A Declarative Approach to Linked Data Anonymization
Companion appendix

Remy Delanaux[1], Angela Bonifati[1], Marie-Christine Rousset[2,3], and Romuald Thion[1]

[1] Université Lyon 1, LIRIS CNRS, 69100 Villeurbanne, France
[name].[surname]@univ-lyon1.fr
[2] Université Grenoble Alpes, CNRS, INRIA, Grenoble INP, 38000 Grenoble, France
[name].[surname]@imag.fr
[3] Institut Universitaire de France, 75000 Paris, France

1 Full proofs

For the sake of conciseness we write $Q = \langle \bar{x}, G \rangle$ for the query `SELECT` $\bar{x}$ `WHERE` $G(\bar{x}, \bar{y})$. Similarly, we write $\mathsf{delete}(H, W)$ for the function to the deletion query `DELETE` H `WHERE` W, that is

$$\mathsf{delete}(H, W) = \lambda DB.\, \mathsf{Result}(\texttt{DELETE}\ H\ \texttt{WHERE}\ W, DB))$$

Lemma 1 (BGP queries are monotonic). *Let $Q_1 = \langle \bar{x}, G_1 \rangle$ and $Q_2 = \langle \bar{x}, G_2 \rangle$ be two queries (with identical heads) and $Q_1 \subseteq Q_2$, then for all DB and DB' such that $DB \subseteq DB'$, it is the case that $\mathsf{Ans}(Q_2, DB) \subseteq \mathsf{Ans}(Q_1, DB')$.*

Proof. Writing $\iota : Q_1 \hookrightarrow Q_2$ and $\iota' : DB \hookrightarrow DB'$ the inclusion morphisms, any morphism $\mu : Q_2 \hookrightarrow DB$ can be extended to a morphism $\iota' \circ \mu \circ \iota : Q_1 \hookrightarrow DB'$ which is identical to μ on Q_1's variables.

We now provide a slightly extended version of the main Algorithm where H is not a renaming of G^P but any of subset with a morphism $\eta : G^P \hookrightarrow H$. Indeed, there is no need to traverse all G^P but only an H such that $\mathsf{Core}(G^P) \subseteq H \subseteq G^P$.

Algorithm 1: Find delete operations to satisfy a unitary privacy policy

Input : a unitary privacy policy $\mathcal{P} = \{P\}$ with $P = \langle \bar{x}^P, G^P \rangle$
Input : a utility policy $\mathcal{U}$ made of m queries $U_j = \langle \bar{x}^U_j, G^U_j \rangle$
Output: a set of operations *ops* satisfying both $\mathcal{P}$ and $\mathcal{U}$

```
function find-ops-unit(P,U):
    Let H ⊆ G'^P with an additional η : G'^P ↪ H where G'^P is a renaming of G^P;
    Let ops := ∅;
    forall (s,p,o) ∈ H do
        Let c := true;
        forall G^U_j do
            forall (s',p',o') ∈ G^U_j do
                if ∃σ such that σ(s,p,o) = σ(s',p',o') then
                    c := false;
                end
            end
        end
        if c then
            ops := ops ∪ {DELETE {(s,p,o)} WHERE H};
        end
    end
    return ops;
end
```

Lemma 2 (Boolean satisfiability). *Let $Q = \langle \bar{x}, G \rangle$ be a query, let $DB \in \mathbf{BGP}$ be a graph and let H be a subset of G together with a morphism $\eta : G \hookrightarrow H$, then* $\mathsf{Ans}(\langle \bar{x}, G \rangle, DB) = \emptyset$ if and only if $\mathsf{Ans}(\langle \langle\rangle, H \rangle, DB) = \emptyset$

