0 has atoms $\text{Arg}_1(c_\gamma^\pi, t_{w_1}^{\pi(w_1)}), \dots, \text{Arg}_3(c_\gamma^\pi, t_{w_3}^{\pi(w_3)})$ for each clause γ in ψ over variables w_1, \dots, w_3 and each assignment π of w_1, \dots, w_3 satisfying γ , where $t_w^{\pi(w)}$ is $\mathbf{b}_w^{\pi(w)}$ if $w \in \bar{u}$ and $c_w^{\pi(w)}$ if $w \in \bar{r} \cup \bar{s} \cup \bar{v}$, and
- \mathcal{D}'_0 additionally has atoms **UValues**($\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}}$) for every $u \in \bar{u}$.

Then, \mathcal{D}_0 is obtained from \mathcal{D}'_0 in the same way as before, by making $2 \cdot |\bar{r}| + 1$ copies of all constants and nulls of \mathcal{D}'_0 except c_r^{false} and c_r^{true} for $r \in \bar{r}$, and all atoms involving these constants and nulls.

Next, the policy p is the same CQ as before except that it has answer variables x_s , for $s \in \bar{s}$ (and all other arguments are existential variables) and additionally has atoms **U**(x_u), for each $u \in \bar{u}$.

Finally, let \mathcal{C} consist of the following two closures with atomic quantifier-free CQs, where c and c' are two fresh constants:

- [**UValues**(x, y), $\{(c, c'), (c', c)\}$], and
- [**U**(x), $\{c\}$].

Having the construction completed, we note that, in contrast to the Π_3^p reduction in the proof of Lemma 3, we have the (copies of the) $\mathbf{b}_u^{\text{false}}$ and $\mathbf{b}_u^{\text{true}}$ already anonymised in \mathcal{D}_0 . This allows us to use only one pair of constants c and c' in the closures \mathcal{C} instead of a separate pair for each $u \in \bar{u}$ in the Π_3^p reduction. Intuitively, this construction is a combination of constructions in the proofs of Lemma 3 and statement 1 of this theorem: for each variable r one and only one of $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$ should be anonymised in

such a way that compliance holds for each answer tuple consisting of the c_s^{false} and c_s^{true} to the policy, which means that there should exist a re-identification of each pair $\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}}$ to c, c' or c', c , such that the variables x_v cannot be sent to the c_v^{false} and c_v^{true} with an appropriate homomorphism from the policy. Next we formally proof this intuition correct.

Assume first that ϕ is valid—that is, there exists a truth assignment σ of \bar{r} such that for any of its extensions to \bar{s} there is a further extension to \bar{u} such that for any further extension to \bar{v} the resulting assignment turns ψ to **false**. Consider the \mathcal{D}_0 -suppressor f that anonymises, for each r such that $\sigma(r) = \text{false}$, the only position in atom $R(c_r^{\text{true}})$ to a fresh null \mathbf{b}_r and, for each r such that $\sigma(r) = \text{true}$, the only position in atom $R(c_r^{\text{false}})$ to a fresh null \mathbf{b}_r . Since $\ell = 2 \cdot |\bar{r}| + (2 \cdot |\bar{r}| + 1) \cdot (6 \cdot |\bar{u}| + N(\psi, \bar{u}))$ and \mathcal{D}_0 contains $2 \cdot |\bar{r}| + 1$ copies of nulls $\mathbf{b}_u^{\text{false}}$ and $\mathbf{b}_u^{\text{true}}$ for every $u \in \bar{u}$ with each copy contributing with $6 \cdot |\bar{u}| + N(\psi, \bar{u})$ to f , the cost of f is ℓ . We claim that the anonymisation $f(\mathcal{D}_0)$ complies with p with respect to \mathcal{C} . The proof of this fact goes along the same lines as the forward direction of the correctness proof in Lemma 3 with four small exceptions: first, when constructing the Herbrand interpretations I (in two places), we assume that each \mathbf{b}_r is represented by a fresh constant, and so each $r \in \bar{r}$ has to be sent to $c_r^{\sigma(r)}$ by a homomorphism from $p(\bar{c})$ to $f(\mathcal{D}_0)$; second, we first argue that if \bar{c} does not consist of the c_s^{false} or c_s^{true} from the same copy of the construction then the claim on existence of a model I is trivial; third, instead of c_u and c'_u we use c and c' for each $u \in \bar{u}$; finally, all copies of every $\mathbf{b}_u^{\text{false}}$ and $\mathbf{b}_u^{\text{true}}$ are sent to c or c' in the same way, according to the extension of σ .

For the converse direction, let f be a \mathcal{D}_0 -suppressor of cost at most ℓ that complies with p . Consider the \mathcal{D}_0 -suppressor f' that agrees with f on all positions involving atoms $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$, but keeps all other positions as they are in \mathcal{D}_0 . The cost of f' is not higher than the cost of f by construction, and f' also complies with p : indeed, dataset \mathcal{D}_0 has $2 \cdot |\bar{r}| + 1$ “copies” of the rest of the construction and the copies of the nulls $\mathbf{b}_u^{\text{false}}, \mathbf{b}_u^{\text{true}}$ contribute to the cost with at least $(2 \cdot |\bar{r}| + 1) \cdot (6 \cdot |\bar{u}| + N(\psi, \bar{u}))$, so “breaking” at most $2 \cdot |\bar{r}|$ copies cannot change the presence of a homomorphism from $p(\bar{c})$ for any \bar{c} . Next, suppressor f' anonymises (the only position of) at least one of $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$ for each $r \in \bar{r}$, because otherwise there is a homomorphism from $p(\bar{d}^{\text{var}})$, for \bar{d}^{var} the tuple consisting of several d^{var} of the same size as \bar{s} , to the anonymisation $f'(\mathcal{D}_0)$ sending

- x^{false} and x^{true} to c_r^{false} and c_r^{true} , respectively, where r is a variable with both $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$ not anonymised by f' ,
- y and z to (the first copies of) d_r^y and d^z ,
- all x_γ and all x_w , for $w \in \bar{r} \cup \bar{u} \cup \bar{v}$ to (the first copies of) d^{cl} and d^{var} , respectively.

Since $\ell = 2 \cdot |\bar{r}| + (2 \cdot |\bar{r}| + 1) \cdot (6 \cdot |\bar{u}| + N(\psi, \bar{u}))$, the only possibility is that f' is f and it anonymises exactly one of $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$ for each $r \in \bar{r}$. Consider the assignment σ of \bar{r} that sends each r to **true** if and only if $R(c_r^{\text{false}})$ is anonymised by f . We claim that for any extension of σ to \bar{s} there exists a further extension to \bar{u} such that for any further extension to \bar{v} the resulting assignment turns ψ to **false**—that is, that ϕ is valid. The proof of this fact goes along the same lines as the backward direction of the correctness proof in Lemma 3 with two small exceptions: first, we consider the first copies of constants and nulls in all relevant cases; second, as in the forward direction proof, we take c and c' instead of all c_u and c'_u , respectively. \square

We now provide the matching Σ_4^p and Σ_2^p lower bounds for the combined complexity of the strict version of the problem. The Σ_2^p bound holds already under the open-world assumption and for ground RDF PPDP instances. Note that, in contrast to Lemma 6, we do not restrict ourselves to Boolean policies even in the Σ_2^p case. This is justified by the fact that strict suppressors cannot change the value of constant-free Boolean queries—that is, if a Boolean constant-free policy p holds for a ground dataset \mathcal{D}_0 and f is a strict \mathcal{D}_0 -suppressor, then p also holds in the corresponding anonymisation $f(\mathcal{D}_0)$. In fact, there is a very simple polynomial algorithm that decides MIN-COMPLIANCE_s for Boolean policies, which is based on the following trivial fact: there is a witnessing \mathcal{D}_0 -suppressor within the cost limit ℓ if and only if there is a constant c mentioned in the policy with anonymisation cost impact at most ℓ .

Lemma 7. *The following holds:*

1. $\text{MIN-COMPLIANCE}_s^{\text{ow}}$ is Σ_2^p -hard for ground RDF PPDP instances, and it is CONP -hard if, additionally, the policies are Boolean; and
2. MIN-COMPLIANCE_s is Σ_4^p -hard for RDF PPDP instances, fixed, quantifier-free and atomic CQs in closures, and it is Σ_2^p -hard if, additionally, the policies are Boolean.

Proof. We first concentrate on the first parts of both statements. The proof relies on the same idea as the proof of Lemma 6, so we concentrate only on the differences between the constructions.

Strict suppressors differ from non-strict ones by the fact that they anonymise not positions, but constants. This has the following consequences for the reduction in Lemma 6:

1. as already mentioned, applying a strict suppressor to a dataset cannot change the answer to any constant-free Boolean query, including the policy in the proof of statement 1 of Lemma 6;
2. making several copies of atoms over Arg_j increases the cost impact of an anonymisation of participating c_r^{false} and c_r^{true} ; and
3. anonymisations of c_r^{false} and c_r^{true} may have different impact on the cost, because they participate in possibly a different number of satisfying assignments of clauses.

To take into account these differences, we make the following modifications to the construction in the proof of Lemma 6.

1. We make variables x_r , $r \in \bar{r}$, of policy p answer variables; as a result, nulls introduced by the anonymisation do not participate in any answer to the policy.
2. Instead of multiplying $2 \cdot |\bar{r}| + 1$ times all the constants different from c_r^{false} and c_r^{true} with all the atoms over these constants, we increase the cost impact of each such constant c to make the impact over $2 \cdot |\bar{r}|$ alone by adding to \mathcal{D}_0 new atoms of the form $P_1(c, d)$, where P_1 is a fresh binary predicate and d is a fresh constant for each such atom. In particular, if c originally participates in k atoms (i.e., its original strict anonymisation would have cost impact $k + 1$), then we add $2 \cdot |\bar{r}| - k$ new atoms with c . Note that we do not need to do the same with these new d by themselves, because

P_1 does not appear in policy p , and therefore the anonymisation of a d does change the answers to p .

3. To make the cost impact of anonymisations of c_r^{false} and c_r^{true} equal, we add to \mathcal{D}_0 the necessary number of new atoms of the form $P_2(c_r^t, d)$ for each $r \in \bar{r}$ and each $t \in \{\text{false}, \text{true}\}$, where P_2 is again a fresh binary predicate and d is a fresh constant for each such atom. In particular, if c_r^t originally participates in k atoms, then we add $3n + 2 - k$ new atoms with c_r^t , where n is the number of clauses in ψ ($3n + 2$ is the maximal number of positions for any of these constants in the original dataset). Again, we do not need to do the previous step with these new d . Finally, we set $\ell = |\bar{r}| + (3n + 2) \cdot |\bar{r}| + (6 \cdot |\bar{u}| + N(\psi, \bar{u}))$, which is the cost of a strict anonymisation that sends to a null exactly one of c_u^{false} and c_u^{true} for each u and keeps everything else intact.

With these modifications, the rest of the proof is completely analogous to the proof of Lemma 6 for both statements, so we omit it for brevity.

To complete the proof, we note that the bounds for the cases with Boolean policies follow immediately from Proposition 2 and the second part of Lemma 3: it is enough to take $\ell = 0$ and, in the second case, $f(\mathcal{D}_0)$ as the dataset. \square

We now proceed to the study of data complexity, and provide matching NP lower bounds to the upper bounds in Lemma 5. These lower bounds hold already under the open-world assumption, and hence transfer immediately to the general setting. We start with the non-strict version of the problem.

Lemma 8. $\text{MIN-COMPLIANCE}^{\text{ow}}$ is NP-hard in data complexity for ground RDF PPDP instances and Boolean policies.

Proof. The proof is by reduction of the node deletion problem $\text{NODE-DELETION}_{\Pi}$ for Π the property of a graph not having cycles of length 3. The input to this problem is a directed graph \mathcal{G} and an integer k , and the question is whether we can obtain a graph \mathcal{G}' by deleting at most k nodes from \mathcal{G} such that \mathcal{G}' satisfies Π . This is an NP-complete problem (Yannakakis, 1978).

In the reduction, we use a unary predicate U and two binary predicates Edge_1 and Edge_2 . The adaptation to RDF case is again straightforward. First, let p be the following Boolean policy, which we consider fixed:

$$\begin{aligned} \exists x_1, x_2, x_3, y_1, y_2, y_3. & U(x_1) \wedge U(x_2) \wedge U(x_3) \wedge \\ & \text{Edge}_1(x_1, y_1) \wedge \text{Edge}_2(y_1, x_2) \wedge \text{Edge}_1(x_2, y_2) \wedge \text{Edge}_2(y_2, x_3) \wedge \\ & \text{Edge}_1(x_3, y_3) \wedge \text{Edge}_2(y_3, x_1). \end{aligned}$$

Let \mathcal{G} , k constitute an instance of $\text{NODE-DELETION}_{\Pi}$. We construct a ground dataset \mathcal{D}_0 and an integer ℓ such that it is possible to obtain \mathcal{G}' by deleting at most k nodes from \mathcal{G} such that \mathcal{G}' has no cycles of length 3 if and only if there exists a \mathcal{D}_0 -suppressor f of cost at most ℓ such that $f(\mathcal{D}_0)$ complies with p .

Let $\ell = 2k$. Dataset \mathcal{D}_0 contains the following:

- for each node v of \mathcal{G} , an atom $\mathbf{U}(c_v)$ with c_v a constant uniquely associated with v ,
- for each edge $e = (u, v)$, the atoms

$$\begin{aligned} &\text{Edge}_1(c_u, d_e^1), \quad \dots, \quad \text{Edge}_1(c_u, d_e^{\ell+1}), \\ &\text{Edge}_2(d_e^1, c_v), \quad \dots, \quad \text{Edge}_2(d_e^{\ell+1}, c_v), \end{aligned}$$

where $d_e^1, \dots, d_e^{\ell+1}$ are constants uniquely associated with e .

Next we prove the correctness of the reduction.

Let \mathcal{G}' be a graph obtained from \mathcal{G} by deleting $k' \leq k$ nodes and that does not have a cycle of length 3. Consider a \mathcal{D}_0 -suppressor f that sends the only position in each atom $\mathbf{U}(c_v)$ with v not in \mathcal{G}' to a fresh null and keeps all other positions as in \mathcal{D}_0 . On the one hand, the cost of f is $2k' \leq \ell$. On the other hand, by construction, p evaluates over $f(\mathcal{D}_0)$ to **false**—that is, $f(\mathcal{D}_0)$ complies with p .

Let f be a \mathcal{D}_0 -suppressor of cost at most ℓ that complies with p . Consider the \mathcal{D}_0 -suppressor f' that agrees with f on all positions involving atoms in \mathbf{U} , but does not anonymise positions in Edge_1 and Edge_2 atoms. The cost ℓ' of f' is not higher than the cost ℓ of f by construction, and f' also complies with p : indeed, \mathcal{D}_0 has $\ell + 1$ “copies” of each edge, so modifying at most ℓ copies cannot change the presence of a homomorphism from p . Consider now a graph \mathcal{G}' obtained from \mathcal{G} by deleting all nodes v such that the only position in $\mathbf{U}(c_v)$ is anonymised by f' . The number of deleted nodes is $\ell'/2 \leq k$, as required. Also, by construction, \mathcal{G}' does not have a cycle of length 3, witnessing a positive answer to NODE-DELETION_Π . \square

We now provide a matching data complexity lower bound for the strict version of the cost minimisation problem for compliance. Our bound applies already under the open-world assumption, and hence extends trivially to the general setting. By the same reason as in Lemma 7, our bound applies only to non-Boolean policies.

Lemma 9. *The following holds:*

1. $\text{MIN-COMPLIANCE}_S^{\text{ow}}$ is NP-hard in data complexity for ground RDF PPDP instances;
2. MIN-COMPLIANCE_S is NP-hard in data complexity for RDF PPDP instances, Boolean policies, and quantifier-free and atomic CQs in closures.

Proof. The proof of statement 1 can be obtained as a modification of the proof for Lemma 8. However, it is more convenient in this case to provide the whole construction from the scratch rather than simply pointing out the required modifications.

The proof is again by reduction of NODE-DELETION_Π for Π the property of a graph not having cycles of length 3. In the reduction, we use two binary predicates **Edge** and **P** (so, the adaptation to RDF case is again straightforward). First, consider the following existential-free policy p , which we consider fixed:

$$\text{Edge}(x_1, x_2) \wedge \text{Edge}(x_2, x_3) \wedge \text{Edge}(x_3, x_1).$$

Given an instance \mathcal{G} , k of NODE-DELETION_Π problem, we construct a ground dataset \mathcal{D}_0 and an integer ℓ as follows. First, we let $\ell = (2n + 1)k$, where n is the number of nodes

in \mathcal{G} . Then, for each edge $e = (u, v)$ of \mathcal{G} , dataset \mathcal{D}_0 contains the atom $\text{Edge}(c_u, c_v)$, where c_u and c_v are constants associated with nodes u and v . Also, for each node v of \mathcal{G} with m_v the number of incoming to and outgoing edges from v , \mathcal{D}_0 contains atoms $\text{P}(c_v, d_i)$, for $i = m_v + 1, \dots, 2n$ and d_i a fresh constant.

The proof of correctness of the reduction is analogous to the proof of Lemma 8, so we omit it for brevity. We simply point out that, for every node v , the aforementioned number m_v is at most $2n$, so atoms over the relation P make the cost impact of anonymisation of any c_v equal to $2n + 1$. This also defines the value of ℓ above, which allows to anonymise at most k constants c_v , which corresponds to deleting at most k nodes from \mathcal{G} .

Statement 2 follows immediately from Lemma 4: same as in the data complexity case in Lemma 7, it is enough to take $\ell = 0$ and $f(\mathcal{D}_0)$ as the dataset. \square

The following theorem summarises the results of Lemmas 5, 6, 7, 8, and 9 and settles the complexity of the cost minimisation problems associated to policy compliance.

Theorem 2. *The following holds:*

1. $\text{MIN-COMPLIANCE}^{\text{ow}}$ is Σ_2^p -complete and NP-complete in data complexity; the lower bounds hold already for ground RDF PPDP instances and for Boolean policies;
2. $\text{MIN-COMPLIANCE}_s^{\text{ow}}$ is Σ_2^p -complete and NP-complete in data complexity; the problem is coNP-complete and in AC^0 in data complexity for Boolean policies; all the lower bounds hold already for ground RDF PPDP instances;
3. MIN-COMPLIANCE is Σ_4^p -complete and NP-complete in data complexity; the Σ_4^p lower bound holds already for RDF PPDP instances and fixed, quantifier-free and atomic CQs in closures; the NP lower bound holds already for ground RDF PPDP instances and Boolean policies; the problem is Σ_2^p -complete for Boolean policies; the lower bound holds already for ground RDF PPDP instances;
4. MIN-COMPLIANCE_s is Σ_4^p -complete and NP-complete in data complexity; the Σ_4^p lower bound holds already for RDF PPDP instances and fixed, quantifier-free and atomic CQs in closures; the NP lower bound holds already for RDF PPDP instances and Boolean policies; the problem is Σ_2^p -complete for Boolean policies; the lower bound holds already for RDF PPDP instances.

