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

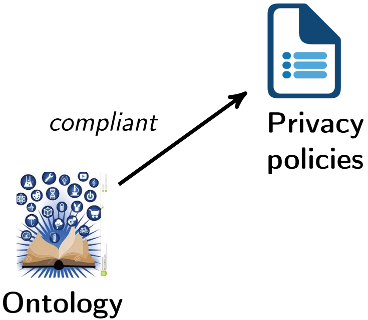
Franz Baader  Francesco Kriegel  **Adrian Nuradiansyah**

**Technische Universität Dresden**

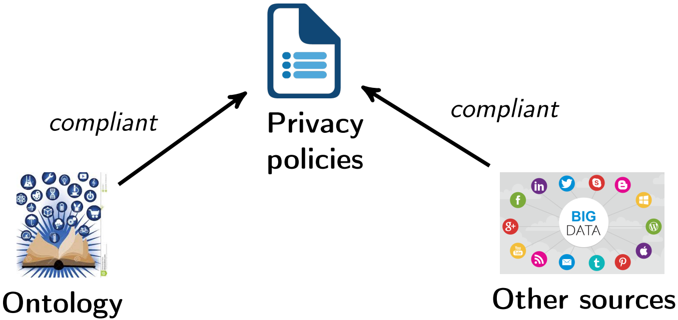
Published in **JELIA 2019** and Submitted to
***Künstliche Intelligenz* (KI)** 2019

August 20, 2019

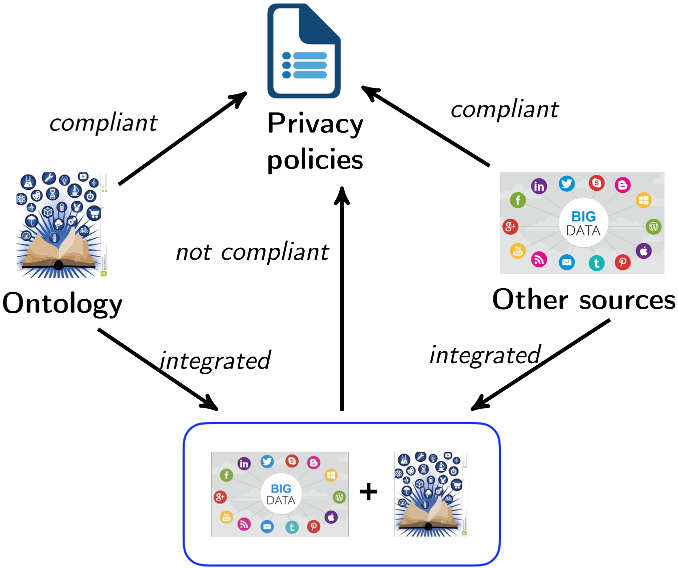
# Privacy-Preserving Ontology Publishing



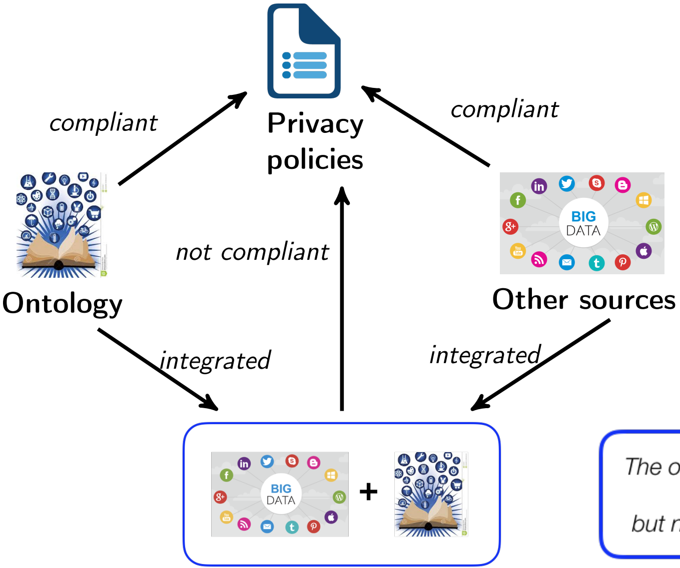
# Privacy-Preserving Ontology Publishing



# Privacy-Preserving Ontology Publishing



# Privacy-Preserving Ontology Publishing




*The ontology is compliant,*

*but not safe w.r.t policies*

# Privacy-Preserving Ontology Publishing

## 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.

# Privacy-Preserving Ontology Publishing

## 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**

# 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 $D(a)$.

# 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.

# 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

# 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 $\xrightarrow{\textit{reduced}}$ 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 $i \in \{1, \ldots, p\}$.

