Knowledge-based Confidentiality-preserving Query Answering

Riccardo Rosati

Dipartimento di Ingegneria informatica, automatica e gestionale
Sapienza Università di Roma, Italy

Joint work with: Piero Bonatti, Gianluca Cima, Domenico Lembo, Lorenzo Marconi, Luigi Sauro, Domenico Fabio Savo

WEBIST 2022 - October 26, 2022

Outline

Outline

Confidentiality-preserving query answering

- Scenario: system providing access (query answering service) to a dataset
- Problem: enforce a **confidentiality-preserving policy**
  - i.e. some data cannot be disclosed to the system users
- Solution 1: **modify the dataset** in order to enforce the policy
  - and do not change the system/query answering service
- Solution 2: **modify the query answering service** in order to enforce the policy
  - and do not change the dataset

Controlled Query Evaluation (CQE) in Description Logics

We study confidentiality-preserving query answering in Description Logics (DLs) in the spirit of **Controlled Query Evaluation (CQE)**

CQE is a confidentiality-preserving query answering approach studied:

- in databases *[Sicherman et al.,TODS 1983]*
- in Description Logic (DL) ontologies *[Bonatti and Sauro,ISWC 2013], [Cuenca Grau et al.,IJCAI 2015]*

In CQE:

- the **policy** is specified in terms of **logical formulas**
- the enforcement of the policy is formalized through the notion of **censor**
- a censor models the **answers** that the query answering system should provide
- an **optimal censor** maximizes query answers still guaranteeing that the policy is not violated

Controlled Query Evaluation (CQE) in Description Logics

We study confidentiality-preserving query answering in Description Logics (DLs) in the spirit of **Controlled Query Evaluation (CQE)**

CQE is a confidentiality-preserving query answering approach studied:

- in databases *[Sicherman et al.,TODS 1983]*
- in Description Logic (DL) ontologies *[Bonatti and Sauro,ISWC 2013], [Cuenca Grau et al.,IJCAI 2015]*

In CQE:

- the **policy** is specified in terms of **logical formulas**
- the enforcement of the policy is formalized through the notion of **censor**
- a censor models the **answers** that the query answering system should provide
- an **optimal censor** maximizes query answers still guaranteeing that the policy is not violated

Controlled Query Evaluation (CQE) in Description Logics

In **[IJCAI 2019]** we consider a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$, where:

- $\mathcal{T}$ is a DL TBox, representing intensional knowledge (i.e., the schema)
- $\mathcal{A}$ is a DL ABox, that is a set of facts (i.e., the data instance)
- $\mathcal{P}$ is the policy, i.e., a set of *denial assertions* of the form $\forall \vec{x}. cq(\vec{x}) \rightarrow \bot$, s.t. $\exists \vec{x}. cq(\vec{x})$ is a Conjunctive Query (CQ)

A user asks queries over the TBox $\mathcal{T}$, but must not get as result data that let her answer a query $\exists \vec{x}. cq(\vec{x})$ occurring in a denial assertions in $\mathcal{P}$.

A censor is a function that modifies query answers to ensure this behaviour!

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \to \bot;\ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \to \bot \}$

Secrets $= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}, \{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$, *and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot;\ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets $= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}, \{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$, *and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot; \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets $= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}, \{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$, *and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq$ $\exists$admissionWard; Diabetic $\sqsubseteq$ $\exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets $= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}, \{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$, *and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot;\ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets$= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}$, $\{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$, *and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Ground Atoms (GA) Censors

Notation: $\mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ denotes the set of facts (aka GAs) implied by $\mathcal{T} \cup \mathcal{A}$.

A *GA censor* $\mathbf{c}$ for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns a set (called the *theory of the censor*) $\mathrm{Th}_{\mathbf{c}} \subseteq \mathsf{cl}_{\mathsf{GA}}^{\mathcal{T}}(\mathcal{A})$ s.t. $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \not\models \neg\mathcal{P}$.