5. Linkage Safety

In this section, we establish complexity bounds for the decision problems associated to linkage safety. Our results are summarised in Table 2 in Section 3.5.

In Section 5.1 we consider the safety checking problem in both its open-world and general variants. Subsequently, in Section 5.2 we study the cost minimisation problem associated to linkage safety across the two dimensions specified in Definition 10: on the one hand, we consider open-world vs. general semantics and, on the other hand, whether suppressors are required to be strict or not.

The reader may have already observed in Table 2 that our combined complexity bounds for all problems associated to linkage safety under the open-world assumption are not

tight, and we leave their precise complexity open; these open-world problems are intimately related to the critical tuple problem in database theory (Miklau & Suciu, 2007), the precise complexity of which has proved elusive. In contrast, all our data complexity bounds as well as all our combined complexity bounds for general, closed-world, settings are tight.

5.1 Checking Linkage Safety

In this section, we focus our attention to the safety checking problem in both its open-world and general variants. We show the following results:

- the open-world variant of the problem is solvable in Π_2^P and in AC^0 in data complexity; concerning lower bounds in combined complexity, the problem is DP-hard and also at least as hard as the complement of the critical tuple problem;
- the general variant (i.e., with closed-world information) is CONEXPTIME-complete in combined complexity and Π_2^P -complete in data complexity.

The lower bounds hold also under various restrictions on datasets, suppressors, policies, and closures, which we discuss independently in each particular case.

We first discuss the upper bounds and start by considering the open-world setting. Recall that the safety condition involves a universal quantification over all possible datasets (infinitely many, and of unbounded size) that could be linked with the anonymised data that we wish to publish (see Definition 4). The crucial observation is that, under the open-world assumption, we can restrict ourselves to consider only external datasets that are polynomially bounded in the size of the input policy query.

Lemma 10. *SAFETY^{ow} is in Π_2^P and in AC^0 in data complexity.*

Proof. Recall that an input PPDP setting (\mathcal{D}_0, f, p) is safe if, for every external dataset \mathcal{D}' such that p has empty answers over \mathcal{D}' , it holds that p also has empty answers over $f(\mathcal{D}_0) \cup \mathcal{D}'$. We argue that, to check the aforementioned condition, it suffices to consider only external datasets \mathcal{D}' of size bounded by the number of atoms in p —that is, if there is an external dataset \mathcal{D}' witnessing the violation of the safety requirement, then there exists one such witness having size at most the size of p . To prove this fact, assume that there exists a dataset \mathcal{D}'' such that $\mathcal{D}'' \not\models p(\bar{c})$ for any tuple of constants \bar{c} , but $f(\mathcal{D}_0) \cup \mathcal{D}'' \models p(\bar{c})$ for some \bar{c} . The latter implies that there must exist a homomorphism g from $p(\bar{c})$ into $f(\mathcal{D}_0) \cup \mathcal{D}''$, so let $\mathcal{D}' \subseteq \mathcal{D}''$ be the homomorphic image of $p(\bar{c})$ over g . Clearly, \mathcal{D}' satisfies all the requirements by construction.

Consequently, we can decide SAFETY^{ow} in Π_2^P by universally checking all \mathcal{D}' of size at most $|p|$ and then check that either \mathcal{D}' does not comply with p or $f(\mathcal{D}_0) \cup \mathcal{D}'$ complies with p , both of which can be done using a call to an NP oracle by Proposition 2.

Moreover, if the policy is fixed, then we can rewrite all relevant possibilities of \mathcal{D}' together with the policy into a fixed first-order logic sentence; checking whether this sentence holds in $f(\mathcal{D}_0)$ is possible in AC^0 (Immerman, 1987; Abiteboul et al., 1995). \square

Lifting the open-world assumption has a significant impact on the complexity of safety checking. In contrast to the open-world case, where we could restrict ourselves to consider external datasets of polynomial size, in the general case we will be forced to consider datasets

of exponential size. We can then obtain a CONEXPTIME upper bound by providing an algorithm for the complement of safety that non-deterministically guesses a witness dataset of exponential size and then checks (in exponential time in the size of the original input) that it satisfies the required properties for safety violation.

Towards providing the aforementioned upper bound it will be convenient to use the following simple lemma, which shows that in case of Boolean policies we can restrict ourselves to null-free witnesses.

Lemma 11. *Let (\mathcal{D}_0, f, p) be a PPDP instance with Boolean p and let \mathcal{C} be a set of closures. If there exists a dataset \mathcal{D}' such that $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$, then there exists a ground dataset with the same properties as \mathcal{D}' and containing at most as many atoms as \mathcal{D}' .*

Proof. Consider a model I of $\mathcal{D}' \cup \mathcal{C}$ such that $I \not\models p$. Since I is a model of \mathcal{D}' , there is a homomorphism from \mathcal{D}' to I , and we can take as \mathcal{D}'' the ground dataset corresponding to the image of this homomorphism. By construction, $\mathcal{D}'' \cup \mathcal{C} \not\models p$ and $\mathcal{D}'' \cup \mathcal{C} \models \mathcal{D}' \cup \mathcal{C}$, and the latter together with $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$ implies that $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C} \models p$. In other words, \mathcal{D}'' is a witness for non-safety of (\mathcal{D}_0, f, p) with respect to \mathcal{C} . \square

We are now ready to establish a CONEXPTIME upper bound for SAFETY. We show that, in search for a counterexample dataset \mathcal{D}' for safety over a Boolean policy, we can restrict ourselves to datasets of exponential size. Intuitively, all we need from \mathcal{D}' is, on the one hand, that it does not witness the policy by itself and, on the other hand, that it witnesses the policy once it is merged with any minimal model of the anonymisation and the closures. Since there are exponentially many such minimal models, corresponding to all possible mergings of the nulls of the anonymisation with constants in the answers of the closures, the maximal required size for \mathcal{D}' is exponential.

Lemma 12. SAFETY is in CONEXPTIME.

Proof. We first note that, by definition, if a PPDP instance (\mathcal{D}_0, f, p) with non-Boolean p is not safe, then there is a tuple of constants \bar{c} such that $(\mathcal{D}_0, f, p(\bar{c}))$ is not safe. Therefore, if we have a CONEXPTIME algorithm for SAFETY restricted to inputs with Boolean policies, then we can easily design a CONEXPTIME algorithm for the general case: such an algorithm runs the restricted procedure for each Boolean policy obtained by replacing the answer variables with a tuple of constants of the size equal to the number of these variables. Therefore, in the rest of the proof we concentrate on the Boolean policies.

We prove the claim in two steps: first, we show that if there exists a counterexample dataset for safety of a PPDP instance with respect to a set of closures, then there exists a ground counterexample of exponential size; second, we show that a ground counterexample of exponential size can be verified in exponential time.

For the first step, consider a PPDP instance (\mathcal{D}_0, f, p) and a set of closures \mathcal{C} . Let a dataset \mathcal{D}' be a counterexample for the safety of (\mathcal{D}_0, f, p) with respect to closures \mathcal{C} —that is, $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. By Lemma 11 we may assume that \mathcal{D}' is ground. On the base of (\mathcal{D}_0, f, p) , \mathcal{C} , and \mathcal{D}' , we build a ground dataset \mathcal{D}'' of exponential size and then show that it also constitutes a counterexample.

For q a Boolean query, let \mathcal{D}_q the dataset obtained from the atoms in q by replacing each variable with a fresh null. Then, let $\mathcal{D}_{\mathcal{C}}$ be the dataset defined as the union of all

the datasets $\mathcal{D}_{q(\bar{c})}$ where $[q, Ans] \in \mathcal{C}$ and $\bar{c} \in Ans$. Let \mathbb{D} be the collection of all ground datasets \mathcal{D} obtained from $f(\mathcal{D}_0) \cup \mathcal{D}_{\mathcal{C}}$ by identifying some nulls with constants mentioned in the answers Ans in \mathcal{C} and replacing all other nulls by fresh constants. Note that the number of datasets in \mathbb{D} is exponential in the number of nulls in $f(\mathcal{D}_0) \cup \mathcal{D}_{\mathcal{C}}$ and polynomial in the number of constants in \mathcal{C} . Furthermore, by construction, for any $\mathcal{D} \in \mathbb{D}$ we have that $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C} \models f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$. Since $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$, there are two options:

- either $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C}$ is inconsistent, in which case there exists $[q, Ans] \in \mathcal{C}$ and a tuple of constants $\bar{c} \notin Ans$ with a homomorphism g from $q(\bar{c})$ to $\mathcal{D} \cup \mathcal{D}'$;
- or $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C}$ is consistent, and there is a homomorphism g from p to $\mathcal{D} \cup \mathcal{D}'$.

In any case, for each $\mathcal{D} \in \mathbb{D}$, let us pick an arbitrary homomorphism g with the relevant properties and denote with \mathcal{D}_g the dataset $\mathcal{D}' \cap g(q(\bar{c}))$ (in the first case) and the dataset $\mathcal{D}' \cap g(p)$ (in the second case). Finally, we let $\mathcal{D}'' = \bigcup_{\mathcal{D} \in \mathbb{D}} \mathcal{D}_g$.

Next we show that \mathcal{D}'' is indeed a required counterexample for safety. First, note that \mathcal{D}'' is at most of exponential size in the size of (\mathcal{D}_0, f, p) and \mathcal{C} : indeed, there are exponentially many datasets in \mathbb{D} and each of them captures at most linearly many atoms from \mathcal{D}' to \mathcal{D}'' . Next, since $\mathcal{D}'' \subseteq \mathcal{D}'$ and $\mathcal{D}' \cup \mathcal{C} \not\models p$, we have that $\mathcal{D}'' \cup \mathcal{C} \not\models p$. So, we are left to show that $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C} \models p$.

To this end, assume for the sake of contradiction that $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C} \not\models p$ and consider a model I of $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C}$ not satisfying p . Since I is a model of $f(\mathcal{D}_0) \cup \mathcal{C}$, there is a homomorphism h from $f(\mathcal{D}_0) \cup \mathcal{D}_{\mathcal{C}}$ to I . Consider the dataset $\mathcal{D} \in \mathbb{D}$ obtained from $f(\mathcal{D}_0) \cup \mathcal{D}_{\mathcal{C}}$ by identifying a null \mathbf{b} with $h(\mathbf{b})$ if $h(\mathbf{b})$ is a constant mentioned in the answers Ans in \mathcal{C} or by replacing \mathbf{b} with the constant $c_{\mathbf{b}}$ otherwise. On the one hand, $\mathcal{D} \cup \mathcal{D}'' \cup \mathcal{C} \not\models p$, because $I \not\models p$ and all the new constants in \mathcal{D} are not mentioned in p . On the other hand, $\mathcal{D} \cup \mathcal{D}'' \cup \mathcal{C} \models p$, because $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C} \models p$ and the required part of the witness for this in \mathcal{D}' (either for inconsistency in $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C}$ or for p itself) is in \mathcal{D}'' by construction. So, we arrived to a contradiction, and hence $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C} \models p$, as required.

To complete the proof, it remains to be argued that a ground counterexample \mathcal{D}' for safety of exponential size can be verified in exponential time in the size of the original input. However, this is straightforward: first, checking $\mathcal{D}' \cup \mathcal{C} \not\models p$ amounts to checking that $\mathcal{D}' \cup \mathcal{D}_{\mathcal{C}}$ implies neither an extra answer to a CQ in \mathcal{C} nor p , and both checks are doable in exponential time; second, to check that $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$ we need to consider each of the exponentially many $\mathcal{D} \in \mathbb{D}$ and perform analogous checks for $\mathcal{D} \cup \mathcal{D}' \cup \mathcal{C}$. \square

Note that the data complexity of the algorithm suggested in the proof of Lemma 12 coincides with its combined complexity, namely CONEXPTIME. This is because the number of datasets in \mathbb{D} is exponential in the number of nulls in both $\mathcal{D}_{\mathcal{C}}$ and $f(\mathcal{D}_0)$. We next show that the algorithm suggested in the proof of Lemma 12 can be optimised to yield a Π_2^P upper bound on data complexity.

Lemma 13. SAFETY is in Π_2^P in data complexity.

Proof. First, note that in this case the policy is fixed, so the number of tuples to check for non-Boolean policies is polynomial. Therefore, in the rest of the proof we concentrate on Boolean policies as in Lemma 12.

Recall the proof of Lemma 12. We next argue that, although \mathbb{D} contains exponentially many datasets \mathcal{D} in the size of \mathcal{D}_0 , it is enough to consider only polynomially many different subsets of \mathcal{D}' that can be obtained by intersecting \mathcal{D}' with a homomorphic image of p or $q(\bar{c})$ over $\mathcal{D} \cup \mathcal{D}'$ as detailed in the proof of Lemma 12. As a result, the constructed counterexample dataset \mathcal{D}'' is of size only polynomial in \mathcal{D}_0 . This is enough to establish Π_2^P -membership: the algorithm first guesses such a \mathcal{D}'' as a counterexample for safety, and then verifies this by checking that, first, $\mathcal{D}'' \cup \mathcal{C} \not\models p$ and, second, $f(\mathcal{D}_0) \cup \mathcal{D}'' \cup \mathcal{C} \models p$; the first condition is verifiable in polynomial time, while checking the second can be done with the help of an NP oracle by nondeterministically trying all $\mathcal{D} \in \mathbb{D}$ and verifying that each of them implies p together with \mathcal{D}'' (the latter can be done in polynomial time, since all the CQs are fixed).

The key observation is that, for any $\mathcal{D} \in \mathbb{D}$, the homomorphism from $q(\bar{c})$ with $\bar{c} \notin \text{Ans}$, for $[q, \text{Ans}] \in \mathcal{C}$, or from p into $\mathcal{D} \cup \mathcal{D}'$ depends only on a part of \mathcal{D} of size bounded by the size of the CQ q or p , respectively, and the rest of \mathcal{D} can be arbitrary. Therefore, the subset of \mathcal{D}' obtained by intersecting \mathcal{D}' with the corresponding homomorphic image contains a witness not only for \mathcal{D} , but also for all other datasets in \mathbb{D} that map the nulls relevant to the homomorphism in the same way as \mathcal{D} . Moreover, the number of relevant nulls is bounded by the size of the largest CQ in the input, which we consider fixed in this lemma.

Formally, let us denote by n the number of nulls in $f(\mathcal{D}_0) \cup \mathcal{D}_C$, by m the number of constants in the answers Ans in \mathcal{C} , and by s the product of the maximal arity in the predicates of the schema and maximal number of atoms in a CQ in (\mathcal{D}_0, f, p) and \mathcal{C} (assuming that this maximal size is not greater than n , otherwise we can take $s = n$). Consider a dataset $\mathcal{D} \in \mathbb{D}$, the identifying homomorphism h from $f(\mathcal{D}_0) \cup \mathcal{D}_C$ to \mathcal{D} , and the witnessing homomorphism g from a CQ to $\mathcal{D} \cup \mathcal{D}'$ as in the proof of Lemma 12. Denote by $\text{Cover}(\mathcal{D})$ a set of nulls required by g —that is, a minimal set of nulls in $f(\mathcal{D}_0) \cup \mathcal{D}_C$ such that g is a homomorphism to $h(\bar{\mathcal{D}})$, where $\bar{\mathcal{D}}$ is a dataset obtained from $f(\mathcal{D}_0) \cup \mathcal{D}_C$ by removing all nulls not in the set and all atoms involving these nulls. Since $\text{Cover}(\mathcal{D})$ is minimal, it contains at most s nulls. If, for any two datasets \mathcal{D}_1 and \mathcal{D}_2 in \mathbb{D} , $\text{Cover}(\mathcal{D}_1) = \text{Cover}(\mathcal{D}_2)$ and the nulls in this set are sent in the same way by the associated homomorphisms, then the subset of \mathcal{D}' needed for witnessing CQ $q(\bar{c})$ or p in $\mathcal{D}_1 \cup \mathcal{D}'$ can play the same role for witnessing the CQ in $\mathcal{D}_2 \cup \mathcal{D}'$. So, if we go through all the datasets \mathcal{D} of \mathbb{D} in an arbitrary order picking an arbitrary homomorphism g satisfying the required properties each time and add the corresponding subset of \mathcal{D}' to \mathcal{D}'' only if there has not been any previously considered dataset with the same covering set as $\text{Cover}(\mathcal{D})$, then \mathcal{D}'' is still a required counterexample for safety.

Therefore, we are left to prove that such \mathcal{D}'' is of polynomial size, assuming s fixed. To this end, note that there are only polynomial number of elements \mathcal{D} in \mathbb{D} with different cover: indeed, there are at most $\binom{n+1}{s}$ possibilities for $\text{Cover}(\mathcal{D})$, and each of them has $(m+1)^s$ possibilities for the nulls (m constants in answers Ans and the fresh constant for each null)—that is, overall, the number is bounded by $(n+1)^s(m+1)^s$, which is a polynomial for a fixed s . So, in the search for \mathcal{D}'' , we can restrict ourselves to ground datasets with the number of atoms bounded by $s(n+1)^s(m+1)^s$. \square

We next turn our attention to the lower bounds, and start with the combined complexity for the open-world version of SAFETY. In the conference version of this paper it was claimed

that $\text{SAFETY}^{\text{ow}}$ is Π_2^p -hard (Cuenca Grau & Kostylev, 2016), thus providing a matching lower bound to the upper bound in Lemma 10. The proof was based on a simple reduction of the complement of CRITICALTUPLE —a well-known problem in database theory, which was claimed to be Σ_2^p -hard (Miklau & Suciu, 2007). Alas, the hardness proof of CRITICALTUPLE by Miklau and Suciu (2007) has a subtle problem, which does not seem to have an easy fix; for now, the best known lower bound for CRITICALTUPLE is NP, and its precise complexity remains open (Kostylev & Suciu, 2018). The following lemma shows that $\text{SAFETY}^{\text{ow}}$ is as hard as the complement of CRITICALTUPLE . Before stating the lemma, we provide a definition of CRITICALTUPLE (Miklau & Suciu, 2007).