# Compliance, Safety, and Optimality

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 $\mathcal{P}$.

# Compliance, Safety, and Optimality

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) $\mathcal{P}$.

# Compliance, Safety, and Optimality

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. $C'' \sqsubset C'$.

# Illustration on Compliance, Safety, and Optimality

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

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

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

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

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

# Illustration on Compliance, Safety, and Optimality

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

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

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

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

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

- Generalizing $C$ to yield a compliant concept

$$C_1 = Female \sqcap \exists seen\_by.(Doctor \sqcap Male \sqcap \exists works\_in.Cardiology)$$

  But, $C_1$ is **not safe for** $D$ since if the attacker knows $Patient(linda)$, then $C_1 \sqcap Patient \sqsubseteq D$ is revealed.

# Illustration on Compliance, Safety, and Optimality

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

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

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

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

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

- Let us **make it safe**!

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

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*

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

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

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

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

- Let us **make it safe**!

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

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

- Make it **optimal**!

$$\begin{aligned} C_3 = Female \quad & \sqcap \exists seen\_by.(Doctor \sqcap Male \sqcap \exists works\_in.\top) \\ & \sqcap \exists seen\_by.(Male \sqcap \exists works\_in.Cardiology) \end{aligned}$$

# Characterizing Compliance

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

# Characterizing Compliance

- Let $\mathtt{con}(C)$ be the set of all **atoms** $A$ or $\exists r.E$ occurring in the **top-level conjunction** of $C$.
- $\mathtt{con}(C)$ **covers** $\mathtt{con}(D)$ iff for all $F \in \mathtt{con}(D)$, there is $E \in \mathtt{con}(C)$ such that $E \sqsubseteq F$

# Characterizing Compliance

- Let $\texttt{con}(C)$ be the set of all **atoms** $A$ or $\exists r.E$ occurring in the **top-level conjunction** of $C$.
- $\texttt{con}(C)$ **covers** $\texttt{con}(D)$ iff for all $F \in \texttt{con}(D)$, there is $E \in \texttt{con}(C)$ such that $E \sqsubseteq F \Rightarrow$ Characterizing $C \sqsubseteq D$.

# Characterizing Compliance

- Let $\mathtt{con}(C)$ be the set of all **atoms** $A$ or $\exists r.E$ occurring in the **top-level conjunction** of $C$.
- $\mathtt{con}(C)$ **covers** $\mathtt{con}(D)$ iff for all $F \in \mathtt{con}(D)$, there is $E \in \mathtt{con}(C)$ such that $E \sqsubseteq F \Rightarrow$ Characterizing $C \sqsubseteq D$.

## Compliance

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

# Characterizing Compliance

- Let $\mathtt{con}(C)$ be the set of all **atoms** $A$ or $\exists r.E$ occurring in the **top-level conjunction** of $C$.
- $\mathtt{con}(C)$ **covers** $\mathtt{con}(D)$ iff for all $F \in \mathtt{con}(D)$, there is $E \in \mathtt{con}(C)$ such that $E \sqsubseteq F \Rightarrow$ Characterizing $C \sqsubseteq D$.

## Compliance

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

## Complexity for Compliance

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

# Characterizing Compliance

- Let $\mathrm{con}(C)$ be the set of all **atoms** $A$ or $\exists r.E$ occurring in the **top-level conjunction** of $C$.
- $\mathrm{con}(C)$ **covers** $\mathrm{con}(D)$ iff for all $F \in \mathrm{con}(D)$, there is $E \in \mathrm{con}(C)$ such that $E \sqsubseteq F$ $\Rightarrow$ Characterizing $C \sqsubseteq D$.

## Compliance

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

## Complexity for Compliance

- Deciding whether $C'$ is compliant w.r.t. $\mathcal{P}$ is in **PTime.**
- One optimal $\mathcal{P}$-compliant generalization can be **computed in ExpTime**.
- The set of all optimal $\mathcal{P}$-compliant generalizations can be **computed in ExpTime**.

