Privacy-Preserving Ontology Publishing for $\mathcal{EL}$ Instance Stores

Franz Baader       Francesco Kriegel       Adrian Nuradiansyah

Technische Universität Dresden

Published in JELIA 2019 and Submitted to Künstliche Intelligenz (KI) 2019

August 20, 2019
Privacy-Preserving Ontology Publishing

Privacy policies

Ontology

compliant
Privacy-Preserving Ontology Publishing

Privacy policies

compliant

Ontology

compliant

Other sources
Privacy-Preserving Ontology Publishing

- Privacy policies
  - compliant
  - not compliant

Ontology

- integrated

Other sources

- integrated

Adrian Nuradiansyah
The ontology is compliant, but not safe w.r.t policies.
What people already did:

In \textit{(Cuenca Grau \& Kostylev, 2016)}:

- Privacy-Preserving Data Publishing
- Information to be published: a relational dataset with (labeled) nulls
- Policy is a conjunctive query.
- Considering three privacy properties when publishing datasets: \textit{policy-compliant, policy-safety, and optimality}.
- Published information does not have background knowledge.
What people already did:

In (Cuenca Grau & Kostylev, 2016):

- Privacy-Preserving Data Publishing
- Information to be published: a relational dataset with (labeled) nulls
- Policy is a conjunctive query.
- Considering three privacy properties when publishing datasets: policy-compliant, policy-safety, and optimality.
- Published information does not have background knowledge.

What we want to do:

- Privacy-Preserving Ontology Publishing (PPOP)
- Addressed in the context of Description Logic Ontologies
**Starting point:** $\mathcal{EL}$ Ontologies with **role-free ABoxes** (instance stores) and empty TBoxes.

An ABox $\mathcal{A}$ is **role-free** if all the axioms $\beta \in \mathcal{A}$ are only in the form of $D(a)$.

---

**Safe Ontologies**

Information to be published for an individual $a$: an $\mathcal{EL}$ concept $C_1(a), C_2(a) \in \mathcal{A}$ implies $(C_1 \sqcap C_2)(a) \in \mathcal{A}$

**Policy** is a finite set of $\mathcal{EL}$ concepts $D_1, \ldots, D_p$, such that $D_i \not\equiv \top$ for all $i \in \{1, \ldots, p\}$.

---

Adrianna Nuradiansyah
Starting point: $\mathcal{EL}$ Ontologies with role-free ABoxes (instance stores) and empty TBoxes.

An ABox $\mathcal{A}$ is role-free if all the axioms $\beta \in \mathcal{A}$ are only in the form of $D(a)$.

Why no TBox? For instance,

- in SNOMED CT $\rightarrow$ Acyclic TBox $\rightarrow$ the TBox can be reduced away
- Even in SNOMED, patient data are usually annotated with SNOMED concepts, not with SNOMED roles.
**Starting point:** $\mathcal{EL}$ Ontologies with **role-free ABoxes** (instance stores) and empty TBoxes.

An ABox $\mathcal{A}$ is **role-free** if all the axioms $\beta \in \mathcal{A}$ are only in the form of $D(a)$.

Why no TBox? For instance,

- in SNOMED CT $\rightarrow$ **Acyclic TBox** $\rightarrow$ the TBox can be reduced away
- Even in SNOMED, patient data are usually annotated with SNOMED concepts, not with SNOMED roles.

W.l.o.g., only **one concept assertion** in $\mathcal{A}$ speaks about one individual $C_1(a), C_2(a) \in \mathcal{A}$ implies $(C_1 \sqcap C_2)(a) \in \mathcal{A}$

**Safe Ontologies** $\xrightarrow{\text{reduced}}$ **Safe Concepts**
PPOP for \( \mathcal{EL} \) instance stores

- **Starting point**: \( \mathcal{EL} \) Ontologies with role-free ABoxes (instance stores) and empty TBoxes.

- An ABox \( \mathcal{A} \) is **role-free** if all the axioms \( \beta \in \mathcal{A} \) are only in the form of \( D(a) \).

- Why no TBox? For instance,
  - in SNOMED CT \( \rightarrow \) **Acyclic TBox** \( \rightarrow \) the TBox can be reduced away
  - Even in SNOMED, patient data are usually annotated with SNOMED concepts, not with SNOMED roles.

- W.l.o.g., only **one concept assertion** in \( \mathcal{A} \) speaks about one individual \( C_1(a), C_2(a) \in \mathcal{A} \) implies \( (C_1 \sqcap C_2)(a) \in \mathcal{A} \)

- Safe Ontologies \( \xrightarrow{reduced} \) Safe Concepts

- **Information to be published** for an individual \( a \): an \( \mathcal{EL} \) concept \( C \)

- **Policy** is a finite set of \( \mathcal{EL} \) concepts \( D_1, \ldots, D_p \), such that \( D_i \not\equiv \top \) for all \( i \in \{1, \ldots, p\} \).
Given a policy $\mathcal{P} = \{D_1, \ldots, D_p\}$ and an $\mathcal{E}\mathcal{L}$ concept $C$, the $\mathcal{E}\mathcal{L}$ concept $C'$ is

- **compliant** with $\mathcal{P}$ if $C' \not\sqsubseteq D_i$ for all $i \in \{1, \ldots, p\}$.

- **safe** for $\mathcal{P}$ if $C' \cap C''$ is compliant with $\mathcal{P}$ for all $\mathcal{E}\mathcal{L}$-concepts $C''$ that are compliant with $\mathcal{P}$.
Given a policy $\mathcal{P} = \{D_1, \ldots, D_p\}$ and an $\mathcal{E}\mathcal{L}$ concept $C$, the $\mathcal{E}\mathcal{L}$ concept $C'$ is

- **compliant** with $\mathcal{P}$ if $C' \not\subseteq D_i$ for all $i \in \{1, \ldots, p\}$.