A ground atom τ is *critical* for a Boolean CQ q if and only if there exists a ground dataset \mathcal{D} with a homomorphism from q to \mathcal{D} such that there is no homomorphism from q to $\mathcal{D} \setminus \{\tau\}$. CRITICAL-TUPLE is the problem of checking whether a ground atom is critical for a Boolean CQ.²

Lemma 14. *$\text{SAFETY}^{\text{ow}}$ is as hard as the complement of CRITICALTUPLE for ground PPDP instances, strict suppressors, and Boolean policies.*

Proof. Let τ, q be an input to CRITICALTUPLE . For the reduction, we take $\{\tau\}$ as dataset \mathcal{D}_0 , suppressor f that trivially maps each position to its value, and q as policy p . It is immediate to see that (\mathcal{D}_0, f, p) is safe if and only if τ is not critical for q . \square

As we mentioned, the best known lower bound for CRITICALTUPLE is NP, so Lemma 14 gives us a CONP lower bound for $\text{SAFETY}^{\text{ow}}$. However, the reduction in the proof of Lemma 14 relies on datasets consisting of single facts and trivial suppressors that do not introduce any nulls. The next lemma exploits the fact that the input to safety checking may involve datasets with many facts and non-trivial suppressors to provide a better DP lower bound for $\text{SAFETY}^{\text{ow}}$ based on a native reduction of a standard homomorphism problem. This bound, however, still does not match the Π_2^p upper bound from Lemma 10, so we leave the precise complexity of $\text{SAFETY}^{\text{ow}}$ open.

Lemma 15. *$\text{SAFETY}^{\text{ow}}$ is DP-hard for ground RDF PPDP instances, strict suppressors, and Boolean policies.*

Proof. The proof is by reduction of the standard $\text{HOMOMORPHISM-NOHOMOMORPHISM}$ problem, which is DP-complete. The input to this problem consists of four connected directed graphs, $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}'_1$ and \mathcal{G}'_2 , and the answer is yes if and only if there exists a homomorphism from \mathcal{G}_1 to \mathcal{G}_2 and there is no homomorphism from \mathcal{G}'_1 to \mathcal{G}'_2 .

Let $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}'_1$, and \mathcal{G}'_2 be an input to $\text{HOMOMORPHISM-NOHOMOMORPHISM}$. We construct a PPDP instance (\mathcal{D}_0, f, p) with a ground dataset \mathcal{D}_0 , a strict \mathcal{D}_0 -suppressor f , and a Boolean policy p , and then prove that the instance is safe if and only if the answer to $\text{HOMOMORPHISM-NOHOMOMORPHISM}$ on $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}'_1$, and \mathcal{G}'_2 is yes. The reduction uses a unary predicate U and two binary predicates Edge and Edge' (so the adaptation to the RDF case is as usual).

We start with the definition of \mathcal{D}_0 . Let it consist of an atom $\text{Edge}(c, c)$, for a constant c , and atoms $\text{Edge}'(c_u, c_v)$, for each edge (u, v) of \mathcal{G}'_2 , where c_u and c_v are constants associated

2. In (Miklau & Suciu, 2007) critical tuples are defined for arbitrary CQs and Σ_2^p -hardness is claimed for the general case; however, the CQ in the proof is Boolean.

with nodes u and v of \mathcal{G}_2 , respectively. Then, let f be the strict \mathcal{D}_0 -suppressor that sends each position over constant c_v of \mathcal{D}_0 , for each node v of \mathcal{G}_2 , to a null \mathbf{b}_v uniquely associated with v .

Next we define the policy p . Let it be the Boolean CQ that consists of the direct representations of

1. graph \mathcal{G}_1 over predicate **Edge** and fresh (existential) variables for all nodes of \mathcal{G}_1 ,
2. graph \mathcal{G}_2 over predicate **Edge** and fresh variables for all nodes of \mathcal{G}_2 , with each such variable x additionally in the atom $\mathbf{U}(x)$, and
3. graph \mathcal{G}'_1 over predicate **Edge'** and fresh variables for all nodes of \mathcal{G}'_1 .

We next show correctness of this reduction.

Assume first that (\mathcal{D}_0, f, p) is safe—that is, for every dataset \mathcal{D}' either there is a homomorphism from p to \mathcal{D}' or there is no homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. We need to prove that there is a homomorphism from \mathcal{G}_1 to \mathcal{G}_2 and there is no homomorphism from \mathcal{G}'_1 to \mathcal{G}'_2 . Consider first a dataset \mathcal{D}' that is the same as parts 2 and 3 of p as defined above, except that it has fresh constants instead of variables. By construction, there is a homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$, which sends all variables in part 1 of p to constant c and all other variables to their counter-parts in \mathcal{D}' . Therefore, since (\mathcal{D}_0, f, p) is safe, there is a homomorphism from p to \mathcal{D}' , which immediately implies that there is a homomorphism from \mathcal{G}_1 to \mathcal{G}_2 , as required. Second, consider \mathcal{D}' that is the same as parts 1 and 2 of p , again over fresh constants instead of variables. There is no homomorphism from p to \mathcal{D}' simply because \mathcal{D}' has no atoms over **Edge'**. Therefore, there is no homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. In particular, since we can send parts 1 and 2 to their copies, there is no homomorphism from part 3 to the part of $f(\mathcal{D}_0) \cup \mathcal{D}'$ over **Edge**—that is, there is no homomorphism from \mathcal{G}'_1 to \mathcal{G}'_2 , as required.

For the converse direction, assume that there is a homomorphism from \mathcal{G}_1 to \mathcal{G}_2 but no homomorphism from \mathcal{G}'_1 to \mathcal{G}'_2 . We need to prove that (\mathcal{D}_0, f, p) is safe. For the sake of contradiction, suppose that this is not the case—that is, there is a dataset \mathcal{D}' without a homomorphism from p to \mathcal{D}' , but with a homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. Anonymisation $f(\mathcal{D}_0)$ has no **U** atoms, so there exists a homomorphism from part 2 of p to \mathcal{D}' . Therefore, since there is a homomorphism from \mathcal{G}_1 to \mathcal{G}_2 , there is a homomorphism from part 1 of p to \mathcal{D}' as well. Finally, since there is no homomorphism from \mathcal{G}'_1 to \mathcal{G}'_2 , anonymisation $f(\mathcal{D}_0)$ has only nulls in the **Edge'** atoms, and \mathcal{G}'_1 is connected, there is a homomorphism from part 3 to \mathcal{D}' . To summarise, there is a homomorphism from all parts of p to \mathcal{D}' , which contradicts the fact that \mathcal{D}' is a witness for the instance (\mathcal{D}_0, f, p) being unsafe. So our assumption was wrong, and (\mathcal{D}_0, f, p) is safe, as required. \square

We next focus on linkage safety in the presence of closed-world information. We can prove a CONEXPTIME lower bound in combined complexity, which matches the upper bound in Lemma 12. For the intuition behind this lower bound, recall that in Lemma 12 we showed that any external dataset that is a counterexample for safety contains a part to witness a homomorphism from a CQ (either the policy or from a closure) for each minimal model of the anonymisation and the closures, each of which corresponds to a possible merging of the nulls of the anonymisation with constants in the answers of the closures. There are

exponentially many such minimal models, and our hardness proof shows that, in the worst case, the witnessing part in the external dataset must be unique for each such model.

Lemma 16. *SAFETY is CONEXPTIME-hard for ground RDF PPDP instances, strict suppressors, and Boolean policies.*

Proof. We show the NEXPTIME-hardness of the complement of SAFETY by a reduction of the exponential tiling problem EXPTILING. The input of EXPTILING is a *tiling instance* (\mathcal{T}, C^x, C^y) , with a finite set of tile types \mathcal{T} , horizontal compatibility relation $C^x \subseteq \mathcal{T} \times \mathcal{T}$, and vertical compatibility relation $C^y \subseteq \mathcal{T} \times \mathcal{T}$, while the answer is **true**—that is, the instance *has a solution*—if and only if it is possible to tile a $2^n \times 2^n$ square, for $n = |\mathcal{T}|$, according to C^x and C^y .

Let (\mathcal{T}, C^x, C^y) be a tiling instance with \mathcal{T} of size n . We construct a PPDP instance (\mathcal{D}_0, f, p) with a ground dataset \mathcal{D}_0 , a strict \mathcal{D}_0 -suppressor f and a Boolean policy p , and a set of closures \mathcal{C} such that the PPDP instance is safe with respect to the closures if and only if the tiling instance has a solution. In fact, it will be convenient to give first a reduction for the relaxed case when suppressors may not be strict, and discuss how to avoid this relaxation in the end of the proof. As usual, we use only unary and binary predicates in the reduction, so the adaptation to the RDF case is standard.

We start with the definition of the vocabulary: let it contain a unary predicate **Elem**, two binary predicates Bit_i^x and Bit_i^y for each $i = n, \dots, 1$, and two other binary predicates **TConst** and **TTile**. In the construction, it will be convenient to use the following abbreviations, for variables or constants $w, w_c, w_s, \bar{u} = u_n, \dots, u_1$, and $\bar{v} = v_n, \dots, v_1$:

$$\begin{aligned} \text{Coords}(w, \bar{u}, \bar{v}) &= \text{Elem}(w) \wedge \\ &\quad \text{Bit}_n^x(w, u_n) \wedge \dots \wedge \text{Bit}_1^x(w, u_1) \wedge \text{Bit}_n^y(w, v_n) \wedge \dots \wedge \text{Bit}_1^y(w, v_1), \\ \text{Tiled}(w, w_c, w_s, \bar{u}, \bar{v}) &= \text{TConst}(w, w_c) \wedge \text{TTile}(w, w_s) \wedge \\ &\quad \text{Bit}_n^x(w, u_n) \wedge \dots \wedge \text{Bit}_1^x(w, u_1) \wedge \text{Bit}_n^y(w, v_n) \wedge \dots \wedge \text{Bit}_1^y(w, v_1); \end{aligned}$$

as usual, if all the arguments are constants, we may look at these abbreviations as datasets rather than conjunctions.

Let \mathcal{D}_0 be $\text{Coords}(e, 0, \dots, 0)$, where e and 0 are constants. Consider also the \mathcal{D}_0 -suppressor f that sends each position of $\text{Coords}(e, 0, \dots, 0)$ (seen as a single atom) to a fresh null; we denote these nulls by $\mathbf{b}_e, \mathbf{b}_n^x, \dots, \mathbf{b}_1^x, \mathbf{b}_n^y, \dots, \mathbf{b}_1^y$ (i.e., \mathbf{b}_e is the argument of the **Elem** atom of $f(\mathcal{D}_0)$ and the first argument of all the **Bit** binary atoms, while the other nulls are the second arguments of the **Bit** atoms).

Next we define the policy p . Let it be the following Boolean CQ, where c is a fresh constant, and \bar{x} and \bar{y} are tuples of variables of size n :

$$\exists z_e, z_t, s, \bar{x}, \bar{y}. \text{Coords}(z_e, \bar{x}, \bar{y}) \wedge \text{Tiled}(z_t, c, s, \bar{x}, \bar{y}).$$

In the definition of the closures we use the following abbreviations, for tuples of variables $\bar{u} = u_n, \dots, u_1$ and $\bar{v} = v_n, \dots, v_1$, where 0 is the constant from \mathcal{D}_0 and 1 is a fresh constant:

$$\begin{aligned} \text{Next}_1(\bar{u}, \bar{v}) &= (u_n = v_n) \wedge \dots \wedge (u_2 = v_2) \wedge (u_1 = 0) \wedge (v_1 = 1), \\ &\dots \\ \text{Next}_i(\bar{u}, \bar{v}) &= (u_n = v_n) \wedge \dots \wedge (u_{i+1} = v_{i+1}) \wedge (u_i = 0) \wedge (v_i = 1) \wedge \\ &\quad (u_{i-1} = 1) \wedge (v_{i-1} = 0) \wedge \dots \wedge (u_1 = 1) \wedge (v_1 = 0), \\ &\dots \\ \text{Next}_n(\bar{u}, \bar{v}) &= (u_n = 0) \wedge (v_n = 1) \wedge (u_{n-1} = 1) \wedge (v_{n-1} = 0) \wedge \dots \wedge (u_1 = 1) \wedge (v_1 = 0). \end{aligned}$$

Intuitively, $\text{Next}_i(\bar{u}, \bar{v})$ is **true** whenever \bar{u} and \bar{v} represent consecutive numbers in binary representations of the form $b_n \dots b_{i+1} 0 1 \dots 1$ and of the form $b_n \dots b_{i+1} 1 0 \dots 0$, respectively, with $b_j \in \{0, 1\}$ for $n \leq j < i$.

Next, we define the set of closures \mathcal{C} . Let it consist of the following closures, for each $i = n, \dots, 1$, where the tile types in \mathcal{T} are used also as constants:

- $[q_i^{\text{x, Bit}}, \{0, 1\}]$, where $q_i^{\text{x, Bit}}(x) = \exists z. \text{Bit}_i^{\text{x}}(z, x)$;
- $[q_i^{\text{y, Bit}}, \{0, 1\}]$, where $q_i^{\text{y, Bit}}(y) = \exists z. \text{Bit}_i^{\text{y}}(z, y)$;
- $[q_i^{\text{x, Next}}, C^{\text{x}}]$, where

$$q_i^{\text{x, Next}}(s_1, s_2) = \exists z_1, z_2, w_c, \bar{x}_1, \bar{x}_2, \bar{y}. \text{Next}_i(\bar{x}_1, \bar{x}_2) \wedge \text{Tiled}(z_1, w_c, s_1, \bar{x}_1, \bar{y}) \wedge \text{Tiled}(z_2, w_c, s_2, \bar{x}_2, \bar{y});$$
- $[q_i^{\text{y, Next}}, C^{\text{y}}]$, where

$$q_i^{\text{y, Next}}(s_1, s_2) = \exists z_1, z_2, w_c, \bar{x}, \bar{y}_1, \bar{y}_2. \text{Next}_i(\bar{y}_1, \bar{y}_2) \wedge \text{Tiled}(z_1, w_c, s_1, \bar{x}, \bar{y}_1) \wedge \text{Tiled}(z_2, w_c, s_2, \bar{x}, \bar{y}_2).$$

Note that some of the queries in this definition use equality atoms in the *Next* abbreviations; this is done just for convenience, and this atoms can be easily eliminated in the usual way, so the resulting queries are usual CQs, as required by the definitions.

Having the construction completed, next we show the correctness of the reduction—that is, we prove that $(\mathcal{T}, C^{\text{x}}, C^{\text{y}})$ has a solution if and only if (\mathcal{D}_0, f, p) is not safe with respect to closures \mathcal{C} . Intuitively, the minimal models of $f(\mathcal{D}_0) \cup \mathcal{C}$ correspond to all possible re-identifications of nulls $\mathbf{b}_n^{\text{x}}, \dots, \mathbf{b}_1^{\text{x}}, \mathbf{b}_n^{\text{y}}, \dots, \mathbf{b}_1^{\text{y}}$ in \mathcal{D}_0 to 0 and 1, and any external dataset \mathcal{D}' witnessing non-safety should have *Tiled* atoms with the last $2n$ arguments being the values of each of these re-identifications. Then, we can see these $2n$ arguments as binary representations of the coordinates in the tiling square, and, by the last two parts of \mathcal{C} , the third arguments of *Tiled* should be tile types that agree with the compatibility relations. Note that the first argument of *Tiled* is used as a representative for the cell in the square with the coordinates, while the second argument plays a technical role: to witness each pair in the answers of the last two parts of \mathcal{C} without affecting the tiling solution, this argument may be any constant different from c , which appears in the policy.

We formally prove this intuition correct by means of the following two claims.

Claim 1. *If $(\mathcal{T}, C^{\text{x}}, C^{\text{y}})$ has a solution, then (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} .*

Proof. Assume that $(\mathcal{T}, C^{\text{x}}, C^{\text{y}})$ has a solution—that is, it is possible to tile a $2^n \times 2^n$ square according to C^{x} and C^{y} . We need to show that (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} —that is, there exists a dataset \mathcal{D}' such that $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. We next construct such a dataset. In particular, let \mathcal{D}' be the ground dataset consisting of the sets of atoms

- $\text{Tiled}(d, c, t, \bar{b}^{\text{x}}, \bar{b}^{\text{y}})$, for each pair of bit vectors \bar{b}^{x} and \bar{b}^{y} both of size n , for a fresh constant d uniquely associated with this pair of vectors, for constant c from policy p , and for the tile type t in the position with coordinates $\bar{b}^{\text{x}}, \bar{b}^{\text{y}}$ in the tiling solution;

- $Tiled(d_x, c'_x, t_1, \bar{b}_{2^i-1}, \bar{0})$ and $Tiled(d'_x, c'_x, t_2, \bar{b}_{2^i}, \bar{0})$ for each pair $(t_1, t_2) \in C^x$, for each $i = 1, \dots, n$, for fresh constants d_x, d'_x , and c'_x , uniquely associated with (t_1, t_2) and i , for the bit vectors b_{2^i-1} and b_{2^i} of length n each, representing numbers $2^i - 1$ and 2^i , respectively, in binary, and for the vector $\bar{0}$ of 0's also of length n ; and
- $Tiled(d_y, c'_y, t_1, \bar{0}, \bar{b}_{2^i-1})$ and $Tiled(d'_y, c'_y, t_2, \bar{0}, \bar{b}_{2^i})$ for each pair $(t_1, t_2) \in C^y$, for each $i = 1, \dots, n$, for fresh constants d_y, d'_y , and c'_y , uniquely associated with (t_1, t_2) and i , and for the bit vectors b_{2^i-1}, b_{2^i} and $\bar{0}$ as before.

We first note that $\mathcal{D}' \cup \mathcal{C} \not\models p$. Indeed, the Herbrand model of \mathcal{D}' —that is the model that interprets all constants by themselves and all predicates exactly according to the atoms in \mathcal{D}' —is a model of $\mathcal{D}' \cup \mathcal{C}$, but has no Elem atom, so it is not a model of p . Note also that the two last items in the construction of \mathcal{D}' are needed to satisfy the last two items in the construction of \mathcal{C} ; this is not guaranteed by the first item of \mathcal{D}' , because some pairs in C^x and C^y may not appear in the tiling solution in each row and column, respectively. This is why the third argument of $Tiled$ is introduced.