# Characterizing Safety

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

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

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

# 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

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

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

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

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

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**.
- The $\mathcal{P}$-optimal safe generalization of $C$ 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 $C'$ 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 $\mathcal{P}$-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 $C$.
- The converse of lower neighbor: **Upper Neighbor** $\sqsubseteq_1$ (Baader, et. al., 2018).
- Only **polynomially many** upper neighbors of $\mathcal{EL}$-concepts and each of them is of **polynomial size** (Kriegel, 2018).

# 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.
- 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 $\mathcal{EL}$-concepts and each of them is of **polynomial size** (Kriegel, 2018).
- The next task: **computing lower neighbors!**

# Characterizing Lower Neighbors

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

# 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. $\Sigma$.

# 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. $\Sigma$.
- $LA_\Sigma(C') := \{A \in \Sigma \cap N_C \mid A \notin \mathrm{con}(C')\} \cup$

# 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. $\Sigma$.
- $LA_\Sigma(C') := \{A \in \Sigma \cap N_C \mid A \notin \mathrm{con}(C')\} \cup$
  $\{\exists r.D \mid r \in N_R \cap \Sigma,\ sig(D) \subseteq \Sigma,\ C' \not\sqsubseteq \exists r.D$ and

# 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. $\Sigma$.
- $$LA_\Sigma(C') := \{A \in \Sigma \cap N_C \mid A \notin \mathtt{con}(C')\} \cup \{\exists r.D \mid r \in N_R \cap \Sigma,\ sig(D) \subseteq \Sigma,\ C' \not\sqsubseteq \exists r.D \text{ and } C' \sqsubseteq \exists r.E \text{ for all } E \text{ with } D \sqsubset_1 E\}$$

# 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. $\Sigma$.
- $$\begin{aligned} LA_\Sigma(C') := \{A \in \Sigma \cap N_C \mid A \notin \mathrm{con}(C')\} \cup \\ \{\exists r.D \mid r \in N_R \cap \Sigma,\ sig(D) \subseteq \Sigma,\ C' \not\sqsubseteq \exists r.D \text{ and} \\ C' \sqsubseteq \exists r.E \text{ for all } E \text{ with } D \sqsubset_1 E\} \end{aligned}$$

**Lemma**

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

# Example of Lower Neighbors

## Example

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

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

# Example of Lower Neighbors

## Example

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

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

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

# Example of Lower Neighbors

## Example

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

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

- if $D := A_i \sqcap B_j \sqcap C_k$ 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 $A_i \sqcap B_j$, $B_j \sqcap C_k$, or $A_i \sqcap C_k$, 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

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

- if $D := A_i \sqcap B_j \sqcap C_k$ 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 $A_i \sqcap B_j$, $B_j \sqcap C_k$, or $A_i \sqcap C_k$, we have $C \sqsubseteq \exists r.E$.
- $C' \sqcap \exists r.D$ is a lower neighbor of $C'$

# Example of Lower Neighbors

## Example

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

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

- if $D := A_i \sqcap B_j \sqcap C_k$ 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 $A_i \sqcap B_j$, $B_j \sqcap C_k$, or $A_i \sqcap C_k$, 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 $\Sigma$.

# Example of Lower Neighbors

## Example

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

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

- if $D := A_i \sqcap B_j \sqcap C_k$ 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 $A_i \sqcap B_j$, $B_j \sqcap C_k$, or $A_i \sqcap C_k$, 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. $\Sigma := sig(C)$, and
- **remove** the ones that do not subsume $C$.

# Generating Lower Neighbors

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

# 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** $A \in \Sigma \setminus \mathrm{con}(C')$ and **output** $A$. If there is no such $A$, fail.
2. **Choose** $r \in N_R \cap \Sigma$, a set $\{\exists r.F_1', \dots, \exists r.F_k'\} \subseteq \mathrm{con}(C')$, and recursively **guess** $F_1 \in LA_\Sigma(F_1'), \dots, F_k \in LA_\Sigma(F_k')$.

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

2. **Choose** $r \in N_R \cap \Sigma$, a set $\{\exists r.F_1', \ldots, \exists r.F_k'\} \subseteq \mathrm{con}(C')$, 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 $C' \sqsubseteq \exists r.(F_1 \sqcap \ldots \sqcap F_k)$, or
   - If $F_1 \sqcap \ldots \sqcap F_k$ has an upper neighbor $E$ such that $C' \not\sqsubseteq \exists r.E$,

   then fail.