- **safe** for $\mathcal{P}$ if $C' \cap C''$ is compliant with $\mathcal{P}$ for all $\mathcal{E}\mathcal{L}$-concepts $C''$ that are compliant with $\mathcal{P}$.

- a $\mathcal{P}$-compliant (safe) generalization of $C$ if
  - $C \subseteq C'$ and
  - $C'$ is compliant with (safe for) $\mathcal{P}$. 
Given a policy $\mathcal{P} = \{D_1, \ldots, D_p\}$ and an $\mathcal{EL}$ concept $C$, the $\mathcal{EL}$ concept $C'$ is

- **compliant** with $\mathcal{P}$ if $C' \not\sqsubseteq D_i$ for all $i \in \{1, \ldots, p\}$.

- **safe** for $\mathcal{P}$ if $C' \sqcap C''$ is compliant with $\mathcal{P}$ for all $\mathcal{EL}$-concepts $C''$ that are compliant with $\mathcal{P}$.

- a $\mathcal{P}$-compliant (safe) generalization of $C$ if
  - $C \sqsubseteq C'$ and
  - $C'$ is compliant with (safe for) $\mathcal{P}$.

- a $\mathcal{P}$-optimal compliant (safe) generalization of $C$ if
  - $C'$ is a $\mathcal{P}$-compliant (safe) generalization of $C$, and
  - there is no $\mathcal{P}$-compliant (safe) generalization $C''$ of $C$ s.t. $C'' \sqsubseteq C'$. 
Consider a policy $\mathcal{P} = \{D\}$ specifying what information should be kept “secret” about *linda*

\[ D = Patient \cap \exists \text{seen}_\text{by}.(\text{Doctor} \cap \exists \text{works}_\text{in}.\text{Cardiology}) \]

Assume information $C$ is published about *linda*

\[ C = Patient \cap Female \cap \exists \text{seen}_\text{by}.(\text{Doctor} \cap Male \cap \exists \text{works}_\text{in}.\text{Cardiology}) \]

Note $C$ is not compliant with $D$, i.e., $C \subseteq D$. 
Consider a **policy** $P = \{D\}$ specifying what information should be kept “secret” about *linda*

$$D = Patient \sqcap \exists \text{seen}_\text{by}. (Doctor \sqcap \exists \text{works}_\text{in}. \text{Cardiology})$$

Assume information $C$ is published about *linda*

$$C = Patient \sqcap Female \sqcap \exists \text{seen}_\text{by}. (Doctor \sqcap Male \sqcap \exists \text{works}_\text{in}. \text{Cardiology})$$

Note $C$ is not compliant with $D$, i.e., $C \subseteq D$.

Generalizing $C$ to yield a compliant concept

$$C_1 = Female \sqcap \exists \text{seen}_\text{by}. (Doctor \sqcap Male \sqcap \exists \text{works}_\text{in}. \text{Cardiology})$$

But, $C_1$ is **not safe for** $D$ since if the attacker knows $Patient(linda)$, then $C_1 \sqcap Patient \subseteq D$ is revealed.
Consider a **policy** $\mathcal{P} = \{D\}$ specifying what information should be kept “secret” about *linda*

$$D = Patient \sqcap \exists \text{seen}_by.(Doctor \sqcap \exists \text{works}_in.\text{Cardiology})$$

Assume information $C$ is published about *linda*

$$C = Patient \sqcap Female \sqcap \exists \text{seen}_by.(Doctor \sqcap Male \sqcap \exists \text{works}_in.\text{Cardiology})$$

Note $C$ is not compliant with $D$, i.e., $C \subseteq D$.

Let us **make it safe**!

$$C_2 = Female \sqcap \exists \text{seen}_by.(Doctor \sqcap Male \sqcap \exists \text{works}_in.\top)$$

But, $C_2$ is still not optimal since more information than necessary is removed.
Illustration on Compliance, Safety, and Optimality

- Consider a policy $\mathcal{P} = \{D\}$ specifying what information should be kept “secret” about *linda*

  $$D = \text{Patient} \land \exists \text{seen}_\text{by}.(\text{Doctor} \land \exists \text{works}_\text{in. Cardiology})$$

- Assume information $C$ is published about *linda*

  $$C = \text{Patient} \land \text{Female} \land \exists \text{seen}_\text{by}.(\text{Doctor} \land \text{Male} \land \exists \text{works}_\text{in. Cardiology})$$

  Note $C$ is not compliant with $D$, i.e., $C \not\subseteq D$.

- Let us make it safe!

  $$C_2 = \text{Female} \land \exists \text{seen}_\text{by}.(\text{Doctor} \land \text{Male} \land \exists \text{works}_\text{in. Cardiology} \land \top)$$

  But, $C_2$ is still not optimal since more information than necessary is removed.

- Make it optimal!

  $$C_3 = \text{Female} \land \exists \text{seen}_\text{by}.(\text{Doctor} \land \text{Male} \land \exists \text{works}_\text{in. Cardiology} \land \top) \land \exists \text{seen}_\text{by}.(\text{Male} \land \exists \text{works}_\text{in. Cardiology})$$
Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r.E$ occurring in the top-level conjunction of $C$. 