Proof. Let us denote the inclusion $H \subseteq G$ by its canonical inclusion morphism $\iota : H \hookrightarrow G$. We prove the *only if* direction by contraposition. Assume that there is an answer in $\mathsf{Ans}(\langle \langle\rangle, H \rangle, DB)$. By the definition of Ans, there is at least one morphism $\mu : H \hookrightarrow DB$. By composing μ and η we obtain a morphism $\mu \circ \eta : G \hookrightarrow DB$, thus $\mathsf{Ans}(\langle \bar{x}, G \rangle, DB)$ is not empty. We prove the *if* direction by contraposition similarly. Assume that there is an answer in $\mathsf{Ans}(\langle \bar{x}, G \rangle, DB)$ and call it $\nu : G \hookrightarrow DB$. By composing ν and ι we obtain a morphism from $\nu \circ \iota : H \hookrightarrow DB$, thus $\mathsf{Ans}(\langle \langle\rangle, H \rangle, DB)$ is not empty.

Lemma 3 (Soundness for privacy). *Let $Q = \langle \bar{x}, G \rangle$ be a query, let H be G renamed with fresh variables and $(s, p, o) \in H$. For all $DB \in \mathbf{BGP}$, the following equality holds:*

$$\mathsf{Ans}(\langle \bar{x}, G \rangle, \mathsf{Result}(\texttt{DELETE } \{(s,p,o)\} \texttt{ WHERE } H, DB)) = \emptyset$$

Proof. Let $DB' = \mathsf{delete}(\{(s,p,o)\}, H)(DB)$ the graph obtained after deletion. By Lemma 2, it is equivalent to prove that $\mathsf{Ans}(\langle \langle\rangle, H \rangle, DB') = \emptyset$ that is, to prove that there is no morphism $\nu : H \hookrightarrow DB'$. For the sake of contradiction, assume that such a ν exists. Let's consider the triple $\nu(s,p,o) \in DB'$. On the other hand, $DB' = DB \setminus \{\mu(s,p,o) \mid \mu : H \hookrightarrow DB\}$ by the definition of delete, but picking $\mu = \nu$ shows that $\nu(s,p,o) \notin DB'$, a contradiction.

Theorem 1 (Correction of Algorithm `find-ops-unit`). *Let $P = \langle \bar{x}^P, G^P \rangle$ be a query and let $\mathcal{U} = \{U_j\}$ be a set of m queries $U_j = \langle \bar{x}_j^U, G_j^U \rangle$. Let $O =$`find-ops-unit(`P,$\mathcal{U}$`)`. For all $o_k \in O$, for all $DB \in \mathbf{BGP}$, it is the case that $\mathsf{Ans}(P, o_k(DB)) = \emptyset$ and $\mathsf{Ans}(U_j, o_k(DB)) = \mathsf{Ans}(U_j, DB)$ for all $U_j \in \mathbf{U}$, in other words, both P and $\mathcal{U}$ are satisfied by each operation o_k.*

Proof. The privacy query P is satisfied because the delete operation created at Line 14 of Algorithm 1 is of the form required by Lemma 3 for all choice of $(s,p,o) \in H$ made in the main loop at Line 4. So the proof amounts to check that all U_j are satisfied, i.e., that $\mathsf{Ans}(G_j^U, o_k(DB)) = \mathsf{Ans}(G_j^U, DB)$ for all $U_j \in \mathbf{U}$. One inclusion is clear by the monotonicity of BGP queries (Lemma 1) because $o_k(DB) \subseteq DB$, thus the end of this proof is to show that $\mathsf{Ans}(G_j^U, DB) \subseteq \mathsf{Ans}(G_j^U, o_k(DB))$ for all G_j^U.

Let $j \in [1..m]$ and $a \in \mathsf{Ans}(G_j^U, DB)$ an answer of G_j^U on DB. By definition of Ans, $a = \mu(\bar{x}_j^U)$ for some $\mu : G_j^U \hookrightarrow DB$, we show that μ is a morphism into $o_k(DB)$ as well so $a \in \mathsf{Ans}(G_j^U, o_k(DB))$ and the proof is complete.