$\mathbf{c}$ is *optimal* if there is no GA censor $\mathbf{c}'$ for $\mathcal{E}$ such that $\mathrm{Th}_{\mathbf{c}} \subset \mathrm{Th}_{\mathbf{c}'}$

Example:

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathbf{c}1, \mathbf{c}2, \mathbf{c}3$*, and* $\mathbf{c}4$ *are the optimal GA censors for* $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$.

Multiple GA censors

How to deal with multiple GA censors?

1. choose any **arbitrary** GA censor
   - a random choice does not seem to make much sense
2. choose a **predefined** GA censor
   - how? More information would be needed
3. keep **all** the GA censors
   - query answering is done with respect to all the GA censors
   - **skeptical entailment**: only the query answers that are true in all the GA censors are returned
   - studied in **[IJCAI 2019]**

Entailment of Boolean Conjunctive Queries (BCQs)

Given a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ and a BCQ q, we are interested in

Boolean Conjunctive Query (BCQ) entailment under GA censors

i.e., deciding whether $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \models q$ for *every* optimal GA censor $\mathbf{c}$ for $\mathcal{E}$.

Example (cntd):

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.$TakesMedA$(x) \wedge$ Diabetic$(x) \rightarrow \bot$; $\forall x.$TakesMedB$(x) \wedge$ Diabetic$(x) \rightarrow \bot \}$

Secrets$= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}$, $\{$ TakesMedA(ann)), Diabetic$(ann)\}$ $\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

TakesMedA(joe) and $\exists x, y.$admissionWard(x, y) are both entailed by $\mathcal{E}$ under GA censors

Entailment of Boolean Conjunctive Queries (BCQs)

Given a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ and a BCQ q, we are interested in

Boolean Conjunctive Query (BCQ) entailment under GA censors

i.e., deciding whether $\mathcal{T} \cup \mathrm{Th}_{\mathbf{c}} \models q$ for *every* optimal GA censor $\mathbf{c}$ for $\mathcal{E}$.

Example (cntd):

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq$ $\exists$admissionWard; Diabetic $\sqsubseteq$ $\exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{$ $\forall x$.TakesMedA(x) $\wedge$ Diabetic(x) $\rightarrow \bot$; $\forall x$.TakesMedB(x) $\wedge$ Diabetic(x) $\rightarrow \bot$ $\}$

Secrets$= \{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}$, $\{$ TakesMedA(ann)), Diabetic(ann)$\}$ $\}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

TakesMedA(joe) and $\exists x, y$.admissionWard(x, y) are both entailed by $\mathcal{E}$ under GA censors

Entailment of Boolean Conjunctive Queries (BCQs)

Major computational problem:

Skeptical entailment is **coNP-hard** in data complexity (i.e., w.r.t. the size of the ABox only) even for ontology languages/Description Logics of **low expressiveness**

e.g.:

- the Description Logics of the ***DL-Lite*** family (and thus **OWL2 QL**)
- the Description Logics of the $\mathcal{EL}$ family (and thus **OWL2 EL**)

Entailment of Boolean Conjunctive Queries (BCQs)

Major computational problem:

Skeptical entailment is **coNP-hard** in data complexity (i.e., w.r.t. the size of the ABox only) even for ontology languages/Description Logics of **low expressiveness**

e.g.:

- the Description Logics of the ***DL-Lite*** family (and thus **OWL2 QL**)
- the Description Logics of the $\mathcal{EL}$ family (and thus **OWL2 EL**)

Outline

From GA censors to IGA censor

Towards the identification of a practical approach, in **[ISWC 2020]** we introduced the notion of Intersection GA (IGA) censor

An *IGA censor* cens_{IGA} for a CQE instance $\mathcal{E} = \langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a function that returns the *intersection* $\mathrm{Th}_{\text{cens}_{\mathrm{IGA}}}$ of the theories of all optimal GA censors for $\mathcal{E}$.

BCQ entailment under IGA censors: deciding if $\mathcal{T} \cup \mathrm{Th}_{\text{cens}_{\mathrm{IGA}}} \models q$ for a BCQ q

From GA censors to IGA censor

Example (cont'd):

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \rightarrow \bot \}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathrm{Th}_{\mathsf{cens}_{\mathrm{IGA}}} = \{\mathsf{TakesMedA(joe)}\}$.