Complexity for Compliance

Deciding whether $C'$ is compliant w.r.t. $P$ is in $\text{PTime}$. One optimal $P$-compliant generalization can be computed in $\text{ExpTime}$. The set of all optimal $P$-compliant generalizations can be computed in $\text{ExpTime}$. 
Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r. E$ occurring in the top-level conjunction of $C$.

$\text{con}(C)$ covers $\text{con}(D)$ iff for all $F \in \text{con}(D)$, there is $E \in \text{con}(C)$ such that $E \sqsubseteq F$.
Characterizing Compliance

- Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r.E$ occurring in the top-level conjunction of $C$.

- $\text{con}(C)$ covers $\text{con}(D)$ iff for all $F \in \text{con}(D)$, there is $E \in \text{con}(C)$ such that $E \sqsubseteq F \Rightarrow$ Characterizing $C \sqsubseteq D$. 
Characterizing Compliance

- Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r. E$ occurring in the top-level conjunction of $C$.

- $\text{con}(C)$ covers $\text{con}(D)$ iff for all $F \in \text{con}(D)$, there is $E \in \text{con}(C)$ such that $E \sqsubseteq F \Rightarrow$ Characterizing $C \sqsubseteq D$.

### Compliance

$C$ is compliant with $\mathcal{P}$ iff $\text{con}(C)$ does not cover $\text{con}(D_i)$ for any $i \in \{1, \ldots, p\}$. 
Characterizing Compliance

- Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r. E$ occurring in the top-level conjunction of $C$.

- $\text{con}(C)$ covers $\text{con}(D)$ iff for all $F \in \text{con}(D)$, there is $E \in \text{con}(C)$ such that $E \preceq F \Rightarrow \text{Characterizing } C \preceq D$.

Compliance

$C$ is compliant with $\mathcal{P}$ iff $\text{con}(C)$ does not cover $\text{con}(D_i)$ for any $i \in \{1, \ldots, p\}$.

Complexity for Compliance

- Deciding whether $C'$ is compliant w.r.t. $\mathcal{P}$ is in $\text{PTime}$.
Characterizing Compliance

- Let $\text{con}(C)$ be the set of all atoms $A$ or $\exists r. E$ occurring in the top-level conjunction of $C$.

- $\text{con}(C)$ covers $\text{con}(D)$ iff for all $F \in \text{con}(D)$, there is $E \in \text{con}(C)$ such that $E \sqsubseteq F \Rightarrow \text{Characterizing } C \sqsubseteq D$.

Compliance

$C$ is compliant with $\mathcal{P}$ iff $\text{con}(C)$ does not cover $\text{con}(D_i)$ for any $i \in \{1, \ldots, p\}$.

Complexity for Compliance

- Deciding whether $C'$ is compliant w.r.t. $\mathcal{P}$ is in $\text{PTime}$.
- One optimal $\mathcal{P}$-compliant generalization can be computed in $\text{ExpTime}$.
- The set of all optimal $\mathcal{P}$-compliant generalizations can be computed in $\text{ExpTime}$.
Assume $\mathcal{P}$ is redundant-free: every $D_i, D_j \in \mathcal{P}$ are incomparable w.r.t. subsumption.
Characterizing Safety

Assume \( \mathcal{P} \) is redundant-free: every \( D_i, D_j \in \mathcal{P} \) are incomparable w.r.t. subsumption.

Safety

\( C' \) is safe for \( \mathcal{P} \) iff there is no pair of atoms \( (E, F) \) such that

\[
E \in \text{con}(C'), \ F \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p) \text{ and } E \subseteq F
\]

Deciding whether \( C' \) is safe for \( \mathcal{P} \) is in PTime.
Assume $\mathcal{P}$ is redundant-free: every $D_i, D_j \in \mathcal{P}$ are incomparable w.r.t. subsumption.

**Safety**

$C'$ is safe for $\mathcal{P}$ iff there is no pair of atoms $(E, F)$ such that

$$E \in \text{con}(C'), \; F \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p) \text{ and } E \sqsubseteq F$$

Deciding whether $C'$ is safe for $\mathcal{P}$ is in PTime.

**The Optimal $\mathcal{P}$-Safe Generalization**

- If $C'_1, C'_2$ are $\mathcal{P}$-safe generalizations of $C$, then $C'_1 \sqcap C'_2$ is also a $\mathcal{P}$-safe generalization of $C$.
  
  $\Rightarrow$ Optimal $\mathcal{P}$-safe generalization is unique up to equivalence.
Characterizing Safety

Assume $\mathcal{P}$ is **redundant-free**: every $D_i, D_j \in \mathcal{P}$ are **incomparable w.r.t. subsumption**.

### Safety

$\mathcal{C}'$ is safe for $\mathcal{P}$ iff there is **no pair of atoms** $(E, F)$ such that

$$E \in \text{con}(\mathcal{C}'), \ F \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p) \text{ and } E \sqsubseteq F$$

Deciding whether $\mathcal{C}'$ is safe for $\mathcal{P}$ is in **PTime**.

### The Optimal $\mathcal{P}$-Safe Generalization

- If $\mathcal{C}_1', \mathcal{C}_2'$ are $\mathcal{P}$-safe generalizations of $\mathcal{C}$, then $\mathcal{C}_1' \cap \mathcal{C}_2'$ is also a $\mathcal{P}$-safe generalization of $\mathcal{C}$.
  - $\Rightarrow$ Optimal $\mathcal{P}$-safe generalization is **unique up to equivalence**.
- The $\mathcal{P}$-optimal safe generalization of $\mathcal{C}$ can be **computed in ExpTime**.
  - $\Rightarrow$ Requiring the computation of $\mathcal{P}$-optimal compliant generalizations.
Deciding Optimality

- **Deciding** whether $C'$ a $\mathcal{P}$-optimal compliant (safe) generalization of $C$.