To complete the proof of the claim, we need to show that $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. To this end, consider any model I of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$. Since I is a model of $f(\mathcal{D}_0) \cup \mathcal{C}$, there are an element e' and bit vectors \bar{b}^x and \bar{b}^y such that I is a model of $\text{Coords}(e', \bar{b}^x, \bar{b}^y)$. Then, the mapping that sends z_e to e' , z_t to d associated with \bar{b}^x and \bar{b}^y , \bar{x} to \bar{b}^x , \bar{y} to \bar{b}^y , and s to the tile type in the tiling solution at coordinates \bar{b}^x, \bar{b}^y is a homomorphism from p to I . So, $I \models p$, as required. \square

Claim 2. *If (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} , then (\mathcal{T}, C^x, C^y) has a solution.*

Proof. Assume that (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} —that is, there exists a dataset \mathcal{D}' such that $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. We need to show that (\mathcal{T}, C^x, C^y) has a solution—that is, it is possible to tile a $2^n \times 2^n$ square according to C^x and C^y .

First, note that by Lemma 11 we can assume that \mathcal{D}' is ground.

Next we prove that, for any pair \bar{b}^x, \bar{b}^y of bit vectors of size n each, there exist constants d and t such that dataset \mathcal{D}' contains atoms $Tiled(d, c, t, \bar{b}^x, \bar{b}^y)$. To this end, consider any such pair \bar{b}^x, \bar{b}^y and the dataset \mathcal{D}'' extending \mathcal{D}' with $\text{Coords}(e', \bar{b}^x, \bar{b}^y)$, for a fresh constant e' , and, as in the proof of Claim 1, with

- $Tiled(d_x, c'_x, t_1, \bar{b}_{2^i-1}, \bar{0})$ and $Tiled(d'_x, c'_x, t_2, \bar{b}_{2^i}, \bar{0})$ for each pair $(t_1, t_2) \in C^x$, for each $i = 1, \dots, n$, for fresh constants d_x, d'_x , and c'_x , uniquely associated with (t_1, t_2) and i , and for the bit vectors b_{2^i-1}, b_{2^i} and $\bar{0}$ as in the proof of Claim 1; and
- $Tiled(d_y, c'_y, t_1, \bar{0}, \bar{b}_{2^i-1})$ and $Tiled(d'_y, c'_y, t_2, \bar{0}, \bar{b}_{2^i})$ for each pair $(t_1, t_2) \in C^y$, for each $i = 1, \dots, n$, for fresh constants d_y, d'_y , and c'_y , uniquely associated with (t_1, t_2) and i , and for the bit vectors b_{2^i-1}, b_{2^i} and $\bar{0}$ as in the proof of Claim 1.

By construction, the Herbrand model I of \mathcal{D}'' is a model of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$: it is a model of $f(\mathcal{D}_0)$ by the presence of $\text{Coords}(e', \bar{b}^x, \bar{b}^y)$, it is a model of \mathcal{D}' because \mathcal{D}'' includes \mathcal{D}' , and it is a model of \mathcal{C} because $\mathcal{D}' \cup \mathcal{C}$ does not model p and hence is satisfiable. Therefore, \mathcal{D}'' is also a model of p by assumption—that is, there is a homomorphism from p to I . This homomorphism sends conjunction $Tiled(z_t, c, s, \bar{x}, \bar{y})$ of p to (the interpretations of) atoms in \mathcal{D}' , because the rest of \mathcal{D}'' does not contain any TTile atoms with c as the second argument

and all TTile atoms not in \mathcal{D}' have fresh constants as the first argument. By construction, there are two options for conjunction $\text{Coords}(z_e, \bar{x}, \bar{y})$: either $\text{Coords}(e', \bar{b}^x, \bar{b}^y)$ or atoms in \mathcal{D}' (a mixture of these two options are not possible, because all atoms in $\text{Coords}(z_e, \bar{x}, \bar{y})$ mention z_e , and e' is a fresh constant). However, the second option is not possible, because this would imply that $\mathcal{D}' \models p$ and hence $\mathcal{D}' \cup \mathcal{C} \models p$, which is not allowed by assumption. Therefore, the homomorphism sends \bar{x} to \bar{b}^x and \bar{y} to \bar{b}^y , and, since I is a model of p , there exists t such that \mathcal{D}' contains atom $\text{Tiled}(d, c, t, \bar{b}^x, \bar{b}^y)$, as required.

Since for any \bar{b}^x, \bar{b}^y there exist d and t with $\text{Tiled}(d, c, t, \bar{b}^x, \bar{b}^y)$ in \mathcal{D}' , we can consider an assignment of the $2^n \times 2^n$ square with constants defined by these atoms: position with coordinates \bar{b}^x, \bar{b}^y is assigned with t means that \mathcal{D}' contains $\text{Tiled}(d, c, t, \bar{b}^x, \bar{b}^y)$ (if for some \bar{b}^x, \bar{b}^y there are more than one such t , we pick any of them). Moreover, since $\mathcal{D}' \cup \mathcal{C}$ is consistent (due to the fact that $\mathcal{D}' \cup \mathcal{C} \not\models p$), the assignment is a tiling solution—that is, each t belongs to \mathcal{T} and adjacent pairs of tiles agree with C^x and C^y , as required. \square

These two claims imply the statement of the theorem for suppressors that are not necessarily strict. Indeed, suppressor f in the construction anonymises the last $2n$ positions in $\text{Coords}(e, 0, \dots, 0)$ to different nulls. However, it is not difficult to adapt the reduction to the case when only strict suppressors are allowed: we just need to use not a single pair of constants 0 and 1, but $2n$ copies of this pair, one for each bit position in the coordinates; this would work because every time 0 or 1 is used in the construction, we know precisely which position it corresponds to. \square

It is worth mentioning at this point that in the conference version of this paper it was claimed that SAFETY is Π_3^p -complete in combined complexity (Cuenca Grau & Kostylev, 2016). Please note that this does not contradict the CONEXPTIME lower bound from Lemma 16: in the conference version we only allowed singleton closure sets, whereas the reduction in Lemma 16 uses several closures. The Π_3^p algorithm in (Cuenca Grau & Kostylev, 2016) is, however, not correct, so we leave open the precise complexity of SAFETY in the particular case where the input closure set is a singleton. It will be also interesting to understand the complexity of SAFETY when the closures are fixed, atomic or existential-free, or any combination of these.

We now turn our attention to the data complexity of SAFETY and provide a matching lower bound to the Π_2^p upper bound from Lemma 13.

Lemma 17. *SAFETY is Π_2^p -hard in data complexity for ground RDF PPDP instances, strict suppressors, Boolean policies, and existential-free CQs in closures.*

Proof. We show Π_2^p -hardness by reduction of the $\forall\exists 3\text{SAT}$ problem, whose input is a formula $\phi = \forall \bar{u}. \exists \bar{v}. \psi$ with ψ a formula in 3CNF over propositional variables \bar{u} and \bar{v} , and the answer is yes if and only if ϕ is valid. Similarly to $\exists\forall 3\text{SAT}$ problem, which is complete for Σ_2^p , $\forall\exists 3\text{SAT}$ is the canonical complete problem for Π_2^p .

Let $\phi = \forall \bar{u}. \exists \bar{v}. \psi$. We construct a PPDP instance (\mathcal{D}_0, f, p) with a ground dataset \mathcal{D}_0 , a strict \mathcal{D}_0 -suppressor f and a Boolean policy p , and a set of closures \mathcal{C} with existential-free CQs such that p and the CQs in \mathcal{C} do not depend on ϕ , and such that the PPDP instance is safe with respect to \mathcal{C} if and only if ϕ is valid. As usual, the arities of the predicates involved in the reduction are bounded by 2, so the adaptation to RDF is straightforward.

In the first step of the reduction, we transform ϕ to an equivalent $\phi' = \forall \bar{u}'. \exists \bar{v}'. \psi'$, where ψ' is a formula in 6CNF such that its each clause has precisely 3 literals over variables in \bar{u}' and 3 literals over variables in \bar{v}' . We do it by rewriting each clause γ in ψ as follows:

- if all literals in γ are over variables in \bar{u} , then rewrite γ to the two clauses $\gamma \vee v \vee v \vee v$ and $\gamma \vee \neg v \vee \neg v \vee \neg v$, for a fresh propositional variable v added to \bar{v} ;
- if all literals in γ are over variables in \bar{v} , then rewrite γ to the clause $\gamma \vee u \vee u \vee u$, for a fresh propositional variable u added to \bar{u} ;
- otherwise, we archive the required number of literals over \bar{u} and \bar{v} by duplicating the existing literals in the straightforward way.

In the result we assume that \bar{u}' and \bar{v}' are the extended \bar{u} and \bar{v} , respectively.

We start the description of the reduction with the definition of the vocabulary: let it contain a unary predicate U and binary predicates V , $U\text{Values}$, $V\text{Values}$, CF_1^u , \dots , CF_3^u , CF_1^{bv} , CF_1^{cv} , \dots , CF_3^{bv} , CF_3^{cv} (as required, the predicates in the vocabulary do not depend on ϕ and are at most binary). Similar to the proof of Lemma 16, we will use the following abbreviation, for variables and constants t , t_1^u , \dots , t_3^u , t_1^{bv} , t_1^{cv} , \dots , t_3^{bv} , t_3^{cv} :

$$\begin{aligned} \text{ClauseFalsification}(t, t_1^u, \dots, t_3^u, t_1^{bv}, t_1^{cv}, \dots, t_3^{bv}, t_3^{cv}) = & CF_1^u(t, t_1^u) \wedge \dots \wedge CF_3^u(t, t_3^u) \wedge \\ & CF_1^{bv}(t, t_1^{bv}) \wedge CF_1^{cv}(t, t_1^{cv}) \wedge \dots \wedge CF_3^{bv}(t, t_3^{bv}) \wedge CF_3^{cv}(t, t_3^{cv}). \end{aligned}$$

Next we define the dataset \mathcal{D}_0 and the \mathcal{D}_0 -suppressor f . Let \mathcal{D}_0 be the ground dataset that uses constants c_γ for each clause γ in ψ' , constants c_w^{false} and c_w^{true} for each $w \in \bar{u}' \cup \bar{v}'$, and constants c_v and d_v^{false} for each $v \in \bar{v}'$ (we will see that in closures \mathcal{C} constants d_v^{true} are also used). Then, let \mathcal{D}_0 consist of

- the atom $V\text{Values}(d_v^{\text{false}}, c_v)$ for each $v \in \bar{v}'$; and
- the atoms $\text{ClauseFalsification}(c_\gamma, a_{u_1}, \dots, a_{u_3}, d_{v_1}^{\text{false}}, a_{v_1}, \dots, d_{v_3}^{\text{false}}, a_{v_3})$ for each clause γ in ψ' with the literals over variables $u_1, \dots, u_3 \in \bar{u}'$ and $v_1, \dots, v_3 \in \bar{v}'$, where, for each of these variables w , a_w is constant c_w^{false} if the literal with w is positive in γ and constant c_w^{true} otherwise (essentially, this conjunction represents the falsifying assignment of the clause).

Let also f be the strict \mathcal{D}_0 -suppressor that anonymises, for each clause γ in ψ' , constant c_γ to a null b_γ uniquely associated with γ , and, for each variable $v \in \bar{v}'$, constant d_v^{false} to a null b_v uniquely associated with v (note that the choice of d_v^{false} in the atoms of \mathcal{D}_0 is arbitrary, and it could be d_v^{true}).

Let then the policy p be the Boolean CQ

$$\begin{aligned} \exists z, x_1, \dots, x_3, y_1^{bv}, y_1^{cv}, \dots, y_3^{bv}, y_3^{cv}. & \text{ClauseFalsification}(z, x_1, \dots, x_3, y_1^{bv}, y_1^{cv}, \dots, y_3^{bv}, y_3^{cv}) \wedge \\ & U(x_1) \wedge \dots \wedge U(x_3) \wedge V(y_1^{bv}, y_1^{cv}) \wedge \dots \wedge V(y_3^{bv}, y_3^{cv}), \end{aligned}$$

which also does not depend on ϕ .

To complete the construction, we next define the set of closures \mathcal{C} . Let it consist of the following pairs, where all CQs are existential-free and do not depend on ϕ :

- $[\text{UValues}(x_1, x_2), \{(c_u^{\text{false}}, c_u^{\text{true}}) \mid u \in \bar{u}'\}]$;
- $[\text{UValues}(x_1, x_2) \wedge \text{U}(x_1) \wedge \text{U}(x_2), \emptyset]$;
- $[\text{VValues}(x, y), \{(d_v^{\text{false}}, c_v), (d_v^{\text{true}}, c_v) \mid v \in \bar{v}'\}]$; and
- $[\text{V}(x, y), \{(d_v^{\text{false}}, c_v^{\text{false}}), (d_v^{\text{true}}, c_v^{\text{true}}) \mid v \in \bar{v}'\}]$.

Having the construction completed, next we prove its correctness—that is, show that ϕ' is not valid if and only if the PPDP instance is unsafe with respect the closures. Intuitively, each assignment to \bar{u}' corresponds to a candidate external dataset \mathcal{D}' witnessing non-safety with one of $\text{U}(c_u^{\text{false}})$ and $\text{U}(c_u^{\text{true}})$ for each $u \in \bar{u}'$, while the gadget on the constants and nulls corresponding to the \bar{v}' variables guarantees that p matches a model of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$ if and only if there is a falsified clause in ψ' . Next we prove this intuition correct formally.

Assume that ϕ' is not valid—that is, there exists an assignment σ of \bar{u}' such that for every extension of σ to \bar{v}' there is a clause in ψ' that evaluates to **false** under the overall assignment. We show that (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} . To this end, consider the ground dataset \mathcal{D}' that consists of atoms $\text{U}(c_u^{\text{false}})$ for each $u \in \bar{u}'$ such that $\sigma(u) = \text{false}$ and atoms $\text{U}(c_u^{\text{true}})$ for each $u \in \bar{u}'$ such that $\sigma(u) = \text{true}$. We have that $\mathcal{D}' \cup \mathcal{C} \not\models p$, because $\mathcal{D}' \cup \mathcal{C}$ is satisfiable and does not mention atoms over the CF predicates, which are required by p . We also claim that $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. Indeed, any model I of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$ witnesses each \mathbf{b}_v by either d_v^{false} or d_v^{true} , which corresponds to an extension of assignment σ to \bar{v}' . Therefore, by assumption, there is a clause in ψ' that evaluates to **false** under the extended σ . By construction, this precisely means that the interpretation of *ClauseFalsification* in I contains a tuple with the second, third and fourth arguments in the interpretation of U and the fifth, seventh and ninth arguments connected by V predicate to the sixth, eighth and tenth, respectively—that is, p holds in I , as required.

Assume now that (\mathcal{D}_0, f, p) is not safe with respect to \mathcal{C} —that is, there exists a dataset \mathcal{D}' such that $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. We need to prove that ϕ' is not valid. By Lemma 11 we may assume that \mathcal{D}' is ground. Since $\mathcal{D}' \cup \mathcal{C} \not\models p$, $\mathcal{D}' \cup \mathcal{C}$ is satisfiable, and, therefore, there is no $u \in \bar{u}'$ such that both $\text{U}(c_u^{\text{false}})$ and $\text{U}(c_u^{\text{true}})$ are in \mathcal{D}' . Consider the assignment σ of \bar{u}' such that, for each u , $\sigma(u)$ is **false** if $\text{U}(c_u^{\text{false}}) \in \mathcal{D}'$ and **true** otherwise (including the case when none of $\text{U}(c_u^{\text{false}})$ and $\text{U}(c_u^{\text{true}})$ are in \mathcal{D}'). We claim that ψ' evaluates to **false** under any extension of σ to \bar{v}' . To this end, consider any such extension and the Herbrand model I of \mathcal{D}' extended by all the V , UValues and VValues atoms enforced by \mathcal{C} , and by all *ClauseFalsification* atoms in $f(\mathcal{D}_0)$ with each \mathbf{b}_γ replaced by a fresh constant c'_γ and each \mathbf{b}_v replaced by either d_v^{false} , if $\sigma(v) = \text{false}$, or by d_v^{true} , if $\sigma(v) = \text{true}$. By construction, $I \models f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$. Therefore, by assumption, $I \models p$ —that is, there is a homomorphism from p to I . The images of the U atoms of p under this homomorphism correspond to atoms in \mathcal{D}' by construction. Therefore, since the c'_γ were fresh and are not mentioned in \mathcal{D}' , the images of all the CF atoms in the *ClauseFalsification* conjunction, which all have a common variable z , cannot correspond to atoms in \mathcal{D}' , because otherwise we have $\mathcal{D}' \cup \mathcal{C} \models p$, which is not allowed by assumption. So, the *ClauseFalsification* atoms are sent to the interpretations of atoms in $f(\mathcal{D}_0)$ with nulls replaced as described above, which means that there exists a clause in ψ' falsified under the extended σ . Since the choice of the extension of σ to \bar{v}' was arbitrary, the original σ assigning \bar{u}' is a witness for the fact that ϕ' is not valid. \square

In the conference version of this paper it was claimed that SAFETY is NP-complete in data complexity (Cuenca Grau & Kostylev, 2016). Although this result does not contradict the Π_2^p lower bound from Lemma 17 (as it requires singleton closure sets), the NP algorithm provided in our conference paper is incorrect, and we leave open the precise data complexity of SAFETY for singleton closure sets. It would also be interesting to understand the data complexity of SAFETY when the closures are atomic.

The following theorem summarises our results of Lemmas 10, 12, 13, 14, 15, 16, and 17.

Theorem 3. *The following results hold:*

1. $\text{SAFETY}^{\text{ow}}$ is in Π_2^p , as hard as the complement of CRITICALTUPLE, and DP-hard in combined complexity. It is in AC^0 in data complexity; both of the combined complexity lower bounds hold already for ground PPDP instances, strict suppressors, and Boolean policies, while the DP lower bound holds already if the instances are additionally required to be RDF; and
2. SAFETY is CONEXPTIME-complete in combined complexity and Π_2^p -complete in data complexity. Both of the lower bounds hold already for ground RDF PPDP instances, strict suppressors, and Boolean policies, while the Π_2^p lower bound holds already if the closure CQs are additionally required to be existential-free.

5.2 Cost Minimisation for Linkage Safety

In this section we study the cost minimisation problems MIN-SAFETY and MIN-SAFETY_s associated to linkage safety. In particular, we show that

- the open-world versions of the problems are both in Σ_3^p and Σ_2^p -hard in combined complexity, and NP-complete in data complexity; and
- the general versions of the problems are both CONEXPTIME-complete in combined complexity and Σ_3^p -complete in data complexity.