TakesMedA(joe) is entailed by $\mathcal{E}$ under IGA censors but $\exists x, y.$admissionWard(x, y) is not.

BCQ Entailment is *first-order (FO) rewritable*, and thus in AC^0 in data complexity for ***DL-Lite***$_R$ and **OWL2QL** *[Cima et al.,ISWC 2020]*.

From GA censors to IGA censor

Example (cont'd):

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \to \bot; \ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \to \bot \}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathrm{Th}_{\mathsf{cens}_{\mathrm{IGA}}} = \{$TakesMedA(joe)$\}$.

TakesMedA(joe) is entailed by $\mathcal{E}$ under IGA censors but $\exists x, y.$admissionWard(x, y) is not.

BCQ Entailment is *first-order (FO) rewritable*, and thus in AC^0 in data complexity for ***DL-Lite***$_R$ and **OWL2QL** *[Cima et al.,ISWC 2020]*.

Example (cont'd):

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)),
Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

$\mathrm{Th}_{\mathbf{c}1} = \{$ TakesMedA(bob), TakesMedB(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}2} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}3} = \{$ Diabetic(bob), TakesMedA(ann), TakesMedA(joe) $\}$
$\mathrm{Th}_{\mathbf{c}4} = \{$ Diabetic(bob), Diabetic(ann), TakesMedA(joe) $\}$

$\mathrm{Th}_{\mathsf{cens}_{\mathrm{IGA}}} = \{\mathsf{TakesMedA(joe)}\}$.

TakesMedA(joe) is entailed by $\mathcal{E}$ under IGA censors but $\exists x, y.\mathsf{admissionWard}(x, y)$ is not.

BCQ Entailment is *first-order (FO) rewritable*, and thus in AC^0 in data complexity for ***DL-Lite***$_R$ and **OWL2QL** *[Cima et al.,ISWC 2020]*.

First-order rewritability

What does **first-order rewritable** mean?

for every BCQ Q, we can build a first-order (e.g. SQL) query Q' such that the evaluation of Q' on the **initial knowledge base** is true iff the query Q is true in the IGA censor

Consequences:

- no need to modify the knowledge base (ABox)
- no need to modify the query answering system

IGA censor: properties

- good computational properties
- deterministic, **unique solution**
- but **non-optimal** in terms of disclosed information
- however, **”quasi-optimal”** (**[IJCAI 2020]**)

Outline

Limitations of the framework seen so far

Two main limitations arise:

- **the policy allows only for CQs** in denial assertions, thus ruling out practically relevant formulas, e.g.,

  $\forall x, y.\mathsf{Diabetic}(x) \wedge \mathsf{admissionWard}(x, y) \wedge y \neq \mathsf{'orthopedics'} \rightarrow \bot$

- **censors** studied so far **do not take into account** possibly available **preferences** about the way in which secret information has to be censored

In **[ISWC 2021]**, we have enriched the framework and considered:

1. CQs using **comparison predicates** (i.e., $\neq, \geq, \leq, \ldots$) in policy assertions
2. priorities between ontology predicates, specifying (intentionally) **preferences in disclosing facts** involved in a secret

Limitations of the framework seen so far

Two main limitations arise:

- **the policy allows only for CQs** in denial assertions, thus ruling out practically relevant formulas, e.g.,

$$\forall x, y.\mathsf{Diabetic}(x) \land \mathsf{admissionWard}(x, y) \land y \neq{}'\mathsf{orthopedics}' \rightarrow \bot$$

- **censors** studied so far **do not take into account** possibly available **preferences** about the way in which secret information has to be censored

In **[ISWC 2021]**, we have enriched the framework and considered:

1. CQs using **comparison predicates** (i.e., $\neq$,$\geq$,$\leq$,...) in policy assertions
2. priorities between ontology predicates, specifying (intentionally) **preferences in disclosing facts** involved in a secret

Prioritized CQE framework

- A prioritized CQE instance $\mathcal{E}_{\succ}$ is a tuple $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ such that $\langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a CQE instance and $\succ$ is an *acyclic priority relation over ontology predicates*.
- If $P_1 \succ P_2$, e.g., TakesMedA$\succ$Diabetic, in case a secret involves facts over P_1 and P_2, we prefer to disclose those over P_1 and hide those over P_2.