# 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** $A \in \Sigma \setminus \mathrm{con}(C')$ and **output** $A$. If there is no such $A$, fail.
2. **Choose** $r \in N_R \cap \Sigma$, a set $\{\exists r.F'_1, \ldots, \exists r.F'_k\} \subseteq \mathrm{con}(C')$, 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 $C' \sqsubseteq \exists r.(F_1 \sqcap \ldots \sqcap F_k)$, or
   - If $F_1 \sqcap \ldots \sqcap F_k$ has an upper neighbor $E$ such that $C' \not\sqsubseteq \exists r.E$,

   then fail. Otherwise, **output** $\exists r.(F_1 \sqcap \ldots \sqcap F_k) \equiv \exists r.D$.

# Complexity for the Optimality Problem

**Theorem**

*The optimality problem is in* **coNP** *for compliance and for safety in* $\mathcal{EL}$.

# Complexity for the Optimality Problem

**Theorem**

*The optimality problem is in **coNP** for compliance and for safety in $\mathcal{EL}$.*

- 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** $\mathcal{G}$ and $\mathcal{H}$, Dual asks whether $\mathcal{H}$ consists exactly of the minimal hitting sets of $\mathcal{G}$.

# Complexity for the Optimality Problem

**Theorem**

*The optimality problem is in **coNP** for compliance and for safety in $\mathcal{EL}$.*

- 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** $\mathcal{G}$ and $\mathcal{H}$, Dual asks whether $\mathcal{H}$ consists exactly of the minimal hitting sets of $\mathcal{G}$.

**Proposition**

*There is a **polynomial reduction** of* Dual *to the optimality problem for compliance and safety*

# Considering Different Attacker's Knowledge

- 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?**

# 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:
$$C, D ::= \top \mid A \mid C \sqcap D \mid \forall r.C$$
- $\mathcal{FLE}$ concepts:
$$C, D ::= \top \mid A \mid C \sqcap D \mid \exists r.C \mid \forall r.D$$

# 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:

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

- $\mathcal{FLE}$ concepts:

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

- 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.

# Extending the Definition of Compliance and Safety

Let $C$ be an $\mathcal{EL}$ concept, $\mathcal{P}$ be an $\mathcal{EL}$ policy, $Q \in \{\forall, \forall\exists\}$, and $\mathcal{L}_\forall = \mathcal{FL}_0, \mathcal{L}_{\forall\exists} = \mathcal{FLE}$.

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 $C'' \sqsubset C'$.

We now focus on $\forall$-safety and $\forall\exists$-safety

# Illustrations on ∀-Safety and ∀∃-Safety

- Let us consider again

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

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

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

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

- Compute the **optimal safe generalization**

$$\begin{aligned} C_3 = Female \quad & \sqcap \exists seen\_by.(Doctor \sqcap Male \sqcap \exists works\_in.\top) \\ & \sqcap \exists seen\_by.(Male \sqcap \exists works\_in.Cardiology) \end{aligned}$$

But then, if the attacker's knowledge is given by an $\mathcal{FL}_0$ concept $F_1 = \forall seen\_by.\forall works\_in.Cardiology$, then $C_3 \sqcap F_1 \sqsubseteq D$.

# Illustrations on $\forall$-Safety and $\forall\exists$-Safety

- Let us consider again

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

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

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

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

- Compute an **optimal $\forall$-safe generalization**

$$C_4 = Male \sqcap Patient \sqcap \exists seen\_by.(Doctor \sqcap Female)$$

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

# Illustrations on ∀-Safety and ∀∃-Safety

- Let us consider again

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

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

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

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

- Compute an **optimal ∀-safe generalization**

$$C_4 = Male \sqcap Patient \sqcap \exists seen\_by.(Doctor \sqcap Female)$$

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

- Compute the **optimal ∀∃-safe generalization** $C_5 = Male$

# Characterizing ∀-Safety

## ∀-Safety

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

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

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

# Characterizing ∀-Safety

## ∀-Safety

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

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

## Complexity for ∀-Safety

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

# Characterizing ∀∃-Safety

## ∀∃-Safety

$C$ is **∀∃-safe for $\mathcal{P}$** iff

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

# Characterizing ∀∃-Safety

## ∀∃-Safety

$C$ is **∀∃-safe for $\mathcal{P}$** iff

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

## Complexity for ∀-Safety

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

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

# 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.
  $\Rightarrow$ *the stronger knowledge of the attacker, the more radical we need to change the concept to make it safe*

# 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.
  $\Rightarrow$ *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

# Thank You

ROSI