As for the previous problems, the lower bounds hold also under various restrictions on the datasets, suppressors, policies, and closures, which we discuss in each particular case.

We start with the upper bounds. In all cases, the algorithm just tries all possible suppressors within the cost limit and uses the corresponding algorithm designed in Section 5.1 to check each of these suppressors for safety. Hence, the upper bounds are obtained by simple guess-and-check algorithms using the algorithms for SAFETY as “black boxes”.

Lemma 18. *The following results hold:*

1. MIN-SAFETY^{ow} and MIN-SAFETY_s^{ow} are in Σ_3^p and in NP in data complexity;
2. MIN-SAFETY and MIN-SAFETY_s are in CONEXPTIME and in Σ_3^p in data complexity.

Proof. As discussed in Section 3.6 and Lemma 5, the sizes of all possible \mathcal{D}_0 -suppressors are linearly bounded by the size of \mathcal{D}_0 . So, on input \mathcal{D}_0 , p , \mathcal{C} (when relevant) and ℓ , our algorithms (except for the one used to establish the CONEXPTIME bound) work by first guessing a \mathcal{D}_0 -suppressor f with cost at most ℓ and then calling for the corresponding oracle

to check safety of the resulting PPDP instance (\mathcal{D}_0, f, p) with respect to \mathcal{C} , the complexity of which is devised either in Lemma 10 or in Lemma 13. In the most general, CONEXPTIME case, we do not need to guess a suppressor, because we can just check all of them one by one in CONEXPTIME for safety using the algorithm from Lemma 12. \square

The lower bounds, which we establish next, are generally much more difficult to prove. We start with the combined complexity of the open-world versions of the problems. As discussed in the previous section, we do not have a matching lower bound for SAFETY^{ow}. As a result, it is not surprising that our Σ_2^p lower bounds for MIN-SAFETY^{ow} and MIN-SAFETY_s^{ow}, obtained in the next two lemmas, also do not match our Σ_3^p upper bounds from Lemma 18. However, we anticipate that, on the one hand, if CRITICALTUPLE turns out to be Σ_2^p -hard, then it should be possible to use the ideas in our proof to show Σ_3^p -hardness of MIN-SAFETY^{ow} and MIN-SAFETY_s^{ow}, and, on the other hand, if it were to be in NP, we should be able to use the corresponding NP algorithm for CRITICALTUPLE to obtain a Σ_2^p algorithm for our problems.

To establish the Σ_2^p lower bound of MIN-SAFETY^{ow} in the following lemma we reuse once again the construction from the proof of statement 1 of Lemma 6. Note that it is essential here that the PPDP instances are not required to be ground. We leave open the question whether the problem is still Σ_2^p -hard for ground instances.

Lemma 19. *MIN-SAFETY^{ow} is Σ_2^p -hard for RDF PPDP instances and Boolean policies.*

Proof. As in case of statement 1 of Lemma 7, the proof of this lemma is based on that of statement 1 of Lemma 6. In fact, the only difference in the construction is that this time we use only nulls in the dataset \mathcal{D}_0 instead of constants. As the result, any external dataset \mathcal{D}' that does not imply the policy does not influence the answer to the policy on $f(\mathcal{D}_0) \cup \mathcal{D}'$, whatever is a suppressor f , because \mathcal{D}' cannot mention the nulls of \mathcal{D}_0 , and \mathcal{D}_0 is connected. So, in this case, safety boils down to compliance, which is Σ_2^p -hard by reduction in the proof of Lemma 6. \square

As established by the following lemma, the strict version of the problem, MIN-SAFETY_s^{ow}, is also Σ_2^p -hard. Unfortunately, we cannot reuse the proof of Lemma 6 as we just did for MIN-SAFETY^{ow}: as mentioned before Lemma 7, strict suppressors cannot change the value of constant-free Boolean queries. As a result, we had to devise a direct reduction of $\exists\forall 3\text{SAT}$. Such reduction works for ground instances, so the result is somewhat stronger than that in Lemma 19 for MIN-SAFETY^{ow}. Also, the reduction uses only Boolean policies, which is interesting when comparing it with the reduction we used in the proof of statement 1 of Lemma 7 to show Σ_2^p -hardness of MIN-COMPLIANCE_s^{ow}.

Lemma 20. *MIN-SAFETY_s^{ow} is Σ_2^p -hard for ground RDF PPDP instances and Boolean policies.*

Proof. We show Σ_2^p -hardness of MIN-SAFETY^{ow} by reduction of $\exists\forall 3\text{SAT}$. The reduction is very similar to the ones in the proofs of statements 1 of Lemmas 6 and 7.

Let $\phi = \exists \bar{r}. \forall \bar{v}. \neg \psi$ be a formula with ψ in 3CNF over propositional variables \bar{r} and \bar{v} . We construct a ground dataset \mathcal{D}_0 , a Boolean policy p , and an integer ℓ such that there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ for which the instance (\mathcal{D}_0, f, p) is safe if

and only if ϕ is valid. As usual, we use only unary and binary predicates in the reduction, so the adaptation to the RDF case is straightforward.

First, let the predicates in the vocabulary be the same as in the proof of statement 1 of Lemma 6—that is, a unary predicate R and binary predicates Arg_1 , Arg_2 , Arg_3 , $\text{Back}_1^{\text{false}}$, $\text{Back}_1^{\text{true}}$, and Back_2 —plus a unary predicate Anon and a binary predicate P .

Next we define dataset \mathcal{D}_0 in a similar way as we defined \mathcal{D}'_0 in Lemma 6: the only difference is that it has no atoms $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$, for $r \in \bar{r}$, but has several atoms $P(c_r^{\text{false}}, d)$ and $P(c_r^{\text{true}}, d)$ as in the proof of Lemma 7, which are necessary to make the costs of the anonymisation of all c_r^t the same, and has atoms $\text{Anon}(c)$ for each constant c different from all c_r^{false} , all c_r^{true} and all d in the P atoms. In particular, let \mathcal{D}_0 consist of the following facts, where we use constants c^{false} , c^{true} , c^y , c^z , constants c_γ^π for every clause γ and each (of at most 7) assignment π of the variables of γ that satisfies γ , constants c_w^{false} and c_w^{true} for each $w \in \bar{r} \cup \bar{v}$, constants d_r^y for every $r \in \bar{r}$, constants d^z , d^{cl} , d^{var} , and constants d_i , for $i = 1, \dots, 3n + 1$, where n is the number of clauses in ψ :

- $R(c^{\text{false}})$ and $R(c^{\text{true}})$;
- $\text{Back}_1^{\text{false}}(c^{\text{false}}, c^y)$, $\text{Back}_1^{\text{true}}(c^{\text{true}}, c^y)$, and $\text{Back}_2(c^y, c^z)$;
- $\text{Anon}(c^{\text{false}})$, $\text{Anon}(c^{\text{true}})$, $\text{Anon}(c^y)$, and $\text{Anon}(c^z)$;
- $\text{Cl}_\gamma(c^z, c_\gamma^\pi)$, $\text{Arg}_1(c_\gamma^\pi, c_{w_1}^{\pi(w_1)})$, \dots , $\text{Arg}_3(c_\gamma^\pi, c_{w_3}^{\pi(w_3)})$, and $\text{Anon}(c_\gamma^\pi)$, for each clause γ in ψ over variables w_1, \dots, w_3 and each assignment π of w_1, \dots, w_3 satisfying γ ;
- $\text{Anon}(c_v^{\text{false}})$ and $\text{Anon}(c_v^{\text{true}})$, for each $v \in \bar{v}$;
- $\text{Back}_1^{\text{false}}(c_r^{\text{false}}, d_r^y)$, $\text{Back}_1^{\text{true}}(c_r^{\text{true}}, d_r^y)$, $\text{Back}_2(d_r^y, d^z)$, and $\text{Anon}(d_r^y)$, for each $r \in \bar{r}$;
- $P(c_r^t, d_i)$, for each $r \in \bar{r}$, each $t \in \{\text{false}, \text{true}\}$, and each $i = m + 1, \dots, 3n + 1$, where m is the number of atoms with c_r^t above;
- $\text{Cl}_\gamma(d^z, d^{\text{cl}})$, for each clause γ in ψ ; and
- $\text{Arg}_1(d^{\text{cl}}, d^{\text{var}}) \dots \text{Arg}_3(d^{\text{cl}}, d^{\text{var}})$, $R(d^{\text{var}})$, $\text{Anon}(d^z)$, $\text{Anon}(d^{\text{cl}})$, and $\text{Anon}(d^{\text{var}})$.

Next, let ℓ be the cost of the strict \mathcal{D}_0 -suppressor that anonymises all constants of \mathcal{D}_0 except for all c_r^{false} , for $r \in \bar{r}$, and all d_i , for $i = 1, \dots, 3n + 1$.

Finally, the policy p is the same as the one in Lemma 6, except that it additionally contains an atom $\text{Anon}(x)$ for each variable x different from all x_u , for $u \in \bar{u}$. In particular, it is the Boolean CQ consisting of the following atoms, where all arguments are existential variables:

- $R(x^{\text{false}})$ and $R(x^{\text{true}})$,
- $\text{Back}_1^{\text{false}}(x^{\text{false}}, y)$, $\text{Back}_1^{\text{true}}(x^{\text{true}}, y)$, and $\text{Back}_2(y, z)$,
- $\text{Anon}(x^{\text{false}})$, $\text{Anon}(x^{\text{true}})$, $\text{Anon}(y)$, and $\text{Anon}(z)$,
- $\text{Cl}_\gamma(z, x_\gamma)$, $\text{Arg}_1(x_\gamma, x_{w_1})$, \dots , $\text{Arg}_3(x_\gamma, x_{w_3})$, and $\text{Anon}(x_\gamma)$, for each clause γ in ψ over propositional variables w_1, \dots, w_3 ,

- $R(x_r)$, for each $r \in \bar{r}$, and
- $\text{Anon}(x_v)$, for each $v \in \bar{v}$.

We next show that ϕ is valid if and only if there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ such that (\mathcal{D}_0, f, p) is safe. The intuition is as follows. A strict suppressor f that does not anonymise a constant different from c_r^{false} , c_r^{true} and d_i cannot be safe, because these constants are in **Anon**, and a witnessing external dataset may consist of a copy of p without the corresponding **Anon** atom. Also, same as in the proof of Lemma 6, a safe f should anonymise at least one of c_u^{false} and c_u^{true} for each u , but it cannot anonymise both, because of the cost limit ℓ . So, the external dataset \mathcal{D}' can contain one and only one of $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$, which defines an assignment to \bar{r} variables. Then, same as Lemma 6, a homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$ corresponds precisely to an extension of the assignment that satisfies ψ . Next we formally prove this intuition correct.

For the forward direction, let ϕ be valid—that is, there exists a truth assignment σ of \bar{r} such that for its any extension to \bar{v} the resulting assignment turns ψ to **false**. Consider the strict \mathcal{D}_0 -suppressor f that anonymises all constants c in \mathcal{D}_0 except for all $c_r^{\sigma(r)}$, for $r \in \bar{r}$, and all d_i , for $i = 1, \dots, 3n + 1$. By definition, the cost of f is precisely ℓ . We claim that (\mathcal{D}_0, f, p) is safe—that is, there is no external dataset \mathcal{D}' without a homomorphism from p to \mathcal{D}' but with a homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. Indeed, if such a dataset \mathcal{D}' exists, it contains $R(c_r^{\sigma(r)})$ for each r , and the homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$, in particular, the images of \bar{v} , identifies the extension of σ to \bar{v} that satisfies ψ , which is not possible by assumption. Therefore, (\mathcal{D}_0, f, p) is safe, as required.

Now we show the backward direction of the correctness claim. Let f be a strict \mathcal{D}_0 -suppressor of cost at most ℓ such that (\mathcal{D}_0, f, p) is safe—that is, for any external dataset \mathcal{D}' either there is a homomorphism from p to \mathcal{D}' or there is no homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. We need to prove that ϕ is valid. First, we claim that f anonymises all constants in \mathcal{D}_0 different from all c_r^{false} and c_r^{true} , for $r \in \bar{r}$ and all d_i , for $i = 1, \dots, 3n + 1$. Indeed, if this is not the case, and there is such a constant c , then the dataset that consists of the atoms $S(c, c)$, for each binary predicate S , and the atoms $A(c)$ for each unary predicate A different from **Anon**, would be witnessing \mathcal{D}' for unsafety. Next, note that f anonymises at least one of c_r^{false} and c_r^{true} , for each $r \in \bar{r}$, because otherwise the dataset consisting of two atoms $R(c_r^{\text{false}})$ and $R(c_r^{\text{true}})$ would be witnessing \mathcal{D}' for unsafety. However, since the cost of suppressor f is at most ℓ , f cannot anonymise both c_r^{false} and c_r^{true} for any r by construction. Consider the assignment σ of \bar{r} such that, for each $r \in \bar{r}$, $\sigma(r) = \text{false}$ if and only if f anonymises c_r^{true} . We claim that for any extension of σ to \bar{v} the resulting assignment turns ψ to **false**—that is, that ϕ is valid. Indeed, if this is not the case, then, on the base of such an extension to \bar{v} , we can construct a homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$, where \mathcal{D}' consists of atoms $R(c_r^{\sigma(r)})$ for all r : such a homomorphism sends

- x^{false} and x^{true} to the nulls anonymising c^{false} and c^{true} , respectively,
- y and z to the nulls anonymising c^y and c^z , respectively,
- each x_γ to the null anonymising c_γ^π , where π is the restriction of σ to the variables of γ , for γ a clause in ψ ,

- each x_r to $c_r^{\sigma(r)}$, for $r \in \bar{r}$, and
- each x_v to the null anonymising $c_v^{\sigma(v)}$, for $v \in \bar{v}$.

Therefore, ϕ is indeed valid, as required. \square

Having established the combined complexity of the open-world versions of the cost minimisation problems associated to safety, we next turn our attention to their data complexity. We will prove NP-hardness for both $\text{MIN-SAFETY}^{\text{ow}}$ and $\text{MIN-SAFETY}_s^{\text{ow}}$, which matches the upper bounds from Lemma 18. In the case of $\text{MIN-SAFETY}^{\text{ow}}$, we will be able to reuse the reduction in the proof of Lemma 8.

Lemma 21. *$\text{MIN-SAFETY}^{\text{ow}}$ is NP-hard in data complexity for RDF PPDP instances and Boolean policies.*

Proof. The proof of this lemma is based on the proof of Lemma 8 of the NP-hardness of $\text{MIN-COMPLIANCE}^{\text{ow}}$ in data complexity in the same way as the proof of Lemma 19 of hardness of $\text{MIN-SAFETY}^{\text{ow}}$ in combined complexity is based on the proof of statement 1 of Lemma 6 of hardness of $\text{MIN-COMPLIANCE}^{\text{ow}}$ in combined complexity for Boolean policies. Again, the only difference in the construction is that this time we use only nulls in the dataset \mathcal{D}_0 instead of constants. As a result, any external dataset \mathcal{D}' that does not imply the policy does not influence the answer to the policy on $f(\mathcal{D}_0) \cup \mathcal{D}'$, regardless of what f is, and hence safety boils down to compliance, which is NP-hard by Lemma 8. \square

The case of data complexity of $\text{MIN-SAFETY}_s^{\text{ow}}$ is similar to its combined complexity. First, we cannot reuse any existing reduction and have to develop a direct NP-hardness proof. Second, it works already for ground instances, but cannot be easily adapted to $\text{MIN-SAFETY}^{\text{ow}}$. Finally, it also works for Boolean policies, which is in contrast to the lower bound for $\text{MIN-COMPLIANCE}_s^{\text{ow}}$ in Lemma 9.

Lemma 22. *$\text{MIN-SAFETY}_s^{\text{ow}}$ is NP-hard in data complexity, even for ground RDF PPDP instances and Boolean policies.*

Proof. The proof is an adaptation of the proofs of Lemmas 8 and 9 of NP-hardness of $\text{MIN-COMPLIANCE}^{\text{ow}}$ and $\text{MIN-COMPLIANCE}_s^{\text{ow}}$ in data complexity, in a similar way as in the case of combined complexity where the proof of Lemma 20 is a modification of the proofs of Lemmas 6 and 7.

The proof is again by reduction of $\text{NODE-DELETION}_{\Pi}$ for Π the property of a graph not having cycles of length 3.

In the reduction, we will use a vocabulary consisting of two unary predicates U and Anon , as well as four binary predicates Edge_1 , Edge_2 , Edge_c , and P (so, the adaptation to the case of RDF is again straightforward).

Consider first the policy p , which is the following Boolean CQ:

$$\begin{aligned} \exists x_1, x_2, x_3, y_1, y_2, y_3, z. & \text{U}(x_1) \wedge \text{U}(x_2) \wedge \text{U}(x_3) \wedge \text{Anon}(y_1) \wedge \text{Anon}(y_2) \wedge \text{Anon}(y_3) \wedge \text{Anon}(z) \wedge \\ & \text{Edge}_1(x_1, y_1) \wedge \text{Edge}_2(y_1, x_2) \wedge \text{Edge}_1(x_2, y_2) \wedge \text{Edge}_2(y_2, x_3) \wedge \text{Edge}_1(x_3, y_3) \wedge \text{Edge}_2(y_3, x_1) \wedge \\ & \text{Edge}_c(y_1, z) \wedge \text{Edge}_c(y_2, z) \wedge \text{Edge}_c(y_3, z). \end{aligned}$$

Given a directed graph \mathcal{G} and an integer k as an input to the $\text{NODE-DELETION}_{\Pi}$ problem, we next construct a ground dataset \mathcal{D}_0 and an integer ℓ such that there exists a strict \mathcal{D}_0 -suppressor f with cost at most ℓ and such that (\mathcal{D}_0, f, p) is safe if and only if it is possible to delete at most k nodes from \mathcal{G} and obtain a graph without a cycle of length 3.

Let dataset \mathcal{D}_0 use a constant c , a constant c_v for each node v in \mathcal{G} , a constant d_e for each edge e in \mathcal{G} , and a constant d_i for each $i = 1, \dots, 2n$ with n the number of nodes in \mathcal{G} . Let also \mathcal{D}_0 consist of

- the atom $\text{Anon}(c)$;
- the atoms $\text{Edge}_1(c_u, d_e)$, $\text{Edge}_2(d_e, c_v)$, $\text{Edge}_c(d_e, c)$ and $\text{Anon}(d_e)$ for each edge $e = (u, v)$ of \mathcal{G} ; and
- the atoms $\text{P}(c_v, d_i)$, for each node v of \mathcal{G} with m_v the total number of incoming to v and outgoing from v edges and for each $i = m_v + 1, \dots, 2n$.