- It can be done in ExpTime
  - Compute the set of all $\mathcal{P}$-optimal compliant (safe) generalization of $C$.
  - Check whether $C'$ belongs to the set.
Deciding Optimality

- **Deciding** whether $C'$ is a $\mathcal{P}$-optimal compliant (safe) generalization of $C$.

- It can be done in ExpTime
  - Compute the set of all $\mathcal{P}$-optimal compliant (safe) generalization of $C$.
  - Check whether $C'$ belongs to the set.

- It can be improved to **coNP**.

- **Idea**: Design an NP algorithm for deciding non-optimality
  1. Guess a **lower neighbor** $C''$ of $C'$ subsuming $C$.
     $C \sqsubseteq C'' \sqsubseteq C'$ and there is no $C'''$ such that $C'' \sqsubset C''' \sqsubset C'$.
  2. Check whether $C''$ is a compliant (safe)-generalization of $C$. 
Deciding Optimality

- **Deciding** whether $C'$ a $P$-optimal compliant (safe) generalization of $C$.

- It can be done in ExpTime
  - Compute the set of all $P$-optimal compliant (safe) generalization of $C$.
  - Check whether $C'$ belongs to the set.

- It can be improved to \textbf{coNP}.

- **Idea**: Design an NP algorithm for deciding non-optimality
  1. Guess a \textit{lower neighbor} $C''$ of $C'$ subsuming $C$.
     
     $C \sqsubseteq C'' \sqsubseteq C'$ and there is no $C'''$ such that $C'' \sqsubseteq C''' \sqsubseteq C'$.

  2. Check whether $C''$ is a compliant (safe)-generalization of $C$.

- The converse of lower neighbor: \textbf{Upper Neighbor} $\sqsubseteq_1$ (Baader, et. al., 2018).

- Only \textbf{polynomially many} upper neighbors of $\mathcal{EL}$-concepts and each of them is of \textbf{polynomial size} (Kriegel, 2018).
Deciding Optimality

- **Deciding** whether $C'$ a $P$-optimal compliant (safe) generalization of $C$.

  It can be done in ExpTime
  - Compute the set of all $P$-optimal compliant (safe) generalization of $C$.
  - Check whether $C'$ belongs to the set.

- It can be improved to coNP.

- **Idea**: Design an NP algorithm for deciding non-optimality
  1. Guess a lower neighbor $C''$ of $C'$ subsuming $C$.
     
     $C \sqsubseteq C'' \sqsubseteq C'$ and there is no $C'''$ such that $C'' \sqsubset C''' \sqsubset C'$.
  2. Check whether $C''$ is a compliant (safe)-generalization of $C$.

- The converse of lower neighbor: **Upper Neighbor** $\sqsubseteq_1$ (Baader, et. al., 2018).

- Only polynomially many upper neighbors of $EL$-concepts and each of them is of polynomial size (Kriegel, 2018).

- The next task: **computing lower neighbors**!
Lower neighbors $C''$ of $C'$ can be obtained by conjoining an atom not implied by $C'$ to $C'$.
Lower neighbors $C''$ of $C'$ can be obtained by conjoining an atom not implied by $C'$ to $C'$.

Let $\Sigma$ be a finite set of concept and role names. We define the set $LA_{\Sigma}(C')$ of lowering atoms for $C'$ w.r.t. $\Sigma$. 

Lemma $C''$ is a lower neighbor of $C'$ w.r.t. $\Sigma$ iff there is an atom $At \in LA_{\Sigma}(C')$ such that $C'' \equiv C' \sqcap At$. 
Lower neighbors $C''$ of $C'$ can be obtained by conjoining an atom not implied by $C'$ to $C'$.

Let $\Sigma$ be a finite set of concept and role names. We define the set $LA_\Sigma(C')$ of lowering atoms for $C'$ w.r.t. $\Sigma$.

$L A_\Sigma(C') := \{A \in \Sigma \cap N_C \mid A \notin con(C')\} \cup$
Characterizing Lower Neighbors

- Lower neighbors $C''$ of $C'$ can be obtained by **conjoining an atom** not implied by $C'$ to $C'$.

- Let $\Sigma$ be a **finite set** of concept and role names. We define the set $LA_\Sigma(C')$ of **lowering atoms** for $C'$ w.r.t. $\Sigma$.

$$LA_\Sigma(C') := \{A \in \Sigma \cap N_C | A \not\in \text{con}(C')\} \cup \{\exists r.D | r \in N_R \cap \Sigma, \text{sig}(D) \subseteq \Sigma, C' \not\subseteq \exists r.D \text{ and}$$
Lower neighbors $C''$ of $C'$ can be obtained by \textit{conjoining an atom} not implied by $C'$ to $C'$.

Let $\Sigma$ be a \textbf{finite set} of concept and role names. We define the set $\text{LA}_\Sigma(C')$ of \textbf{lowering atoms} for $C'$ w.r.t. $\Sigma$.

$$\text{LA}_\Sigma(C') := \{ A \in \Sigma \cap N_C \mid A \not\in \text{con}(C') \} \cup \{ \exists r.D \mid r \in N_R \cap \Sigma, \ \text{sig}(D) \subseteq \Sigma, \ C' \not\subseteq \exists r.D \text{ and } C' \subseteq \exists r.E \text{ for all } E \text{ with } D \sqsubseteq_1 E \}$$
Characterizing Lower Neighbors

- Lower neighbors $C''$ of $C'$ can be obtained by conjoining an atom not implied by $C'$ to $C'$.