Desiderata:

- new notion of censor taking in the due account the preference relation and that in case $\succ = \emptyset$ *coincides with the notion of GA censor*.
- definition allowing for practical CQE over prioritized ontologies, i.e., for which BCQ entailment is *FO rewritable* for the DLs of the *DL-Lite* family.
- experimental validation: using preferences may increase the amount of data disclosed to users (still preserving confidential information).

Prioritized CQE framework

- A prioritized CQE instance $\mathcal{E}_{\succ}$ is a tuple $\langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ such that $\langle \mathcal{T}, \mathcal{A}, \mathcal{P} \rangle$ is a CQE instance and $\succ$ is an *acyclic priority relation over ontology predicates*.
- If $P_1 \succ P_2$, e.g., TakesMedA$\succ$Diabetic, in case a secret involves facts over P_1 and P_2, we prefer to disclose those over P_1 and hide those over P_2.

Desiderata:

- new notion of censor taking in the due account the preference relation and that in case $\succ = \emptyset$ *coincides with the notion of GA censor*.
- definition allowing for practical CQE over prioritized ontologies, i.e., for which BCQ entailment is *FO rewritable* for the DLs of the *DL-Lite* family.
- experimental validation: using preferences may increase the amount of data disclosed to users (still preserving confidential information).

Globally-optimal and Pareto-optimal censors

- We initially adapted to our framework the well-known *Pareto* and *Global* optimality notions introduced by *[Staworko et al.,AMAI 2012]* for consistent query answering (CQA) over databases, and then adopted in *[Bienvenu and Bourgaux,KR 2020]* for CQA over prioritized DL ontologies.
- We first defined Pareto (P) and Global (G) censors, and then their approximated version based on intersection, i.e., Intersection P (IP) and Intersection G (IG) censors
- For $\succ = \emptyset$, the sets of Pareto, Global and optimal GA censors coincide, and the IP, IG and IGA censor concide.
- Data Complexity of BCQ entailment for *DL-Lite*:

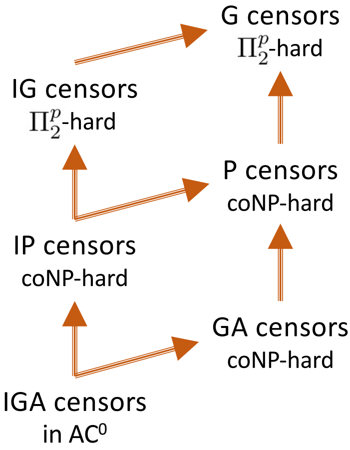
  — under P and IP censors is **coNP-hard**

  — under G and IG censor is Π_2^p**-hard**

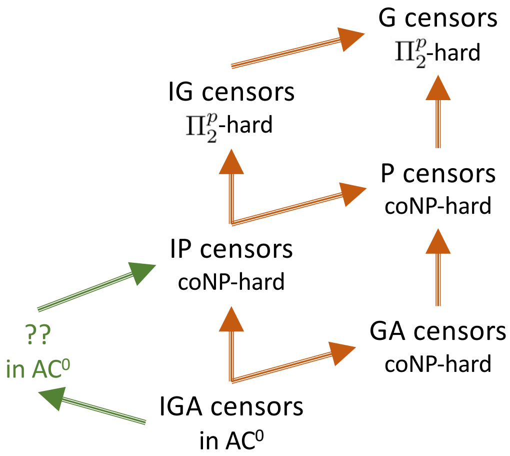
  (follow from *[Bienvenu and Bourgaux,KR 2020]*)

Relationship between censors

(and data complexity of BCQ entailment for *DL-Lite*)