Let $\ell = (2n+1)k+5m+1$ for n and m the numbers of nodes and edges in \mathcal{G} , respectively—that is, ℓ is precisely the cost of the strict anonymisation of k constants c_v , all constants d_e , and constant c .

Next we prove that $\text{NODE-DELETION}_{\Pi}$ holds for a graph \mathcal{G} and an integer k if and only if there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ such that (\mathcal{D}_0, f, p) is safe. The first intuitive idea is similar to the one in Lemma 20: a safe suppressor should anonymise c and all d_e , because all these constants are in Anon and if any of them is not anonymised then we can construct an external dataset witnessing non-safety. So, a safe suppressor can anonymise c_v for at most k nodes v , which corresponds to deleting these v from the graph, because an external dataset can contain an U atom, required by the policy, only for v with c_v not anonymised. Note also that constant c and atoms Edge_c are needed to keep the anonymised part of \mathcal{D}_0 connected, so \mathcal{D}' cannot have just a part of it. Next we formally prove this intuition correct.

We start with the forward direction. Let a directed graph \mathcal{G} and a number k be such that it is possible to delete at most k nodes from \mathcal{G} with the resulting graph not having a cycle of length 3. We need to prove that there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ such that (\mathcal{D}_0, f, p) is safe—that is, for any external dataset \mathcal{D}' either there is a homomorphism from p to \mathcal{D}' or there is no homomorphism from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. To this end, consider a subgraph \mathcal{G}' of \mathcal{G} with the required properties. Consider also the strict \mathcal{D}_0 -suppressor f that anonymises all constants c_v for v deleted from \mathcal{G} , all constants d_e for edges e of \mathcal{G} , and constant c . The cost of f is at most ℓ , as required. We claim that (\mathcal{D}_0, f, p) is safe. Indeed, assume, for the sake of contradiction, that this is not the case and there exists an external dataset \mathcal{D}' without a homomorphism from p to \mathcal{D}' and with a homomorphism g from p to $f(\mathcal{D}_0) \cup \mathcal{D}'$. If $g(z)$ is mapped to something other than the anonymisation of c under f , then, by construction, g is also a homomorphism to \mathcal{D}' alone, which is not allowed by the assumption on \mathcal{D}' . So, $g(z)$ is the anonymisation of c , and therefore each $g(y_i)$ is the anonymisation of d_{e_i} , for e_i an edge of \mathcal{G} . However, each $g(x_i)$ must be c_i , for some node c_i that is not anonymised, because \mathcal{D}_0 does not have any U atoms, while p has $\text{U}(x_i)$. In other words, these c_i are in \mathcal{G}' , and these nodes form a cycle of length 3 in \mathcal{G}' . This contradicts the construction of \mathcal{G}' . Hence, our assumption was wrong and (\mathcal{D}_0, f, p) is safe, as required.

For the backward direction of the claim, assume that there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ such that (\mathcal{D}_0, f, p) is safe. We need to prove that it is possible to delete at most k nodes from \mathcal{G} such that the resulting graph does not have a cycle of length 3. To this end, consider such a suppressor f . First, note that f anonymises constant c and all constants d_e by the same reason as in Lemma 20: otherwise the external dataset \mathcal{D}' consisting of the atoms that mention this constant in all positions over all predicates in the vocabulary except for **Anon** is a counterexample for safety of (\mathcal{D}_0, f, p) . Consider now the subgraph \mathcal{G}' of \mathcal{G} that is obtained by removing all nodes v such that c_v is anonymised by f . First, the number of removed nodes is at most k , because the cost of f is at most ℓ . Second, \mathcal{G}' does not have a cycle of length 3, because otherwise \mathcal{D}' that consists of $\mathcal{U}(c_v)$ for all nodes v of \mathcal{G}' would be a counterexample for safety. So, **NODE-DELETION**_{II} holds for \mathcal{G} and k , as required. \square

We now study the combined complexity of the general **MIN-SAFETY** and **MIN-SAFETY**_s problems. In both cases we can show **CONEXPTIME**-hardness by reusing the reduction in the proof of Lemma 16, where we showed **CONEXPTIME**-hardness of **SAFETY** by reduction of the **EXPTILING** problem. These bounds match the upper bounds for the problems in Lemma 18. Same as in Lemma 16, we do not require the closures to be fixed, singleton, atomic or existential-free, and it is an interesting problem for future work to establish the precise complexity of **MIN-SAFETY** and **MIN-SAFETY**_s when any combination of these restrictions is enforced. Additionally, as in Lemmas 19 and 21, we do not restrict instances to be ground for the non-strict version of the problem (but we do for the strict version).

Lemma 23. *The following holds:*

1. **MIN-SAFETY** is **CONEXPTIME**-hard for *RDF PPDP instances and Boolean policies*;
2. **MIN-SAFETY**_s is **CONEXPTIME**-hard for *ground RDF PPDP instances and Boolean policies*.

Proof. In both cases, we can essentially reuse the reduction of **EXPTILING** to **SAFETY** in the proof of Lemma 16. In the case of **MIN-SAFETY**, the only modification in the construction is that we take $f(\mathcal{D}_0)$ from Lemma 16 as the dataset \mathcal{D}_0 in the input to **MIN-SAFETY**, and set $\ell = 0$ (note here that we do not require instances to be ground in the first statement of this lemma). In the case of **MIN-SAFETY**_s, the only modification is that we do not fix the strict suppressor as in Lemma 16, but allow it to be arbitrary—that is, set $\ell = 6n + 4$, which is the cost of f in the proof of Lemma 16 (note that it is important to use $2n$ copies of 0 and 1 here, as described in the end of the proof of Lemma 16, so that f is the strict suppressor with the maximal possible cost). In both cases, the rest of the proof goes along the same lines as the proof of Lemma 16. \square

To conclude this section we provide Σ_3^P lower bounds in data complexity for **MIN-SAFETY** and **MIN-SAFETY**_s, thus matching the upper bounds from Lemma 12. The proof for **MIN-SAFETY** in the following lemma uses the same ideas as the proof of Lemma 17, where we proved the Π_2^P -lower bound of **SAFETY** in data complexity, extended with a gadget to deal with the third level of the polynomial hierarchy. Again, we do not require the instances to be ground and closures to be singleton or atomic, and it would be interesting in future

work to establish the complexity with any of these restrictions. However, note that CQs in the closures are restricted to be existential-free.

Lemma 24. *MIN-SAFETY is Σ_3^P -hard in data complexity for RDF PPDP instances, Boolean policies, and existential-free CQs in closures.*

Proof. We show Σ_3^P -hardness by reduction of the $\exists\forall\exists$ SAT problem, whose input is a formula $\phi = \exists \bar{s}. \forall \bar{u}. \exists \bar{v}. \psi$ with ψ a formula in 3CNF over variables $\bar{s} \cup \bar{u} \cup \bar{v}$, and the answer is yes if and only if ϕ is valid. Problem $\exists\forall\exists$ SAT is a canonical Σ_3^P -complete problem.

Let $\phi = \exists \bar{s}. \forall \bar{u}. \exists \bar{v}. \psi$ be a formula with ψ in 3CNF. We construct a dataset \mathcal{D}_0 , an integer ℓ , a Boolean policy p and a set of closures \mathcal{C} with existential-free CQs such that p and queries in \mathcal{C} do not depend on ϕ , and such that there exists a \mathcal{D}_0 -suppressor f of cost at most ℓ with the PPDP instance (\mathcal{D}_0, f, p) being safe with respect to \mathcal{C} if and only if ϕ is valid. As usual, the arities of the used predicates are bounded by 2, so the adaptation to the RDF case is straightforward.

As we did in the proof of Lemma 17, in the first step of the reduction we transform ϕ to an equivalent formula $\phi' = \exists \bar{s}'. \forall \bar{u}'. \exists \bar{v}'. \psi'$, where ψ' is a propositional formula in 9CNF such that its each clause has precisely 3 literals over variables in each of \bar{s}' , \bar{u}' , and \bar{v}' . We can do it in the same way as in the proof of Lemma 17, so we omit this step here for brevity.

We then continue the description of the reduction with the definition of predicates in the vocabulary: let it contain unary predicates S and U , and binary predicates V , $S\text{Values}^{\text{false}}$, $S\text{Values}^{\text{true}}$, $U\text{Values}$, $V\text{Values}$, $CF_1^s, \dots, CF_3^s, CF_1^u, \dots, CF_3^u, CF_1^{bv}, CF_1^{cv}, \dots, CF_3^{bv}, CF_3^{cv}$ (as required, the vocabulary does not depend on ϕ). Similar to the proofs of Lemmas 16 and 17, we will use the following abbreviation, for variables and constants $t, t_1^s, \dots, t_3^s, t_1^u, \dots, t_3^u, t_1^{bv}, t_1^{cv}, \dots, t_3^{bv}, t_3^{cv}$:

$$\begin{aligned} \text{ClauseFalsification}(t, t_1^s, \dots, t_3^s, t_1^u, \dots, t_3^u, t_1^{bv}, t_1^{cv}, \dots, t_3^{bv}, t_3^{cv}) = \\ CF_1^s(t, t_1^s) \wedge \dots \wedge CF_3^s(t, t_3^s) \wedge CF_1^u(t, t_1^u) \wedge \dots \wedge CF_3^u(t, t_3^u) \wedge \\ CF_1^{bv}(t, t_1^{bv}) \wedge CF_1^{cv}(t, t_1^{cv}) \wedge \dots \wedge CF_3^{bv}(t, t_3^{bv}) \wedge CF_3^{cv}(t, t_3^{cv}). \end{aligned}$$

We also set $\ell = 2|\bar{s}'|$.

Next we define the dataset \mathcal{D}_0 . Let it consist of,

- for each $s \in \bar{s}'$, the atoms

$$\begin{aligned} S\text{Values}^{\text{false}}(\mathbf{b}_s^i, \mathbf{b}_s^{\text{false}}), S\text{Values}^{\text{true}}(\mathbf{b}_s^i, \mathbf{b}_s^{\text{true}}), \quad \text{for each } i = 1, \dots, \ell + 1, \\ S(\mathbf{b}_s^{\text{false}}, \mathbf{b}_s^{\text{false}}), \text{ and } S(\mathbf{b}_s^{\text{true}}, \mathbf{b}_s^{\text{true}}), \end{aligned}$$

where \mathbf{b}_s^i , $\mathbf{b}_s^{\text{false}}$, and $\mathbf{b}_s^{\text{true}}$ are fresh nulls associated with s ;

- for each $v \in \bar{v}'$, the atom $V\text{Values}(\mathbf{b}_v, c_v)$, where \mathbf{b}_v and c_v are a fresh null and constant associated with v ;
- for each clause γ in ψ' with the literals over variables $s_1, \dots, s_3 \in \bar{s}'$, $u_1, \dots, u_3 \in \bar{u}'$, and $v_1, \dots, v_3 \in \bar{v}'$, the atoms

$$\text{ClauseFalsification}(\mathbf{b}_\gamma, t_{s_1}, \dots, t_{s_3}, a_{u_1}, \dots, a_{u_3}, \mathbf{b}_{v_1}, a_{v_1}, \dots, \mathbf{b}_{v_3}, a_{v_3}),$$

where

- b_γ is a fresh null associated with γ ,
- for each $s = s_1, \dots, s_3$, parameter t_s is null b_s^{false} if the literal with s is positive in γ and null b_s^{true} otherwise, and,
- for each $w = u_1, \dots, u_3, v_1, \dots, v_3$, parameter a_w is constant c_w^{false} if the literal with w is positive in γ and constant c_w^{true} otherwise,

(essentially, these atoms represent the falsifying assignment of the clause).

Next, let the policy p be the conjunction of the Boolean CQ p_{cf} defined as

$$\begin{aligned} \exists z, y_1^s, \dots, y_3^s, x_1, \dots, x_3, y_1^{\text{bv}}, y_1^{\text{cv}}, \dots, y_3^{\text{bv}}, y_3^{\text{cv}}. \\ \text{ClauseFalsification}(z, y_1^s, \dots, y_3^s, x_1, \dots, x_3, y_1^{\text{bv}}, y_1^{\text{cv}}, \dots, y_3^{\text{bv}}, y_3^{\text{cv}}) \wedge \\ S(y_1^s, y_1^s) \wedge \dots \wedge S(y_3^s, y_3^s) \wedge U(x_1) \wedge \dots \wedge U(x_3) \wedge V(y_1^{\text{bv}}, y_1^{\text{cv}}) \wedge \dots \wedge V(y_3^{\text{bv}}, y_3^{\text{cv}}) \end{aligned}$$

and the Boolean CQ p_s defined as

$$\begin{aligned} \exists x, x^{\text{false}}, x^{\text{true}}. \\ \text{SValues}^{\text{false}}(x, x^{\text{false}}) \wedge \text{SValues}^{\text{true}}(x, x^{\text{true}}) \wedge S(x^{\text{false}}, x^{\text{false}}) \wedge S(x^{\text{true}}, x^{\text{true}}); \end{aligned}$$

note that p does not depend on ϕ , as required.

To complete the construction, we need to define the set of closures \mathcal{C} . Let it be precisely as in the proof of Lemma 17—that is, it consist of the following pairs, where all CQs are existential-free and do not depend on ϕ :

- $[\text{UValues}(x_1, x_2), \{(c_u^{\text{false}}, c_u^{\text{true}}) \mid u \in \bar{u}'\}]$;
- $[\text{UValues}(x_1, x_2) \wedge U(x_1) \wedge U(x_2), \emptyset]$;
- $[\text{VValues}(x, y), \{(d_v^{\text{false}}, c_v), (d_v^{\text{true}}, c_v) \mid v \in \bar{v}'\}]$; and
- $[\text{V}(x, y), \{(d_v^{\text{false}}, c_v^{\text{false}}), (d_v^{\text{true}}, c_v^{\text{true}}) \mid v \in \bar{v}'\}]$.

We next establish the correctness of the reduction—that is, we show that ϕ' is valid if and only if there exists a \mathcal{D}_0 -suppressor of cost at most ℓ such that the resulting PPDP instance is safe with respect to the closures.

For the forward direction, let ϕ' be valid—that is, there exists an assignment σ of \bar{s}' such that for every extension of σ to \bar{u}' there is its further extension to \bar{v}' such that all clauses in ψ' evaluate to **true** under the overall assignment. We show that there exists a \mathcal{D}_0 -suppressor f such that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} —that is, for every external dataset \mathcal{D}' , either $\mathcal{D}' \cup \mathcal{C} \models p$ or $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \not\models p$. To this end, consider the \mathcal{D}_0 -suppressor f that anonymises to a fresh null, for each $s \in \bar{s}'$, the first position of the atom $S(b_s^{\text{true}}, b_s^{\text{true}})$ in \mathcal{D}_0 , if $\sigma(s) = \text{false}$, and the first position of $S(b_s^{\text{false}}, b_s^{\text{false}})$ otherwise. Note that the cost of f is $\ell = 2|\bar{s}'|$, as required.

We need to show that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} . Let, for the sake of contradiction, this not be the case—that is, there exist a dataset \mathcal{D}' such that $\mathcal{D}' \cup \mathcal{C} \not\models p$ but $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. By Lemma 11 we may assume that \mathcal{D}' is ground. Since $\mathcal{D}' \cup \mathcal{C} \not\models p$, $\mathcal{D}' \cup \mathcal{C}$ is satisfiable, and, therefore, there is no $u \in \bar{u}'$ such that both $U(c_u^{\text{false}})$

and $U(c_u^{\text{true}})$ are in \mathcal{D}' . Consider the extension of σ to \bar{u}' such that, for each $u \in \bar{u}'$, $\sigma(u)$ is **false** if $U(c_u^{\text{false}}) \in \mathcal{D}'$ and **true** otherwise (including the case when none of $U(c_u^{\text{false}})$ and $U(c_u^{\text{true}})$ are in \mathcal{D}'). To falsify our assumption, we claim that ψ' evaluates to **false** under any further extension of σ to \bar{v}' . To this end, consider any such extension and the Herbrand model I of \mathcal{D}' extended by

- all the V , $U\text{Values}$ and $V\text{Values}$ atoms enforced by \mathcal{C} , and
- all $S\text{Values}^{\text{false}}$, $S\text{Values}^{\text{true}}$, S , and *ClauseFalsification* atoms in $f(\mathcal{D}_0)$ with each null b_v replaced either by c_v^{false} if $\sigma(v) = \text{false}$ or by c_v^{true} if $\sigma(v) = \text{true}$, and each other null by a fresh constant.

By construction, $I \models f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$. Therefore, by assumption, $I \models p$ —that is, there is a homomorphism from p to I . The images of atoms in the p_s part of p under this homomorphism correspond to atoms in \mathcal{D}' , because either $S(b_s^{\text{false}}, b_s^{\text{false}})$ or $S(b_s^{\text{true}}, b_s^{\text{true}})$ is not in $f(\mathcal{D}_0)$ for any $s \in \bar{s}'$ by construction, and because all the $S\text{Values}^{\text{false}}$, $S\text{Values}^{\text{true}}$ and S atoms in \mathcal{D}_0 use only nulls. The images of the U atoms of the p_{cf} part of p also correspond to atoms in \mathcal{D}' , because \mathcal{D}_0 does not mention U . The images of V atoms correspond to the atoms enforced by \mathcal{C} . The images of the CF atoms in the *ClauseFalsification* conjunction of p_{cf} , which all have a common variable z , and the S atoms of p_{cf} should correspond either all to atoms in \mathcal{D}' or all to atoms in $f(\mathcal{D}_0)$ with nulls replaced as described, because all nulls b_γ , b_s^{false} and b_s^{true} are replaced by fresh constants, and *ClauseFalsification* and S are not mentioned in \mathcal{C} . However, they cannot correspond to atoms in \mathcal{D}' , because otherwise we would have $\mathcal{D}' \cup \mathcal{C} \models p$, which is not allowed by assumption. So, the *ClauseFalsification* atoms are sent to the interpretations of atoms in $f(\mathcal{D}_0)$ with the replaced nulls, which means that there exists a clause in ψ' falsified under the extended σ , same as in the proof of Lemma 17. Therefore, the extension of σ to \bar{u}' is such that ψ' is not valid for any further extension to \bar{v}' . Hence, our assumption was wrong, and (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} , as required.