- Let $\Sigma$ be a finite set of concept and role names. We define the set $LA_\Sigma(C')$ of lowering atoms for $C'$ w.r.t. $\Sigma$.

$$LA_\Sigma(C') := \{ A \in \Sigma \cap N_C \mid A \notin \text{con}(C') \} \cup \{ \exists r. D \mid r \in N_R \cap \Sigma, \text{sig}(D) \subseteq \Sigma, C' \not\sqsubseteq \exists r. D \text{ and } C' \sqsubseteq \exists r. E \text{ for all } E \text{ with } D \sqsubseteq E \}$$

**Lemma**

$C''$ is a lower neighbor of $C'$ w.r.t. $\Sigma$ iff there is an atom $At \in LA_\Sigma(C')$ such that $C'' \equiv C' \cap At$. 

Adrian Nuradiansyah  
Thursday Seminar  
August 20, 2019 10 / 20
Example of Lower Neighbors

Example

\[ \Sigma := \{ r, A_1, A_2, B_1, B_2, C_1, C_2 \} \] and

\[ C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \cap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \cap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2). \]
Example of Lower Neighbors

Example

\( \Sigma := \{ r, A_1, A_2, B_1, B_2, C_1, C_2 \} \) and

\( C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \cap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \cap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2) \).

- if \( D := A_i \cap B_j \cap C_k \) for \( i, j, k \in \{1, 2\} \), then \( \exists r. D \in LA_\Sigma(C'') \).
Example of Lower Neighbors

Example

\[ \Sigma := \{ r, A_1, A_2, B_1, B_2, C_1, C_2 \} \text{ and } \]

\[ C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \cap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \cap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2). \]

- if \( D := A_i \cap B_j \cap C_k \) for \( i, j, k \in \{1, 2\} \), then \( \exists r. D \in LA_\Sigma (C'') \).
- For all upper neighbors \( E \) of \( D \), where \( E \) is only either \( A_i \cap B_j \), \( B_j \cap C_k \), or \( A_i \cap C_k \), we have \( C \sqsubseteq \exists r. E. \)
Example of Lower Neighbors

Example

\[ \Sigma := \{ r, A_1, A_2, B_1, B_2, C_1, C_2 \} \] and

\[ C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \cap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \cap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2). \]

- If \( D := A_i \cap B_j \cap C_k \) for \( i, j, k \in \{1, 2\} \), then \( \exists r. D \in \text{LA}_\Sigma(C'') \).
- For all upper neighbors \( E \) of \( D \), where \( E \) is only either \( A_i \cap B_j \), \( B_j \cap C_k \), or \( A_i \cap C_k \), we have \( C \subseteq \exists r. E \).
- \( C' \cap \exists r. D \) is a lower neighbor of \( C' \).
**Example of Lower Neighbors**

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Sigma := {r, A_1, A_2, B_1, B_2, C_1, C_2}$ and $C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \cap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \cap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2)$.</td>
</tr>
</tbody>
</table>

- If $D := A_i \cap B_j \cap C_k$ for $i, j, k \in \{1, 2\}$, then $\exists r. D \in LA_\Sigma(C'')$.
- For all upper neighbors $E$ of $D$, where $E$ is only either $A_i \cap B_j$, $B_j \cap C_k$, or $A_i \cap C_k$, we have $C \subseteq \exists r. E$.
- $C' \cap \exists r. D$ is a lower neighbor of $C'$

Given $C$ and $\Sigma$, in general, $|LA_\Sigma(C)|$ can be **exponential** in the size of $C$ and $\Sigma$. 
Example

$\Sigma := \{r, A_1, A_2, B_1, B_2, C_1, C_2\}$ and $C' := \exists r. (A_1 \cap A_2 \cap B_1 \cap B_2) \sqcap \exists r. (A_1 \cap A_2 \cap C_1 \cap C_2) \sqcap \exists r. (B_1 \cap B_2 \cap C_1 \cap C_2)$.

- if $D := A_i \cap B_j \cap C_k$ for $i, j, k \in \{1, 2\}$, then $\exists r. D \in LA_\Sigma(C'')$.
- For all upper neighbors $E$ of $D$, where $E$ is only either $A_i \cap B_j$, $B_j \cap C_k$, or $A_i \cap C_k$, we have $C \sqsubseteq \exists r. E$.
- $C' \sqcap \exists r. D$ is a lower neighbor of $C'$

Given $C$ and $\Sigma$, in general, $|LA_\Sigma(C)|$ can be **exponential** in the size of $C$ and $\Sigma$.

To produce exactly the lower neighbors of $C'$ that subsume $C$, let us

- **generate** all $At \in LA_\Sigma(C')$ w.r.t. $\Sigma := \text{sig}(C)$, and
- **remove** the ones that do not subsume $C$. 
But $LA_{\Sigma}(C')$ does not show directly how appropriate $\exists r. D$ can be found!
Generating Lower Neighbors

But $LA_{\Sigma}(C')$ does not show directly how appropriate $\exists r.D$ can be found!

The NP-algorithm generating exactly the elements of $LA_{\Sigma}(C')$ works as follows

1. **Choose** $A \in \Sigma \setminus \text{con}(C')$ and **output** $A$. If there is no such $A$, fail.
But $L_{\Sigma}(C')$ does not show directly how appropriate $\exists r. D$ can be found!