An arrow from X to Y indicates that X is a sound approximation of Y, i.e, what is entailed under X is entailed also under Y



An arrow from X to Y indicates that X is a sound approximation of Y, i.e, what is entailed under X is entailed also under Y

DD-censors

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$

$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot; \quad \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= $\{\{$ TakesMedA(bob), Diabetic(bob) $\}, \{$ TakesMedB(bob), Diabetic(bob) $\}, \{$ TakesMedA(ann)), Diabetic(ann) $\}\}$

Priorities: TakesMedB$\succ$Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $\quad DC_1(\mathcal{E}_\succ) = \emptyset$

$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $\quad DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$

$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $\quad DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$

$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $\quad DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

DD-censors

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$
$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq$ ∃admissionWard; Diabetic $\sqsubseteq$ ∃admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot; \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

Priorities: TakesMedB≻Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $DC_1(\mathcal{E}_\succ) = \emptyset$
$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$
$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$
$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

DD-censors

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$
$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \land \mathsf{Diabetic}(x) \to \bot;\ \forall x.\mathsf{TakesMedB}(x) \land \mathsf{Diabetic}(x) \to \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

Priorities: TakesMedB≻Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $DC_1(\mathcal{E}_\succ) = \emptyset$

$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$

$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$

$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$
$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

Priorities: TakesMedB$\succ$Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $\quad DC_1(\mathcal{E}_\succ) = \emptyset$

$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $\quad DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$

$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $\quad DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$

$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $\quad DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

DD-censors

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$
$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot; \ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

Priorities: TakesMedB$\succ$Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $\quad DC_1(\mathcal{E}_\succ) = \emptyset$
$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $\quad DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$
$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $\quad DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$
$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $\quad DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

DD-censors

Given $\mathcal{E}_\succ = \langle \mathcal{T}, \mathcal{A}, \mathcal{P}, \succ \rangle$ we define $DD_0(\mathcal{E}_\succ) = DC_0(\mathcal{E}_\succ) = \emptyset$, and

$$DD_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{for each secret } \mathcal{S} \text{ s.t. } \alpha \in \mathcal{S} \text{ there is } \beta \in \mathcal{S} \text{ s.t. } \beta \neq \alpha \text{ and either } \alpha \succ \beta \text{ or } \beta \in DC_i(\mathcal{E}_\succ)\}$$
$$DC_{i+1}(\mathcal{E}_\succ) = \{\alpha \in \mathsf{cl}^{\mathcal{T}}_{\mathsf{GA}}(\mathcal{A}) \mid \text{there is a secret } \mathcal{S} \text{ s.t. } \mathcal{S} \setminus DD_i(\mathcal{E}_\succ) = \{\alpha\}\}$$

A Deterministically Disclosed (DD) censor for $\mathcal{E}_\succ$ returns the least fix point $DD(\mathcal{E}_\succ)$ for $DD_i(\mathcal{E}_\succ)$, which always exists and it is unique.

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq$ ∃admissionWard; Diabetic $\sqsubseteq$ ∃admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\mathsf{TakesMedA}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot;\ \forall x.\mathsf{TakesMedB}(x) \wedge \mathsf{Diabetic}(x) \rightarrow \bot \}$

Secrets= {{ TakesMedA(bob), Diabetic(bob) }, { TakesMedB(bob), Diabetic(bob) }, { TakesMedA(ann)), Diabetic(ann) }}