For the backward direction, let there exist a \mathcal{D}_0 -suppressor f such that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} —that is, for every external dataset \mathcal{D}' , either $\mathcal{D}' \cup \mathcal{C} \models p$ or $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \not\models p$. We need to show that ϕ' be valid.

First, we show that f anonymises at least one of the four positions in the atoms $S(b_s^{\text{false}}, b_s^{\text{false}})$ and $S(b_s^{\text{true}}, b_s^{\text{true}})$ in \mathcal{D}_0 for each $s \in \bar{s}'$. Indeed, if this is not the case and there is s with both of these atoms in $f(\mathcal{D}_0)$, then (\mathcal{D}_0, f, p) is not safe, because $f(\mathcal{D}_0)$ has a homomorphism from part p_s of policy p and we can take as a witnessing counterexample for safety of the external dataset \mathcal{D}' that is obtained from p_{cf} by replacing all the variables by fresh nulls (or by constants in a way that $\mathcal{D}' \cup \mathcal{C}$ is consistent). In particular, a homomorphism from p_s to $f(\mathcal{D}_0)$ exists because f cannot anonymise, within the given cost ℓ , a position in the pair of atoms $S\text{Values}^{\text{false}}(b_s^i, b_s^{\text{false}})$, $S\text{Values}^{\text{true}}(b_s^i, b_s^{\text{true}})$ in \mathcal{D}_0 for every i . Also, $\mathcal{D}' \cup \mathcal{C} \not\models p$ holds for such \mathcal{D}' , which is required for witnessing non-safety, because \mathcal{D}' does not have any $S\text{Values}$ atoms needed for part p_s of p .

On the one hand, we proved that f anonymises a position in $S(b_s^{\text{false}}, b_s^{\text{false}})$ and $S(b_s^{\text{true}}, b_s^{\text{true}})$ for each $s \in \bar{s}'$. On the other, the cost of f is bounded by $\ell = 2|\bar{s}'|$. Therefore, f anonymises exactly one of these positions for each s . Consider the assignment σ of \bar{s}' defined, for each s , as $\sigma(s) = \text{true}$ if a position in $S(b_s^{\text{false}}, b_s^{\text{false}})$ is anonymised and $\sigma(s) = \text{false}$ otherwise.

We next claim that for any extension of σ to \bar{u}' there exists a further extension to \bar{v}' such that ψ' evaluates to **true** under the overall assignment. Let, for the sake of contradiction, this not be the case, and consider a witnessing extension of σ to \bar{u}' . We next show that (\mathcal{D}_0, f, p) is not safe, which contradicts our original assumption. To this end, consider the external dataset \mathcal{D}' that consists of

- atoms $U(c_u^{\text{false}})$ for each $u \in \bar{u}'$ such that $\sigma(u) = \text{false}$ and atoms $U(c_u^{\text{true}})$ for each $u \in \bar{u}'$ such that $\sigma(u) = \text{true}$; and
- all the atoms obtained from atoms in p_s by replacing all the variables by fresh constants.

We have that $\mathcal{D}' \cup \mathcal{C} \not\models p$, because $\mathcal{D}' \cup \mathcal{C}$ is satisfiable and does not mention any atoms over CF predicates, which are required by p . We also claim that $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \models p$. Indeed, any model I of $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C}$ witnesses each \mathbf{b}_v by either d_v^{false} or d_v^{true} , which corresponds to an extension of assignment σ to \bar{v}' . Therefore, by assumption, there is a clause in ψ' that evaluates to **false** under the extended σ . By construction, this precisely means that the interpretation of *ClauseFalsification* in I contains a tuple that is required for a homomorphism from p_{cf} to I . Since a copy of p_s is in \mathcal{D}' , overall p holds in I , and (\mathcal{D}_0, f, p) is not safe, as required. \square

Finally, we prove that MIN-SAFETY_s is Σ_3^P -hard in data complexity, again same as MIN-SAFETY . The proof is similar to the proof of Lemma 24, but also uses the ideas from the proofs of Lemmas 20 and 22 of Σ_2^P - and NP-hardness of $\text{MIN-SAFETY}_s^{\text{ow}}$ in combined and data complexity, respectively. We do not give here the full reduction in this case, but rather describe in detail the differences with the reduction in Lemma 24. Note that it could be again an interesting problem for future work to determine the precise complexity of the problem in the case when the closures are restricted to be singleton or atomic, or both.

Lemma 25. *MIN-SAFETY_s is Σ_3^P -hard in data complexity for ground RDF PPDP instances, Boolean policies, and existential-free CQs in closures.*

Proof. This proof is again by reduction of $\exists\forall\exists\text{3SAT}$. As we just mentioned, it is very similar to the proof of the previous Lemma 24. The first conceptual difference is that strict suppressors require another (in fact, simpler) gadget to deal with the third level of the polynomial hierarchy. The second difference is that to deal with ground PPDP instances we use the same ideas as in proofs of Lemmas 20 and 22.

Let $\phi = \exists \bar{s}. \forall \bar{u}. \exists \bar{v}. \psi$ be a formula with ψ in 3CNF. We construct a ground dataset \mathcal{D}_0 , an integer ℓ , a Boolean policy p and a set of closures \mathcal{C} with existential-free CQs such that p and queries in \mathcal{C} do not depend on ϕ , and such that there exists a strict \mathcal{D}_0 -suppressor f of cost at most ℓ with the PPDP instance (\mathcal{D}_0, f, p) being safe with respect to \mathcal{C} if and only if ϕ is valid. As usual, the arities of the used predicates are bounded by 2, so the adaptation to the RDF case is straightforward.

As in the proof of Lemma 24, we first transform ϕ to an equivalent $\phi' = \exists \bar{s}'. \forall \bar{u}'. \exists \bar{v}'. \psi'$ with ψ' in 9CNF such that each clause has precisely 3 literals over variables in each of \bar{s}' , \bar{u}' , and \bar{v}' .

The vocabulary used in this reduction is almost the same as in the proof of Lemma 24: the only differences are that S is unary instead of binary, there is only one $S\text{Values}$ binary

predicate instead of two $SValues^{\text{false}}$ and $SValues^{\text{true}}$, and that a unary **Anon** and a binary **P** predicates are additionally used (their roles are very similar to the ones in the proof of Lemma 20). We will also use the abbreviation *ClauseFalsification*, which is the same as in the proof of Lemma 24.

The dataset \mathcal{D}_0 is also similar to the one in the proof of Lemma 24. The only differences are that

- instead of all $SValues^{\text{false}}$, $SValues^{\text{true}}$ and **S** atoms, dataset \mathcal{D}_0 contains the atom $SValues(c_s^{\text{false}}, c_s^{\text{true}})$, for each $s \in \bar{s}'$, where c_s^{false} and c_s^{true} are fresh constants associated with s ;
- in the **VValues** and *ClauseFalsification* atoms instead of nulls b_v for each $v \in \bar{v}'$, nulls b_γ for each clause γ in ψ' , and nulls b_s^{false} and b_s^{true} for each $s \in \bar{s}'$, \mathcal{D}_0 uses constants d_v^{false} , constants c_γ , and constants c_s^{false} and c_s^{true} , respectively; and
- additionally, dataset \mathcal{D}_0 contains the atom **Anon**(c_γ) for each clause γ , the atoms **Anon**(d_v^{false}) for each $v \in \bar{v}'$, and the atoms of the form $R(d_v^{\text{false}}, d_i)$ that make the cost of the strict anonymisations of all d_v^{false} to be equal (in exactly the same way as done with constants c_u^{false} and c_u^{true} in the proof of Lemma 20).

We next set ℓ to be the cost of the strict anonymising all c_γ , all d_v^{false} and all c_s^{false} .

Next, let the policy p be again the conjunction of the Boolean CQs p_{cf} and p_s , both of which are very similar to the ones in the proof of Lemma 24: the only differences are that

- p_{cf} has atoms $S(y_1^s) \wedge \dots \wedge S(y_3^s)$ instead of their binary versions;
- p_{cf} additionally has atoms **Anon**(y_1^{bv}), \dots , **Anon**(y_3^{bv}) and **Anon**(z); and
- p_s has the form $\exists x^{\text{false}}, x^{\text{true}}. SValues(x^{\text{false}}, x^{\text{true}}) \wedge S(x^{\text{false}}) \wedge S(x^{\text{true}})$.

Finally, the set of closures \mathcal{C} is precisely the same as in the proof of Lemma 24.

We next argue the correctness of the reduction—that is, we show that ϕ' is valid if and only if there exists a strict \mathcal{D}_0 -suppressor of cost at most ℓ such that the resulting PPDP instance is safe with respect to the closures.

For the forward direction, let ϕ' be valid—that is, there exists an assignment σ of \bar{s}' such that for every extension of σ to \bar{u}' there is a further extension to \bar{v}' such that all clauses in ψ' evaluate to **true** under the overall assignment. We show that there exists a strict \mathcal{D}_0 -suppressor f such that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} —that is, for every external dataset \mathcal{D}' , either $\mathcal{D}' \cup \mathcal{C} \models p$ or $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \not\models p$. To this end, consider the strict \mathcal{D}_0 -suppressor f that anonymises to fresh nulls

- constant c_γ for each clause γ ,
- constant d_v^{false} for each $v \in \bar{v}'$,
- constant c_s^{true} for each $s \in \bar{s}'$ such that $\sigma(s) = \text{false}$ and c_s^{false} for each s such that $\sigma(s) = \text{true}$.

The cost of f is precisely ℓ by construction. Moreover, the proof that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} goes along the same lines as the proof of the same fact in Lemma 24, except for two minor differences. First, the images of atoms in the p_s part of p under the homomorphism from p to I correspond to atoms in \mathcal{D}' by another reason, in particular, because p_s requires both of its variables to be in S , and predicate S is not mentioned in \mathcal{D}_0 and \mathcal{C} . Second, the images of the S atoms of part p_{cf} correspond to atoms in \mathcal{D}' by the same reason, so only the CF atoms correspond to atoms in $f(\mathcal{D}_0)$.

For the backward direction, assume that there exists a strict \mathcal{D}_0 -suppressor f such that the PPDP instance (\mathcal{D}_0, f, p) is safe with respect to \mathcal{C} —that is, for every external dataset \mathcal{D}' , either $\mathcal{D}' \cup \mathcal{C} \models p$ or $f(\mathcal{D}_0) \cup \mathcal{D}' \cup \mathcal{C} \not\models p$. We need to show that ϕ' be valid.

First, we claim that f anonymises all c_γ and all d_v^{false} . Indeed, if it is not the case for such a constant c , then $f(\mathcal{D}_0)$ contains the atom $\text{Anon}(c)$, and we can construct an external dataset \mathcal{D}' witnessing non-safety. In particular, if c is c_γ for some γ , then \mathcal{D}' consists of all the atoms of p except for $\text{Anon}(z)$ with z replaced by c , all y_i^{bv} by $d_{v_1}^{\text{false}}$ where v_1 is the first variable of \bar{v}' , all y_i^{cv} by $c_{v_1}^{\text{false}}$, and all other variables by fresh constants. Also, if c is d_v^{false} for some $v \in \bar{v}'$, then \mathcal{D}' consists of all the atoms of p except for $\text{Anon}(y_1^{\text{bv}}), \dots, \text{Anon}(y_3^{\text{bv}})$ with all y_i^{bv} replaced by d_v^{false} , all y_i^{cv} by c_v^{false} , and all other variables by fresh constants.

We also claim that f anonymises at least one of c_s^{false} and c_s^{true} for each $s \in \bar{s}'$. Indeed, if this is not the case and there is s with the atom $S\text{Values}(c_s^{\text{false}}, c_s^{\text{true}})$ in $f(\mathcal{D}_0)$, then (\mathcal{D}_0, f, p) is not safe, with a witnessing \mathcal{D}' consisting of atoms $S(c_s^{\text{false}})$ and $S(c_s^{\text{true}})$, as well as all the atoms obtained from p_{cf} by replacing all the variables by fresh nulls (same as in the proof of Lemma 24).

So, we proved that f anonymises each c_γ , each d_v^{false} and at least one of each pair c_s^{false} and c_s^{true} . Therefore, the cost bound ℓ allows to anonymise exactly one of c_s^{false} and c_s^{true} for each s . So we can consider the assignment σ of \bar{s}' defined, for each s , as $\sigma(s) = \text{true}$ if c_s^{false} is anonymised and $\sigma(s) = \text{false}$ otherwise.

Finally, we argue that for any extension of σ to \bar{u}' there exists a further extension to \bar{v}' such that ψ' evaluates to **true** under the overall assignment. The proof of this fact goes along the same lines as the proof of the same fact in Lemma 24 with one minor modification: the external dataset contains additionally atoms $S(c_s^{\text{false}})$ and $S(c_s^{\text{true}})$ for every $s \in \bar{s}'$. \square

The following theorem summarises our results on the cost minimisation problems associated to linkage safety, established in Lemmas 18–25.

Theorem 4. *The following results hold:*

1. $\text{MIN-SAFETY}^{\text{ow}}$ and $\text{MIN-SAFETY}_s^{\text{ow}}$ are both in Σ_3^p and Σ_2^p -hard in combined complexity, and NP-complete in data complexity; all lower bounds hold already for RDF PPDP instances and Boolean policies, while the $\text{MIN-SAFETY}_s^{\text{ow}}$ lower bounds hold already if PPDP instances are additionally required to be ground; and
2. MIN-SAFETY and MIN-SAFETY_s are both CONEXPTIME-complete in combined complexity and Σ_3^p -complete in data complexity; all lower bounds hold already for RDF PPDP instances and Boolean policies, the $\text{MIN-SAFETY}_s^{\text{ow}}$ lower bounds hold already if PPDP instances are additionally required to be ground, and the data complexity lower bounds hold already if the closure CQs are additionally required to be existential-free.

6. Related Work

The problem of preventing disclosure of sensitive data in information systems while ensuring that the data remains maximally accessible to users has received significant attention in the literature. Existing approaches can be roughly categorised as follows.

1. *Perturbation models*, where (typically numeric) data is modified by the introduction of noise (e.g., in the form of a random variable with suitable properties). The aim is for the data to remain useful for statistical analysis, rather than to preserve its integrity. This includes, for instance, work on *differential privacy* (Chawla, Dwork, McSherry, Smith, & Wee, 2005; Dwork, 2006, 2008; Dwork, Naor, Reingold, Rothblum, & Vadhan, 2009).

2. *View-based models*, where the data that is made accessible (or inaccessible) to users is declaratively defined by means of views expressed in a logic-based language. In contrast to anonymisation approaches, data in view-based models is not modified; instead, access to it is controlled by means of a declaratively specified layer. These approaches have been so far the main focus of research in the context of RDF and ontologies (Abel, De Coi, Henze, Koesling, Krause, & Olmedilla, 2007; Bonatti & Sauro, 2013; Calvanese, De Giacomo, Lenzerini, & Rosati, 2012; Cuenca Grau, Kharlamov, Kostylev, & Zheleznyakov, 2015; Flouris, Fundulaki, Michou, & Antoniou, 2010; Kagal & Pato, 2010; Kirrane, Abdelrahman, Mileo, & Decker, 2013; Benedikt, Cuenca Grau, & Kostylev, 2017; Benedikt, Bourhis, ten Cate, & Puppis, 2016).

3. *Anonymisation models*, where some data is suppressed, by, for example, replacing constants with generated identifiers, or generalised, by, for example, replacing a numeric value with a value range, in a way that preserves data integrity. This includes work on *k*-anonymity and related notions in databases (Samarati, 2001; Sweeney, 2002; Bayardo & Agrawal, 2005; Machanavajjhala, Kifer, Gehrke, & Venkitasubramaniam, 2007), graph anonymisation (Backstrom, Dwork, & Kleinberg, 2007; Hay, Miklau, Jensen, Towsley, & Li, 2010; Liu & Terzi, 2008; Zhou & Pei, 2008), as well as our prior work on anonymisation in linked data (Cuenca Grau & Kostylev, 2016).

We next discuss each of these approaches in more detail.

6.1 Perturbation Models

Differential privacy (Chawla et al., 2005; Dwork, 2006, 2008; Dwork et al., 2009) is a prominent approach for publishing quantitative facts about a population in a way that the personal details of each individual in the population remain protected.

To understand the kinds of scenarios in which differential privacy can be applied and the guarantees provided, consider the case of a researcher who is running a survey and asks a number of people to submit their answers anonymously. The researcher collects in a dataset \mathcal{D} the data from all participants, performs some analysis on the data, and finally releases the results \mathcal{R} of the analysis (not \mathcal{D} itself in any form) to the public. The released results \mathcal{R} are typically conceptualised as the answers to a collection of aggregate queries; for instance, if the study is about the prevalence of diabetes in the UK, the released results \mathcal{R} could include the number of people participating in the survey, the number of people with diabetes by age group and gender, the number of overweight participants, and so on. Differential privacy is a guarantee from the researcher to each individual who took the survey; it ensures that the removal or addition of any individual to \mathcal{D} will not “substantially”

(as per a parameter of the framework set by the researcher) change the released results \mathcal{R} . Thus, for each individual i , an attacker would not be able to reliably tell by looking at \mathcal{R} whether the results were obtained from \mathcal{D} or from the dataset \mathcal{D}' in which the information about i has been removed. Differential privacy thus hides the differences between datasets that differ in one individual. It does not, however, provide absolute privacy guarantees, and hence makes no assumption about the background knowledge an attacker may have; for instance, regardless of whether *Bob* took the survey or not, the fact that he is over 60 and overweight would tell an attacker that he has higher chances than average of suffering from diabetes. In practice, differential privacy is achieved by introducing noise in each of the numeric quantities released in the results \mathcal{R} . The amount of noise that is needed in order to provide the guarantee is dependent on the parameters set by the researcher, as well as on the data \mathcal{D} and the queries generating the results. Differential privacy methods have been mostly developed in the context of databases, and there has also been some interest in extending them to Linked Data as well (Aron, 2013).

Differential privacy is a very useful approach in scenarios where the information released to the public consists of results from a numerical analysis. There are situations, however, where there is a need for publishing or exchanging actual data, and not just statistical information. In such cases, view-based or anonymisation approaches are more readily applicable.

