# Query-based 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 Example 5 complete results

This is the full list of operation sequences found using Algorithm 2, distributing operations in $O_1$ and $O_2$:

$\mathcal{O} = \{\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P, \texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(?u, \texttt{vcard:hasAddress}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(?u, \texttt{vcard:hasAddress}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{tcl:user}, ?u)\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(?c, \texttt{tcl:user}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(?u, \texttt{vcard:hasAddress}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(?c, \texttt{tcl:user}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_2^P\},$

$\{\texttt{DELETE}\ \{(?u, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{INSERT}\ \{(\texttt{[ ]}, \texttt{vcard:hasAddress}, ?ad)\}\ \texttt{WHERE}\ G_1^P,$
$\texttt{DELETE}\ \{(?c, \texttt{tcl:user}, ?u)\}\ \texttt{INSERT}\ \{(?c, \texttt{tcl:user}, \texttt{[ ]})\}\ \texttt{WHERE}\ G_2^P\},\}$

## 2 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 $\mathsf{update}(H, I, W)$ for the function to the update query DELETE $H$ INSERT $I$ WHERE $W$ and $\mathsf{delete}(H, W)$ for the function of the deletion query DELETE $H$ WHERE $W$, that is:

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

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

**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 $\mu$ 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 $O$ satisfying both $\mathcal{P}$ and $\mathcal{U}$

**function** `find-ops-unit(`$P$,$\mathcal{U}$`)`**:**
    Let $H \subseteq G'^P$ with an additional $\eta : G'^P \hookrightarrow H$ where $G'^P$ is a renaming of $G^P$;
    Let $O := \emptyset$;
    **forall** $(s, p, o) \in H$ **do**
        Let $c :=$ true;
        **forall** $G^U_j$ **do**
            **forall** $(s', p', o') \in G^U_j$ **do**
                **if** $\exists \sigma \ (\sigma(s', p', o') = \sigma(s, p, o))$ **then**
                    $c :=$ false;
        **if** $c$ **then**
            $O := O \cup \{$`DELETE` $\{(s, p, o)\}$ `WHERE` $H\}$;
            **if** `check-subject`$((s, p, o), H) \ \vee \ s \in \bar{x}^P$ **then**
                $O := O \cup \{$`DELETE` $\{(s, p, o)\}$ `INSERT` $\{($`[]`$, p, o)\}$ `WHERE` $H\}$;
            **if** $o \in \mathbf{I} \ \wedge$ (`check-object`$((s, p, o), H) \ \vee \ o \in \bar{x}^P$) **then**
                $O := O \cup \{$`DELETE` $\{(s, p, o)\}$ `INSERT` $\{(s, p,$ `[]`$)\}$ `WHERE` $H\}$;
    **return** ops;

**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 $\mathsf{Ans}$, there is at least one morphism $\mu : H \hookrightarrow DB$. By composing $\mu$ and $\eta$ 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 $\nu$ and $\iota$ 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 update queries satisfy privacy policy $\mathcal{P} = \{Q\}$:*

$$\texttt{DELETE } \{(s, p, o)\} \texttt{ WHERE } H, DB)$$

$$\texttt{DELETE } \{(s, p, o)\} \texttt{ INSERT } \{(x_u, p, o)\} \texttt{ WHERE } H, DB)$$

$$\texttt{DELETE}\ \{(s,p,o)\}\ \texttt{INSERT}\ \{(s,p,x_u)\}\ \texttt{WHERE}\ H, DB)$$

*where $x_u \in \mathbf{B}$ a fresh blank node (equivalent to the `[]` convention used in the main article).*

*Proof.* Let's consider the three possible cases.

**First query:** By Lemma 2 and by definition of query answers and privacy policy satisfiability, 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 $\nu$ exists. Let $DB' = \mathsf{delete}(\{(s,p,o)\}, H)(DB)$ the graph obtained after deletion. 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 $\mathsf{delete}$, but picking $\mu = \nu$ shows that $\nu(s,p,o) \notin DB'$, a contradiction.

**Second query:** Let $DB' = \mathsf{update}(\{(s,p,o)\}, \{x_u,p,o)\}, H)(DB)$ the graph obtained after subject update. Three possibles cases can trigger this operation.