Priorities: TakesMedB≻Diabetic

$DD_1(\mathcal{E}_\succ) = \{\mathsf{TakesMedA(joe)}, \mathsf{TakesMedB(bob)}\}$ $\quad DC_1(\mathcal{E}_\succ) = \emptyset$
$DD_2(\mathcal{E}_\succ) = DD_1(\mathcal{E}_\succ)$ $\quad DC_2(\mathcal{E}_\succ) = \{\mathsf{Diabetic(bob)}\}$
$DD_3(\mathcal{E}_\succ) = DD_2(\mathcal{E}_\succ) \cup \{\mathsf{TakesMedA(bob)}\}$ $\quad DC_3(\mathcal{E}_\succ) = DC_2(\mathcal{E}_\succ)$
$DD_4(\mathcal{E}_\succ) = DD_3(\mathcal{E}_\succ)$ $\quad DC_4(\mathcal{E}_\succ) = DC_3(\mathcal{E}_\succ)$

from DD-censors to k-DD-censors

BCQ entailment under DD censors, i.e., deciding whether $\mathcal{T} \cup DD(\mathcal{E}_\succ) \models q$ for a BCQ q, is **PTIME-hard** in data complexity for ***DL-Lite***

By fixing a k, we get $DD_k(\mathcal{E}_\succ)$, which we call k-DD censor for $\mathcal{E}_\succ$.

Give a positive integer k, BCQ entailment under k-DD censors, i.e., deciding whether $\mathcal{T} \cup DD_k(\mathcal{E}_\succ) \models q$ for a BCQ q, is FO rewritable and thus in AC^0 in data complexity for *DL-Lite$_A$*

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.$TakesMedA$(x) \wedge$ Diabetic$(x) \rightarrow \bot$; $\forall x.$TakesMedB$(x) \wedge$ Diabetic$(x) \rightarrow \bot \}$

Priorities: TakesMedB$\succ$Diabetic

for $k = 3$ we have $DD_k(\mathcal{E}_\succ) = \{$TakesMedA(joe), TakesMedB(bob), TakesMedA(bob)$\}$

Given the CQ $q(x) : -$TakesMedA(x) and $k = 3$, $q(\vec{x})$ is reformulated in the FO query $q'(x) : -$TakesMedA$(x) \wedge (\neg$Diabetic$(x) \vee$ TakesMedB$(x))$. The evaluation of q' over $\mathcal{A}$ returns $\{$bob, joe$\}$, which is the set of answers under k-DD-censor for $k = 3$ (in this specific case this holds for any $k \geq 3$).

from DD-censors to k-DD-censors

BCQ entailment under DD censors, i.e., deciding whether $\mathcal{T} \cup DD(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is **PTIME-hard** in data complexity for ***DL-Lite***

By fixing a k, we get $DD_k(\mathcal{E}_{\succ})$, which we call k-DD censor for $\mathcal{E}_{\succ}$.

Give a positive integer k, BCQ entailment under k-DD censors, i.e., deciding whether $\mathcal{T} \cup DD_k(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is FO rewritable and thus in AC^0 in data complexity for *DL-Lite*$_A$

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{ \forall x.\text{TakesMedA}(x) \wedge \text{Diabetic}(x) \rightarrow \bot; \ \forall x.\text{TakesMedB}(x) \wedge \text{Diabetic}(x) \rightarrow \bot \}$

Priorities: TakesMedB$\succ$Diabetic

for $k = 3$ we have $DD_k(\mathcal{E}_{\succ}) = \{\text{TakesMedA(joe)}, \text{TakesMedB(bob)}, \text{TakesMedA(bob)}\}$

Given the CQ $q(x) :- \text{TakesMedA}(x)$ and $k = 3$, $q(\vec{x})$ is reformulated in the FO query $q'(x) :- \text{TakesMedA}(x) \wedge (\neg\text{Diabetic}(x) \vee \text{TakesMedB}(x))$. The evaluation of q' over $\mathcal{A}$ returns $\{\text{bob}, \text{joe}\}$, which is the set of answers under k-DD-censor for $k = 3$ (in this specific case this holds for any $k \geq 3$).

from DD-censors to k-DD-censors

BCQ entailment under DD censors, i.e., deciding whether $\mathcal{T} \cup DD(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is **PTIME-hard** in data complexity for ***DL-Lite***

By fixing a k, we get $DD_k(\mathcal{E}_{\succ})$, which we call k-DD censor for $\mathcal{E}_{\succ}$.