6.2 View-Based Models

Miklau and Suciu (2007) study the problem of whether a given set of views logically discloses information about a secret, where both the views and the secret are expressed as conjunctive queries. They introduce and study the *perfect privacy* guarantee, which requires that the views and the secret do not have a common critical tuple—a notion that we also exploit in our paper to obtain complexity lower bounds.³

Benedikt et al. (2016) consider the scenario in which the relations in a database are partitioned into visible (where the contents are fully available to users) and hidden. A background logical theory provides semantic information (e.g., integrity constraints) about both types of relations. Analogously to our privacy model, sensitive information is specified by means of a query, and the problem is to determine whether any answer to such query can be logically derived from the contents of the visible relations and the background theory.

In the *Controlled Query Evaluation (CQE)* framework, the data is assumed to be hidden and users interact with the system by means of a query interface. Analogously to our anonymisation framework, as well as most view-based approaches, the sensitive information is also represented as a policy query. A key component of the framework is the *censor*, which ensures that answers to user queries that may compromise the policy are either distorted, or not returned to users. CQE was introduced in the context of databases (Sicherman, de Jonge, & van de Riet, 1983) and has received significant attention since (Biskup & Bonatti, 2004; Biskup & Weibert, 2008; Bonatti, Kraus, & Subrahmanian, 1995). CQE has also been recently extended to RDF and ontologies (Bonatti & Sauro, 2013; Cuenca Grau et al., 2015; Studer & Werner, 2014). Censors can be realised in different ways, and the possibility studied in the literature that is closest to our work is to consider censors that construct an anonymisation of the underpinning data (Cuenca Grau et al., 2015).

3. We refer the reader back to Section 5 for additional details on the critical tuple problem.

Furthermore, censors provide privacy guarantees comparable with our policy compliance notion: a malicious user should not be able to find out any answer to the policy by posing any number of queries to the system. The focus on CQE, however, is not on deciding whether the privacy guarantee is satisfied, but rather on constructing censors that satisfy it; hence, their technical results are incomparable to ours. Furthermore, although CQE allows for the formalisation of external background knowledge, the CQE literature does not consider guarantees akin to linkage safety.

A number of recent works focus on information disclosure in the context of ontology-based data access (OBDA)—a popular approach to data integration where various data sources are linked by means of declaratively specified mappings to an ontology, which provides both relevant domain background knowledge and the vocabulary for users to formulate queries. In the context of OBDA, users do not have direct access to data sources, and can only retrieve information by querying the ontology; thus, their knowledge about the data is inherently incomplete. In our recent work (Benedikt et al., 2017; Benedikt, Cuenca Grau, & Kostylev, 2018), we considered the setting where a conjunctive policy query over the source schema is used to specify sensitive information, and a privacy breach occurs whenever there is an answer to the policy query that holds in all the possible instantiations of the source schema compatible with the information visible to users. Similar settings are considered by Nash and Deutsch (2007), where they require that not only positive, but also negative information should not be disclosed, and by Chirkova and Yu (2017), where they assume that mappings are restricted to conjunctive views, there is no ontology, and the source schema comes equipped with a set of integrity constraints. Calvanese et al. (2012) extend the database authorisation framework by Zhang and Mendelzon (Zhang & Mendelzon, 2005) to the context of OBDA. In their framework, users are assigned a set of conjunctive authorisation views and each user query is then answered by the system using only the information that follows from the ontology and their respective views; in this setting, sensitive information is not explicitly represented—answers to user queries that do not follow from the ontology and the materialisation of the views over the data source are assumed to be sensitive by default.

6.3 Anonymisation Models

We finally discuss anonymisation models, which are the closest to our work.

We start by discussing our results as an extension of our prior conference publication (Cuenca Grau & Kostylev, 2016). The most significant addition is the study of the decision problems associated to cost minimisation. These problems are relevant for practice since the main goal of an anonymisation algorithm is to compute (rather than check) a most informative anonymisation satisfying the required properties. Another significant addition to our conference paper is the study of *strict suppressors*, which capture a very natural class of anonymisations where all occurrences of the same constant are mapped to the same null. In addition to considering new decision problems and types of suppressors, we also corrected certain mistakes concerning the complexity of SAFETY (see discussion of Lemmas 16 and 17) and strengthened a number of complexity bounds; in particular, several lower bounds are now proved under additional requirements (for instance, the requirement that datasets are ground, suppressors are strict, or closures are quantifier-free). We also generalised the

notions surrounding anonymisation with closed-world information. On the one hand, we now allow for several closures as opposed to just a single one as in our conference paper; this is a natural generalisation since the information considered to be complete is likely to involve rather distinct parts of a published graph. On the other hand, data complexity of the problems involving closures is also defined slightly differently: in the conference version both the query q and the known answers Ans of the closure $[q, Ans]$ were considered to be fixed for data complexity analysis, while we now consider only the query to be fixed (but not the answers); we believe that this is a more natural definition of data complexity and does not change our results with the only exception that certain lower bounds can be strengthened by considering only quantifier-free closures (e.g., see Lemma 4).

We now turn our attention to discussing k -anonymity: a popular technique for anonymising databases while providing protection against linkage attacks (Samarati & Sweeney, 1998; Sweeney, 2002). The input to the k -anonymity approach is a relational table T ; then, some of the entries (i.e., positions) in T are replaced with unlabelled nulls so that each tuple in the anonymised table has at least $k - 1$ corresponding tuples in T . In this setting, the cost is given by the number of entries in T replaced by nulls, and the goal is to find a k -anonymisation of minimal cost. The underpinning decision problem was shown NP-hard for $k \geq 3$ by Meyerson and Williams (2004) and tractable for $k = 2$ by Blocki and Williams (2010). Practical algorithms were proposed by Bayardo and Agrawal (2005). k -Anonymity has been generalised to handle multiple relations in a database (Nergiz, Clifton, & Nergiz, 2009), and to apply only to given sets of attributes in a relation (Wang & Fung, 2006). Finally, k -anonymity has also been refined to take into account probabilistic bounds on the attacker’s confidence on inferring a sensitive value (Machanavajjhala et al., 2007; Wang & Fung, 2006; Wong, Li, Fu, & Wang, 2006). The direct application of k -anonymity to RDF, where a graph corresponds to a single table with three attributes, is however of rather limited use in practice. For instance, the only 2-anonymisation of our example graph G_0 in Section 3 is the trivial one where all IRIs are replaced by fresh nulls. Consequently, our notions of compliance and safety, which utilise named nulls, provide a much more fine-grained control over the information to be anonymised than k -anonymity since both policies and closed-world requirements can be described by CQs.

We conclude this section by mentioning that there has also been a considerable recent interest in *graph anonymisation* techniques for social networks, where the goal is to ensure privacy while preserving the global network properties for analysis. Backstrom et al. (2007), however, showed that the graph’s structure can reveal individual identities, even if all node identifiers have been anonymised. To address this problem, Hay et al. (2010) propose the notion of k -candidate anonymity where the requirement is to modify the original anonymised graph via edge additions or deletions until all nodes have the same degree of at least $k - 1$ other nodes. Similar notions were studied by Liu and Terzi (2008) and Zhou and Pei (2008). Note, however, that the application of these techniques to publishing RDF graphs is of limited use as they involve anonymising all node identifiers in a graph as a first step.

7. Conclusion and Future Work

We have proposed and studied reasoning problems designed to ensure that anonymised RDF graphs can be published on the Semantic Web with provable privacy guarantees.

The problem of RDF anonymisation remains rather unexplored and we see many avenues for future work.

1. As mentioned in the relevant places throughout the paper, our complexity lower bounds could be tightened in a number of ways. For instance, it would be interesting to study whether our lower bounds for the cost minimisation problems associated to policy compliance and linkage safety still hold if we additionally assume the closures to be singleton, quantifier-free, or atomic.
2. Our current combined complexity bounds for the problems associated to linkage safety under the open-world assumption are not tight. Closing the existing gaps may require as a first step determining the precise complexity of the critical tuple problem in databases—a very interesting and challenging problem for future work.
3. The adoption of the UNA has no influence on the proofs for all our open-world results as well as all our results on policy compliance. However, all our lower and upper bounds for SAFETY, MIN-SAFETY, and MIN-SAFETY_s (i.e., Lemmas 12, 13, 16, 17, 18 (part 2), 23, 24, and 25) critically rely on the UNA; the precise complexity of these problems in the case when the UNA is not adopted remains open.
4. Our framework does not yet capture OWL 2 ontologies, which are used in many applications to describe the meaning of RDF graphs. We anticipate that the introduction of ontologies into the picture will lead to significant technical challenges, especially in combination with closed-world information; an interesting starting point to address these challenges is the recent work by Ngo, Ortiz, and Simkus (2016) and Seylan, Franconi, and de Bruijn (2009).
5. Our decidability results open the door to the future design of practical anonymisation algorithms. Although most of the problems we considered are intractable in both combined and data complexity, anonymisation in data publishing often constitutes an offline process that is only performed once for each data release.

Acknowledgements

This work was funded by the EPSRC projects DBOnto and ED3, as well as by the Royal Society under a University Research Fellowship.

References

- Abel, F., De Coi, J. L., Henze, N., Koesling, A., Krause, D., & Olmedilla, D. (2007). Enabling advanced and context-dependent access control in RDF stores. In *Proceedings of the 6th International Semantic Web Conference and the 2nd Asian Semantic Web Conference (ISWC/ASWC 2007)*, Vol. 4825 of *Lecture Notes in Computer Science*, pp. 1–14. Springer.
- Abiteboul, S., Hull, R., & Vianu, V. (1995). *Foundations of databases*. Addison-Wesley.
- Ahmetaj, S., Ortiz, M., & Simkus, M. (2016). Polynomial Datalog rewritings for expressive description logics with closed predicates. In *Proceedings of the 25th International*

- Joint Conference on Artificial Intelligence (IJCAI 2016)*, pp. 878–885. IJCAI/AAAI Press.
- Aron, Y. (2013). Information privacy for linked data. MSc Thesis.
- Backstrom, L., Dwork, C., & Kleinberg, J. M. (2007). Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web (WWW 2007)*, pp. 181–190. ACM.
- Bayardo, R. J., & Agrawal, R. (2005). Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering, (ICDE 2005)*, pp. 217–228. IEEE Computer Society.
- Benedikt, M., Bourhis, P., ten Cate, B., & Puppis, G. (2016). Querying visible and invisible information. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2016)*, pp. 297–306.
- Benedikt, M., Cuenca Grau, B., & Kostylev, E. V. (2017). Source information disclosure in ontology-based data integration. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI 2017)*, pp. 1056–1062. AAAI Press.
- Benedikt, M., Cuenca Grau, B., & Kostylev, E. V. (2018). Logical foundations of information disclosure in ontology-based data integration. *Artificial Intelligence Journal (AIJ)*, 262, 52–95.
- Biskup, J., & Bonatti, P. (2004). Controlled query evaluation for enforcing confidentiality in complete information systems. *International Journal of Information Security (IJIS)*, 3(1), 14–27.
- Biskup, J., & Weibert, T. (2008). Keeping secrets in incomplete databases. *International Journal of Information Security (IJIS)*, 7(3), 199–217.
- Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked Data - the story so far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3), 1–22.
- Blocki, J., & Williams, R. (2010). Resolving the complexity of some data privacy problems. In *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010), Part II*, Vol. 6199 of *Lecture Notes in Computer Science*, pp. 393–404. Springer.
- Bonatti, P., Kraus, S., & Subrahmanian, V. S. (1995). Foundations of secure deductive databases. *IEEE Transactions on Data and Knowledge Engineering (TKDE)*, 7(3), 406–422.
- Bonatti, P. A., & Sauro, L. (2013). A confidentiality model for ontologies. In *Proceedings of the 12th International Semantic Web Conference (ISWC 2013)*, Vol. 8218 of *Lecture Notes in Computer Science*, pp. 17–32. Springer.
- Calvanese, D., De Giacomo, G., Lenzerini, M., & Rosati, R. (2012). View-based query answering in description logics: Semantics and complexity. *Journal of Computer and System Sciences (JCSS)*, 78(1), 26–46.

- Chawla, S., Dwork, C., McSherry, F., Smith, A. D., & Wee, H. (2005). Toward privacy in public databases. In *Proceedings of the 2nd Theory of Cryptography Conference (TCC 2005)*, Vol. 3378 of *Lecture Notes in Computer Science*, pp. 363–385. Springer.
- Chirkova, R., & Yu, T. (2017). Exact detection of information leakage: Decidability and complexity. *LNCS Theory Large-Scale Data- and Knowledge-Centered Systems (T-LSD-KCS)*, 32, 1–23.
- Cuenca Grau, B., Kharlamov, E., Kostylev, E. V., & Zheleznyakov, D. (2015). Controlled query evaluation for Datalog and OWL 2 profile ontologies. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 2883–2889. AAAI Press.
- Cuenca Grau, B., & Kostylev, E. V. (2016). Logical foundations of privacy-preserving publishing of Linked Data. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, pp. 943–949. AAAI Press.
- Dwork, C. (2006). Differential privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP 2006)*, Part II, Vol. 4052 of *Lecture Notes in Computer Science*, pp. 1–12. Springer.
- Dwork, C. (2008). Differential privacy: A survey of results. In *Proceedings of the 5th International Conference Theory and Applications of Models of Computation (TAMC 2008)*, pp. 1–19.
- Dwork, C., Naor, M., Reingold, O., Rothblum, G. N., & Vadhan, S. P. (2009). On the complexity of differentially private data release: efficient algorithms and hardness results. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC 2009)*, pp. 381–390.
- Flouris, G., Fundulaki, I., Michou, M., & Antoniou, G. (2010). Controlling access to RDF graphs. In *Proceedings of the 3rd Future Internet Symposium (FIS 2010)*, Vol. 6369 of *Lecture Notes in Computer Science*, pp. 107–117. Springer.
- Fung, B. C. M., Wang, K., Chen, R., & Yu, P. S. (2010). Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 42(4), 14:1–14:53.
- Harris, S., & Seaborne, A. (2013). SPARQL 1.1 query language. W3C Recommendation.
- Hay, M., Miklau, G., Jensen, D. D., Towsley, D. F., & Li, C. (2010). Resisting structural re-identification in anonymized social networks. *The International Journal on Very Large Data Bases (VLDBJ)*, 19(6), 797–823.
- Hayes, P. (2004). RDF Semantics. W3C Recommendation.
- Hogan, A., Arenas, M., Mallea, A., & Polleres, A. (2014). Everything you always wanted to know about blank nodes. *Web Semantics: Science, Services and Agents on the World Wide Web (JWS)*, 27–28, 42–69.
- Immerman, N. (1987). Expressibility as a complexity measure: results and directions. In *Proceedings of the Second Annual Conference on Structure in Complexity Theory (CoCo 1987)*, pp. 797–823. IEEE Computer Society.
- Kagal, L., & Pato, J. (2010). Preserving privacy based on semantic policy tools. *IEEE Security & Privacy*, 8(4), 25–30.

- Kirrane, S., Abdelrahman, A., Mileo, A., & Decker, S. (2013). Secure manipulation of Linked Data. In *Proceedings of the 12th International Semantic Web Conference (ISWC 2013), Part I*, Vol. 8218 of *Lecture Notes in Computer Science*, pp. 248–263. Springer.
- Kostylev, E. V., & Suciu, D. (2018). A note on the hardness of the critical tuple problem. arXiv:1804.00443 [cs.DB].
- Lewis, J. M., & Yannakakis, M. (1980). The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences (JCSS)*, 20(2), 219–230.
- Liu, K., & Terzi, E. (2008). Towards identity anonymization on graphs. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 2008)*, pp. 93–106. ACM.
- Lutz, C., Seylan, I., & Wolter, F. (2015). Ontology-mediated queries with closed predicates. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 3120–3126. AAAI Press.
- Machanavajjhala, A., Kifer, D., Gehrke, J., & Venkitasubramaniam, M. (2007). *L*-diversity: Privacy beyond *k*-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1).
- Manola, F., & Miller, E. (2004). RDF Primer. W3C Recommendation.
- Meyerson, A., & Williams, R. (2004). On the complexity of optimal *k*-anonymity. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 2004)*, pp. 223–228. ACM.
- Miklau, G., & Suciu, D. (2007). A formal analysis of information disclosure in data exchange. *Journal of Computer and System Sciences (JCSS)*, 73(3), 507–534.
- Nash, A., & Deutsch, A. (2007). Privacy in GLAV information integration. In *Proceedings of the 11th International Conference on Database Theory (ICDT 2007)*, pp. 89–103.
- Nergiz, M. E., Clifton, C., & Nergiz, A. E. (2009). Multirelational *k*-anonymity. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 21(8), 1104–1117.
- Ngo, N., Ortiz, M., & Simkus, M. (2016). Closed predicates in description logics: Results on combined complexity. In *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning (KR 2016)*, pp. 237–246. AAAI Press.
- Papadimitriou, C. H. (1994). *Computational complexity*. Addison-Wesley.
- Reiter, R. (1992). What should a database know?. *The Journal of Logic Programming (JLP)*, 14(1–2), 127–153.
- Samarati, P. (2001). Protecting respondents’ identities in microdata release. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(6), 1010–1027.
- Samarati, P., & Sweeney, L. (1998). Generalizing data to provide anonymity when disclosing information (abstract). In *Proceedings of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS 1998)*, p. 188. ACM.

- Seylan, I., Franconi, E., & de Bruijn, J. (2009). Effective query rewriting with ontologies over DBoxes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pp. 923–929.
- Sicherman, G. L., de Jonge, W., & van de Riet, R. P. (1983). Answering queries without revealing secrets. *ACM Transactions on Database Systems (TODS)*, 8(1), 41–59.
- Studer, T., & Werner, J. (2014). Censors for Boolean description logic. *Transactions on Data Privacy (TDP)*, 7(3), 223–252.
- Sweeney, L. (2002). k-Anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5), 557–570.
- Wang, K., & Fung, B. C. M. (2006). Anonymizing sequential releases. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pp. 414–423. ACM.
- Wong, R. C., Li, J., Fu, A. W., & Wang, K. (2006). (α, k) -Anonymity: an enhanced k-anonymity model for privacy preserving data publishing. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pp. 754–759. ACM.
- Yannakakis, M. (1978). Node- and edge-deletion NP-complete problems. In *Proceedings of the 10th Annual ACM Symposium on Theory of Computing (STOC 1978)*, pp. 253–264.
- Zhang, Z., & Mendelzon, A. O. (2005). Authorization views and conditional query containment. In *Proceedings of the 10th International Conference on Database Theory (ICDT 2005)*, pp. 259–273.
- Zhou, B., & Pei, J. (2008). Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 24th International Conference on Data Engineering (ICDE 2008)*, pp. 506–515.