The NP-algorithm generating exactly the elements of $L_{\Sigma}(C')$ works as follows

1. Choose $A \in \Sigma \setminus \text{con}(C')$ and output $A$. If there is no such $A$, fail.

2. Choose $r \in N_R \cap \Sigma$, a set $\{\exists r.F'_1, \ldots, \exists r.F'_k\} \subseteq \text{con}(C')$, and recursively guess $F_1 \in L_{\Sigma}(F'_1), \ldots, F_k \in L_{\Sigma}(F'_k)$. 
Generating Lower Neighbors

But $LA_\Sigma(C')$ does not show directly how appropriate $\exists r.D$ can be found!

The NP-algorithm generating exactly the elements of $LA_\Sigma(C')$ works as follows

1. **Choose** $A \in \Sigma \setminus \text{con}(C')$ and **output** $A$. If there is no such $A$, fail.

2. **Choose** $r \in N_R \cap \Sigma$, a set $\{\exists r.F'_1, \ldots, \exists r.F'_k\} \subseteq \text{con}(C')$, and recursively **guess** $F_1 \in LA_\Sigma(F'_1), \ldots, F_k \in LA_\Sigma(F'_k)$.
   - If for some $i, 1 \leq i \leq k$, it fails to produce $F_i \in LA_\Sigma(F'_i)$, or
   - If $C' \subseteq \exists r.(F_1 \cap \ldots \cap F_k)$, or
   - If $F_1 \cap \ldots \cap F_k$ has an upper neighbor $E$ such that $C' \not\supseteq \exists r.E$, then fail.

Generating Lower Neighbors

But $LA_{\Sigma}(C')$ \textbf{does not show directly} how appropriate $\exists r . D$ can be found!

The NP-algorithm \textit{generating exactly the elements} of $LA_{\Sigma}(C')$ works as follows

1. \textbf{Choose} $A \in \Sigma \setminus \text{con}(C')$ and \textbf{output} $A$. If there is no such $A$, fail.

2. \textbf{Choose} $r \in N_R \cap \Sigma$, a set $\{\exists r . F'_1, \ldots, \exists r . F'_k\} \subseteq \text{con}(C')$, and recursively guess $F_1 \in LA_{\Sigma}(F'_1), \ldots, F_k \in LA_{\Sigma}(F'_k)$.
   - If for some $i, 1 \leq i \leq k$, it fails to produce $F_i \in LA_{\Sigma}(F'_i)$, or
   - If $C' \subseteq \exists r . (F_1 \sqcap \ldots \sqcap F_k)$, or
   - If $F_1 \sqcap \ldots \sqcap F_k$ has an upper neighbor $E$ such that $C' \not\subseteq \exists r . E$, then fail. Otherwise, \textbf{output} $\exists r . (F_1 \sqcap \ldots \sqcap F_k) \equiv \exists r . D$. 
Theorem

The optimality problem is in \( \text{coNP} \) for compliance and for safety in \( \mathcal{EL} \).
The optimality problem is in coNP for compliance and for safety in EL.

We do not know if these problems are also coNP-hard.

The Hypergraph Duality Problem (Dual) can be reduced to them.

Given two families of inclusion-comparable sets $G$ and $H$, Dual asks whether $H$ consists exactly of the minimal hitting sets of $G$. 
The optimality problem is in \textit{coNP} for compliance and for safety in \textit{EL}.

- We \textbf{do not know} if these problems are also \textit{coNP}-hard.
- The Hypergraph Duality Problem (Dual) \textbf{can be reduced} to them.
- Given two \textit{families of inclusion-comparable sets} $\mathcal{G}$ and $\mathcal{H}$, Dual asks whether $\mathcal{H}$ consists exactly of the minimal hitting sets of $\mathcal{G}$.

\textbf{Proposition}

There is a \textit{polynomial reduction} of Dual to the optimality problem for compliance and safety.
What we considered before:

- Knowledge about individuals
- Privacy policies
- Background knowledge of attackers

are represented by $\mathcal{EL}$ concepts.
Considering Different Attacker’s Knowledge

- What we considered before:
  - Knowledge about individuals
  - Privacy policies
  - Background knowledge of attackers

are represented by $\mathcal{EL}$ concepts.

- Background Knowledge of Attackers: $\mathcal{FL}_0$ or $\mathcal{FLE}$ concepts?

---

Subsumption without general TBoxes:
in $\mathcal{FL}_0$: PTime
in $\mathcal{FLE}$: NP-complete

In SNOMED CT, the roles have implicit typing constraints, that may be known to an attacker.
Considering Different Attacker’s Knowledge

- What we considered before:
  - Knowledge about individuals
  - Privacy policies
  - Background knowledge of attackers

  are represented by $\mathcal{EL}$ concepts.

- Background Knowledge of Attackers: $\mathcal{FL}_0$ or $\mathcal{FLE}$ concepts?

- $\mathcal{FL}_0$ concepts:
  $$C, D ::= \top | A | C \sqcap D | \forall r. C$$

- $\mathcal{FLE}$ concepts:
  $$C, D ::= \top | A | C \sqcap D | \exists r. C | \forall r. D$$
Considering Different Attacker’s Knowledge

- What we considered before:
  - Knowledge about individuals
  - Privacy policies
  - Background knowledge of attackers

Background knowledge of attackers are represented by $\mathcal{EL}$ concepts.

- Background Knowledge of Attackers: $\mathcal{FL}_0$ or $\mathcal{FLE}$ concepts?