Let consider $t' = (s', p', o') \in G_j^U$, for the sake of contradiction, assume that $\mu(t') \notin o_k(DB)$, that is $\mu(t') \in DB \setminus o_k(DB)$. By construction in Algorithm 1 and by the definition of the delete operation $DB \setminus o_k(DB) = DB \setminus \mathsf{delete}(\{(s,p,o)\}, H)(DB) = DB \setminus DB \setminus (\bigcup\{\nu(s,p,o) \mid \nu : H \hookrightarrow DB\}) = (\bigcup\{\nu(s,p,o) \mid \nu : H \hookrightarrow DB\})$. Thus $\mu(t') \in DB \setminus o_k(DB)$ implies that $\mu(t') = (\nu)(t)$ for some $t = (s,p,o) \in H$ and $\nu : H \hookrightarrow DB$. As μ and ν have distinct domains thanks to the renaming of G^P, they can be combined into the morphism σ such that $\sigma(t') = \sigma(t)$ defined by $\sigma(v) = \mu(v)$ when $v \in \mathsf{dom}(\mu)$, $\sigma(v) = \nu(v)$ when $v \in \mathsf{dom}(\nu)$ and $\sigma(v) = v$ otherwise. But this is precisely the condition at Line 8 so $o_k \notin O$. We obtained the desired contradiction so $a \in \mathsf{Ans}(U_j^U, o_k(DB))$ and the proof is complete.

Algorithm 2: Find delete operations to satisfy policies

Input : a privacy policy $\mathcal{P}$ made of n queries $P_i = \langle \bar{x}_i^P, G_i^P \rangle$
Input : a utility policy $\mathcal{U}$ made of m queries $U_j = \langle \bar{x}_j^U, G_j^U \rangle$
Output: a set of sets of operations Ops such that each sequence obtained from ordering any $O \in Ops$ satisfies both $\mathcal{P}$ and $\mathcal{U}$

```
1 function find-ops(P,U):
2     Let Ops = {∅};
3     for P_i ∈ P do
4         Let ops_i :=find-ops-unit(P_i,U);
5         Ops := {O ∪ {o'} | O ∈ Ops ∧ o' ∈ ops_i};
6     end
7     return Ops;
8 end
```

Theorem 2 (Correction of Algorithm `find-ops`). *Let $\mathcal{P}$ be a privacy policy made of n queries $P_i = \langle \bar{x}_i^P, G_i^P \rangle$ and let $\mathcal{U}$ be a utility policy made of m queries $U_j = \langle \bar{x}_j^U, G_j^U \rangle$ Let $\mathcal{O} =$ `find-ops`$(\mathcal{P},\mathcal{U})$ and DB an RDF graph. For any set of operations $O_k \in \mathcal{O}$, and for any ordering S_k of O_k, $\forall P_i \in \mathcal{P}, \mathsf{Ans}(P_i, S_k(G)) = \emptyset$ and $\forall U_j \in \mathcal{U}, \mathsf{Ans}(U_j, G) = \mathsf{Ans}(U_j, S_k(G))$, that is both $\mathcal{P}$ and $\mathcal{U}$ are satisfied by each sequence S_k.*

Proof. First of all let us note that O_k is either $\emptyset$ when some ops_i is empty or it is of the form $O_k = \{o_1, \ldots, o_n\}$ with $n = |\mathbf{P}|$. Indeed, the loop at Line 3 is executed once for each P_i, so at line 5, either one ops_i is empty and thus $Ops = \emptyset$ because $\{O \cup \{o'\} \mid O \in Ops \wedge o' \in \emptyset\} = \emptyset$, or all $ops_i \neq \emptyset$ an each $O_k \in Ops$ contains exactly one operation for each P_i.

By construction of Algorithm 2 and by Theorem 1 each $o \in O_k$ satisfies at least one of the P_i and all U_j and each P_i is satisfied by at least one $o \in O_k$. Thus any choice of an ordering S_k of O_k is such that all P_i are satisfied.