- **Case 1:** $\exists (s',p',s) \in H$

Let $a \in Ans(\langle\langle\rangle, H\rangle, DB')$ an answer on $DB'$, so that $\exists \mu \mid \mu(H) \subseteq DB'$. In particular, this applies to the subgraph $\bar{H} = \{(s,p,o),(s',p',s)\}$, and we have $\mu(\bar{H}) \subseteq DB$, which is equivalent to $\{(\mu s, \mu p, \mu o), (\mu s', \mu p', \mu s)\} \subseteq DB'$.

But by the definition of $\mathsf{update}$, $DB' = DB \setminus \{\nu(s,p,o) | \nu : H \hookrightarrow DB\} \cup \{\nu(x_u,p,o) | \nu : H \hookrightarrow DB\}$, as we replace every subject of matching triples $(s,p,o)$ by a fresh blank node. Therefore with $\mu = \nu$, we have $\mu s = b \in \mathbf{B}$, a fresh blank node.

We would then have $\mu(\bar{H}) = \{(b, \mu p, \mu o), (\mu s', \mu p', b)\}$, which is not possible since $b$ is by construction created as a fresh variable in each insertion and cannot be found in two different triples: there is a contradiction and $a$ cannot exist. With this operation and this condition $Ans(\langle\langle\rangle, H\rangle, DB') = \emptyset$ and the privacy is fulfilled.

- **Case 2:** $\exists (s,p',o') \in H$ **and** $\nexists \sigma\ (\sigma(s,p',o') = \sigma(s,p,o))$

We apply the same methodology as in Case 1: Let $a \in Ans(\langle\langle\rangle, H\rangle, DB')$ an answer, and let a subgraph $\bar{H} = \{(s,p,o),(s,p',o')\}$, and we then have $\mu(\bar{H}) = \{(\mu s, \mu p, \mu o), (\mu s, \mu p', \mu o')\} \subseteq DB'$. By construction of $\mathsf{update}$, we have $\mu s = b \in \mathbf{B}$, a fresh blank node. We would then have $\mu(\bar{H}) = \{(b, \mu p, \mu o), (b, \mu p', \mu o')$. Plus, by hypothesis, $(s,p,o)$ and $(s,p',o')$ are not unifiable. Therefore, such a case is not possible and $Ans(\langle\langle\rangle, H\rangle, DB')$ must be empty. The privacy condition is satisfied.

- **Case 3:** $s \in \bar{x}^P$

Let's consider an answer $\mathsf{Ans}(\langle\bar{x}, H\rangle, DB')$. By definition of $\mathsf{update}$, $DB' = DB \setminus \{\mu(s,p,o) | \mu : H \hookrightarrow DB\} \cup \{\mu(x_u,p,o) | \mu : H \hookrightarrow DB\}$, as we replace every subject of matching triples $(s,p,o)$ by a fresh blank node $x_u$. By hypothesis, $s \in \bar{x}^P$, therefore $\forall a \in \mathsf{Ans}(\langle\bar{x}, H\rangle, DB'), mus \in a$, that is $\exists x_u \in \mathbf{B} \mid x_u \subseteq a$. Which entails that for any tuple full of constants $\bar{c}$, $\bar{c} \notin \mathsf{Ans}(\langle\bar{x}, H\rangle, DB')$, which means that the privacy is ensured.

**Third equality:** Let $DB' = \mathsf{update}(\{(s,p,o)\}, \{s,p,x_u)\}, H)(DB)$ the graph obtained after value update. Let's consider the triple $\nu(s,p,o) \in DB'$.

We consider the same 3 cases as the second equality and show using the same rules that:

- In the first case ($\exists (o,p',o') \in H$), an answer to the query would mean that $\mu(\bar{H}) = \{(s, \mu p, b), (b, \mu p, o)\}$ with $b \in \mathbf{B}$ a fresh blank node, which is not possible.
- In the second case $((\exists (s',p',o) \in H$ **and** $\nexists \sigma\ (\sigma(s',p',o) = \sigma(s,p,o))$, an answer would imply $\mu(\bar{H}) = \{(\mu s, \mu p, b), (\mu s', \mu p', b)$ with $b \in \mathbf{B}$ a fresh blank node, which is not possible.
- In the third case ($o \in \bar{x}^P$), we have $\forall a \in \mathsf{Ans}(\langle\bar{x}, H\rangle, DB'), \mu o \in a$, that is $\exists b \in \mathbf{B} \in a$.