  $\mathcal{FL}_0$ concepts:
  
  $$C, D ::= \top | A | C \sqcap D | \forall r.C$$

  $\mathcal{FLE}$ concepts:

  $$C, D ::= \top | A | C \sqcap D | \exists r.C | \forall r.D$$

- Subsumption without general TBoxes:

  - in $\mathcal{FL}_0$: PTime
  - in $\mathcal{FLE}$: NP-complete

- In SNOMED CT, the roles have implicit typing constraints, that may be known to an attacker.
Let $C$ be an $\mathcal{EL}$ concept, $\mathcal{P}$ be an $\mathcal{EL}$ policy, $Q \in \{\forall, \forall\exists\}$, and $\mathcal{L}_\forall = \mathcal{FL}_0, \mathcal{L}_\forall\exists = \mathcal{FL}_E$.

The $\mathcal{L}_Q$ concept $C'$ is **compliant** with $\mathcal{P}$ if $C' \not\sqsubseteq D$ for all $D \in \mathcal{P}$.

The $\mathcal{EL}$ concept $C'$ is

- **$Q$-safe** for $\mathcal{P}$ if $C' \sqcap C''$ is compliant with $\mathcal{P}$ for all $\mathcal{L}_Q$ concepts $C''$ that are compliant with $\mathcal{P}$.
- a **$Q$-safe generalization** of $C$ for $\mathcal{P}$ if $C \sqsubseteq C'$ and $C'$ is $Q$-safe for $\mathcal{P}$,
- an **optimal $Q$-safe generalization** of $C$ for $\mathcal{P}$ if
  - it is a $Q$-safe generalization of $C$ for $\mathcal{P}$ and
  - there is no $Q$-safe generalization of $C$ for $\mathcal{P}$ such that $C'' \sqsubseteq C'$.

We now focus on $\forall$-safety and $\forall\exists$-safety.
Illustrations on $\forall$-Safety and $\forall\exists$-Safety

Let us consider again

$$D = Patient \land \exists\text{seen\_by.}(Doctor \land \exists\text{works\_in.\text{Cardiology}})$$

...and the published information $C$ about linda

$$C = Patient \land Female \land \exists\text{seen\_by.}(Doctor \land Male \land \exists\text{works\_in.\text{Cardiology}})$$

Note $C$ is not compliant with $D$, i.e., $C \subset D$.

Compute the optimal safe generalization

$$C_3 = Female \land \exists\text{seen\_by.}(Doctor \land Male \land \exists\text{works\_in.\top})$$
$$\land \exists\text{seen\_by.}(Male \land \exists\text{works\_in.\text{Cardiology}})$$

But then, if the attacker's knowledge is given by an $\mathcal{FL}_0$ concept

$$F_1 = \forall\text{seen\_by.}\forall\text{works\_in.\text{Cardiology}}$$

then $C_3 \land F_1 \subset D$. 
Illustrations on $\forall$-Safety and $\forall\exists$-Safety

Let us consider again

$$D = Patient \cap \exists\text{seen}_\text{by}. (Doctor \cap \exists\text{works}_\text{in}.\text{Cardiology})$$

...and the published information $C$ about linda

$$C = Patient \cap Female \cap \exists\text{seen}_\text{by}. (Doctor \cap Male \cap \exists\text{works}_\text{in}.\text{Cardiology})$$

Note $C$ is not compliant with $D$, i.e., $C \subseteq D$.

Compute an optimal $\forall$-safe generalization

$$C_4 = Male \cap Patient \cap \exists\text{seen}_\text{by}. (Doctor \cap Female)$$

However, if the attacker’s knowledge is given by an $\mathcal{FLE}$ concept $F_2 = \forall\text{seen}_\text{by}. \exists\text{works}_\text{in}.\text{Cardiology}$, then $C_4 \cap F_2 \subseteq D$. 
Illustrations on $\forall$-Safety and $\forall\exists$-Safety

Let us consider again

$$D = Patient \sqcap \exists \text{seen\_by.}\,(Doctor \sqcap \exists \text{works\_in.Cardiology})$$

... and the published information $C$ about linda

$$C = Patient \sqcap Female \sqcap \exists \text{seen\_by.}\,(Doctor \sqcap Male \sqcap \exists \text{works\_in.Cardiology})$$

Note $C$ is not compliant with $D$, i.e., $C \subseteq D$.

Compute an optimal $\forall$-safe generalization

$$C_4 = Male \sqcap Patient \sqcap \exists \text{seen\_by.}\,(Doctor \sqcap Female)$$

However, if the attacker’s knowledge is given by an $\mathcal{FLE}$ concept $F_2 = \forall \text{seen\_by.}\exists \text{works\_in.Cardiology}$, then $C_4 \sqcap F_2 \subseteq D$.

Compute the optimal $\forall\exists$-safe generalization $C_5 = Male$
Characterizing $\forall$-Safety

$\forall$-Safety

$C'$ is $\forall$-safe for $\mathcal{P}$ iff for all $D \in \mathcal{P}$:

1. if $rd(D) = 0$, then $\text{con}(C) \cap \text{con}(D) = \emptyset$.

Complexity for $\forall$-Safety

Deciding whether $C'$ is $\forall$-safe for $\mathcal{P}$ is in $\text{PTime}$.

One optimal $\forall$-safe generalization for $\mathcal{P}$ can be computed in $\text{ExpTime}$.

The set of all optimal $\forall$-safe generalizations for $\mathcal{P}$ can be computed in $\text{ExpTime}$.

$\forall$-optimality is in $\text{coNP}$. 

Adrian Nuradiansyah
Thursday Seminar
August 20, 2019 17 / 20
Characterizing $\forall$-Safety

$\forall$-Safety

$C'$ is $\forall$-safe for $\mathcal{P}$ iff for all $D \in \mathcal{P}$:

1. if $rd(D) = 0$, then $\text{con}(C) \cap \text{con}(D) = \emptyset$.

2. if $rd(D) > 0$, then there is $\exists r.D' \in \text{con}(D)$ such that
   a. if $rd(D') = 0$, then there is no concept of the form $\exists r.C' \in \text{con}(C)$,
   b. if $rd(D') > 0$, then for all $\exists r.C' \in \text{con}(C)$, $C'$ is $\forall$-safe for $\{D'\}$. 

Complexity for $\forall$-Safety
Deciding whether $C'$ is $\forall$-safe for $\mathcal{P}$ is in PTime.

One optimal $\forall$-safe generalization for $\mathcal{P}$ can be computed in ExpTime.

The set of all optimal $\forall$-safe generalizations for $\mathcal{P}$ can be computed in ExpTime.

$\forall$-optimality is in coNP.
Characterizing $\forall$-Safety

$\forall$-Safety

$C'$ is $\forall$-safe for $\mathcal{P}$ iff for all $D \in \mathcal{P}$:

1. if $rd(D) = 0$, then $\text{con}(C) \cap \text{con}(D) = \emptyset$.
2. if $rd(D) > 0$, then there is $\exists r.D' \in \text{con}(D)$ such that
   a. if $rd(D') = 0$, then there is no concept of the form $\exists r.C' \in \text{con}(C)$,
   b. if $rd(D') > 0$, then for all $\exists r.C' \in \text{con}(C)$, $C'$ is $\forall$-safe for $\{D\}$.

Complexity for $\forall$-Safety

- Deciding whether $C'$ is $\forall$-safe for $\mathcal{P}$ is in $PTime$.
- One optimal $\forall$-safe generalization for $\mathcal{P}$ can be computed in $ExpTime$.
- The set of all optimal $\forall$-safe generalizations for $\mathcal{P}$ can be computed in $ExpTime$.
- $\forall$-optimality is in $coNP$. 

Adrian Nuradiansyah  
Thursday Seminar  
August 20, 2019 17 / 20
Characterizing \( \forall \exists \)-Safety

\( \forall \exists \)-Safety

\( C \) is \( \forall \exists \)-safe for \( \mathcal{P} \) iff

1. \( A \notin \text{con}(C) \) for all concept names \( A \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p) \), and
2. for all existential restrictions \( \exists r.D' \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p) \), there is no concept of the form \( \exists r.E \in \text{con}(C) \)
Characterizing $\forall\exists$-Safety

$\forall\exists$-Safety

$C$ is $\forall\exists$-safe for $\mathcal{P}$ iff

1. $A \notin \text{con}(C)$ for all concept names $A \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p)$, and
2. for all existential restrictions $\exists r.D' \in \text{con}(D_1) \cup \ldots \cup \text{con}(D_p)$, there is no concept of the form $\exists r.E \in \text{con}(C)$

Complexity for $\forall$-Safety

Given $\mathcal{EL}$ concepts $C, C''$ and a redundancy-free $\mathcal{EL}$ policy $\mathcal{P}$, we

- can decide if $C$ is $\forall\exists$-safe for $\mathcal{P}$,
- can compute the unique optimal $\forall\exists$-safe generalization of $C$ for $\mathcal{P}$, and
- can decide if $C''$ is an optimal $\forall\exists$-safe generalization of $C$ for $\mathcal{P}$

in polynomial time
Conclusions and Future Work

Conclusions:

- Define and provide characterizations for compliance, safety, and optimality in privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores.
- Computing $\mathcal{P}$-optimal compliant (safe) generalizations of $\mathcal{EL}$ concepts.
- Deciding the optimality problem via computing lower neighbors of $\mathcal{EL}$ concepts.
- Considering attacker’s knowledge to be given by an $\mathcal{FL}_0$ or $\mathcal{FLE}$ concept.

Future Work:

- $\mathcal{P}$POP in $\mathcal{EL}$ Instance Stores w.r.t. General TBoxes
- $\mathcal{P}$POP in $\mathcal{EL}$ ABoxes
- Representing attacker’s knowledge with more different DLs
Conclusions and Future Work

Conclusions:

- Define and provide characterizations for compliance, safety, and optimality in privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores.
- Computing $\mathcal{P}$-optimal compliant (safe) generalizations of $\mathcal{EL}$ concepts.
- Deciding the optimality problem via computing lower neighbors of $\mathcal{EL}$ concepts.
- Considering attacker’s knowledge to be given by an $\mathcal{FL}_0$ or $\mathcal{FLE}$ concept.
  \[ \Rightarrow \text{the stronger knowledge of the attacker, the more radical we need to change the concept to make it safe} \]

Future Work:

- PPOP in $\mathcal{EL}$ Instance Stores w.r.t. General TBoxes
- PPOP in $\mathcal{EL}$ ABoxes
- Representing attacker’s knowledge with more different DLs
Conclusions and Future Work

Conclusions:

- Define and provide characterizations for **compliance, safety, and optimality** in privacy-preserving ontology publishing for $\mathcal{EL}$ instance stores.
- Computing $\mathcal{P}$-optimal compliant (safe) generalizations of $\mathcal{EL}$ concepts.
- Deciding the optimality problem via computing lower neighbors of $\mathcal{EL}$ concepts.
- Considering attacker’s knowledge to be given by an $\mathcal{FL}_0$ or $\mathcal{FLE}$ concept. ⇒ the stronger knowledge of the attacker, the more radical we need to change the concept to make it safe.

Future Work:

- PPOP in $\mathcal{EL}$ Instance Stores w.r.t. General TBoxes
- PPOP in $\mathcal{EL}$ ABoxes
- Representing attacker’s knowledge with more different DLs