**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 $\forall t, t \in \mathsf{Ans}(P, o_k(DB)) \Rightarrow$ `hasBlank(t)` 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 each operation created at Lines 11,13 and 15 of Algorithm 1 is of a form covered by Lemma 3 for all choice of $(s,p,o) \in H$ made in the main loop at Line 4.

Next, we 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 $U_j \in \mathbf{U}$.

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 $\mathsf{Ans}$, $a = \mu(\bar{x}_j^U)$ for some $\mu : G_j^U \hookrightarrow DB$, we show that $\mu$ 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.

We now have to show that $\mathsf{Ans}(G_j^U, o_k(DB)) \subseteq \mathsf{Ans}(G_j^U, DB)$ We explore the three possibilities given by Lines 11,13 and 15 of the algorithm to be applied as $o_k$.

**Line 11:** 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 $\mathsf{delete}$ 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 $\mu$ and $\nu$ have distinct domains thanks to the renaming of $G^P$, they can be combined into the morphism $\sigma$ 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.

**Line 13:**

Let $j \in [1..m]$ and $a \in \mathsf{Ans}(G_j^U, o_k(DB))$ an answer of $G_j^U$ on $o_k(DB)$. By definition of $\mathsf{Ans}$, $a = \mu(\bar{x}_j^U)$ for some $\mu : G_j^U \hookrightarrow DB$ and by definition of $\mathsf{update}$, we have $\mu(G_j^U) \subseteq DB \setminus \mathsf{update}(\{(s,p,o)\}, \{(x_u,p,o)\}, H)(DB)$. We can write this as $\mu(G_j^U) \subseteq DB \setminus d(DB) \cup i(DB)$ where $d$ is the $d(DB)$ and $i(DB)$ are two graphs, respectively consisting in triples deleted from $DB$ and added to $DB$ by the $\mathsf{update}$ operation. $a \in \mathsf{Ans}(G_j^U, DB)$ would imply that $\mu(G_j^U) \cap i(DB) = \emptyset$.

Let $t$ a triple $(s,p,o)$ such that $t \in \mu(G_j^U)$ and $t \in i(DB)$. The first criteria gives: $\exists t^U \in G^U \mid t = \mu(t^U)$. Additionally, $t \in i(DB)$ implies $\exists t^P \in H, \exists \mu^P, \exists s \mid \mu^P(t^P) = t = (s,p,o)$. We write $t^U$ as $(s^U, p^U, o^U)$ and $t^P$ as $(s^P, p^P, o^P)$. Therefore, the following equalities must hold:

$$\mu(s^U) = x_u,\ \mu(p^U) = p,\ \mu(o^U) = o$$

$$\mu^P(s^P) = s,\ \mu^P(p^P) = p,\ \mu^P(o^P) = o$$

We have a contradiction, since $\mu(t^U) \neq \mu^P(t^P)$ while by hypothesis, $\mu(t^U) = \mu^P(t^P) = t$. Therefore this indeed proves that $\mu(G_j^U) \cap i(DB) = \emptyset$, and by deduction that $\mathsf{Ans}(G_j^U, o_k(DB)) \subseteq \mathsf{Ans}(G_j^U, DB)$.

**Line 15:**

We apply the same reasoning as the Line 13 proof, showing that in this case we would obtain $\mu(t^U) = (s,p,x_u)$ by definition of the $\mathsf{update}$ operation used at this line, while $\mu^P(t^P) = (s,p,o)$.

This proves again that $\mu(G_j^U) \cap i(DB) = \emptyset$, and therefore that $\mathsf{Ans}(G_j^U, o_k(DB)) \subseteq \mathsf{Ans}(G_j^U, DB)$.

---

**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 $\mathcal{U}$

**function** `find-ops(`$\mathcal{P}$,$\mathcal{U}$`)`**:**
    Let $Ops = \{\emptyset\}$;
    **for** $P_i \in \mathcal{P}$ **do**
        Let $ops_i$ :=`find-ops-unit(`$P_i$,$\mathcal{U}$`)`;
        $Ops := \{O \cup \{o'\} \mid O \in Ops \wedge o' \in ops_i\}$;
    **return** $Ops$;

---

**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.