Give a positive integer k, BCQ entailment under k-DD censors, i.e., deciding whether $\mathcal{T} \cup DD_k(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is FO rewritable and thus in AC^0 in data complexity for *DL-Lite$_A$*

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq$ $\exists$admissionWard; Diabetic $\sqsubseteq$ $\exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)), Diabetic(ann), TakesMedA(joe) $\}$

$\mathcal{P} = \{$ $\forall x$.TakesMedA(x) $\wedge$ Diabetic(x) $\rightarrow \bot$; $\forall x$.TakesMedB(x) $\wedge$ Diabetic(x) $\rightarrow \bot$ $\}$

Priorities: TakesMedB$\succ$Diabetic

for $k = 3$ we have $DD_k(\mathcal{E}_{\succ}) = \{$TakesMedA(joe), TakesMedB(bob), TakesMedA(bob)$\}$

Given the CQ $q(x)$: $-$TakesMedA(x) and $k = 3$, $q(\vec{x})$ is reformulated in the FO query $q'(x)$: $-$TakesMedA(x) $\wedge$ ($\neg$Diabetic(x) $\vee$ TakesMedB(x)). The evaluation of q' over $\mathcal{A}$ returns $\{$bob, joe$\}$, which is the set of answers under k-DD-censor for $k = 3$ (in this specific case this holds for any $k \geq 3$).

from DD-censors to k-DD-censors

BCQ entailment under DD censors, i.e., deciding whether $\mathcal{T} \cup DD(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is **PTIME-hard** in data complexity for ***DL-Lite***

By fixing a k, we get $DD_k(\mathcal{E}_{\succ})$, which we call k-DD censor for $\mathcal{E}_{\succ}$.

Give a positive integer k, BCQ entailment under k-DD censors, i.e., deciding whether $\mathcal{T} \cup DD_k(\mathcal{E}_{\succ}) \models q$ for a BCQ q, is FO rewritable and thus in AC^0 in data complexity for *DL-Lite*$_A$

Example (cntd)

$\mathcal{T} = \{$ TakesMedB $\sqsubseteq \exists$admissionWard; Diabetic $\sqsubseteq \exists$admissionWard $\}$

$\mathcal{A} = \{$ TakesMedA(bob), TakesMedB(bob), Diabetic(bob), TakesMedA(ann)),
Diabetic(ann), TakesMedA(joe) $\}$

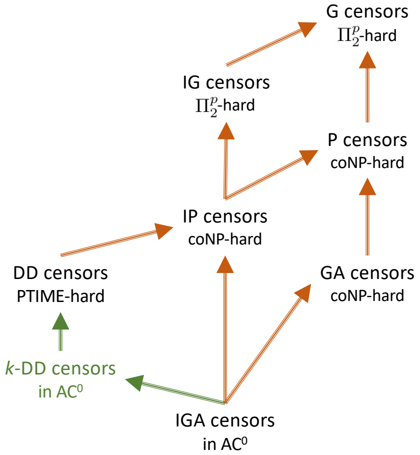
$\mathcal{P} = \{ \forall x.\text{TakesMedA}(x) \wedge \text{Diabetic}(x) \rightarrow \bot;\ \ \forall x.\text{TakesMedB}(x) \wedge \text{Diabetic}(x) \rightarrow \bot \}$

Priorities: TakesMedB$\succ$Diabetic

for $k = 3$ we have $DD_k(\mathcal{E}_{\succ}) = \{\text{TakesMedA(joe)}, \text{TakesMedB(bob)}, \text{TakesMedA(bob)}\}$

Given the CQ $q(x) :- \text{TakesMedA}(x)$ and $k = 3$, $q(\vec{x})$ is reformulated in the FO query $q'(x) :- \text{TakesMedA}(x) \wedge (\neg\text{Diabetic}(x) \vee \text{TakesMedB}(x))$. The evaluation of q' over $\mathcal{A}$ returns $\{\text{bob}, \text{joe}\}$, which is the set of answers under k-DD-censor for $k = 3$ (in this specific case this holds for any $k \geq 3$).

Completing the picture



An arrow from X to Y indicates that X is a sound approximation of Y, i.e, what is entailed under X is entailed also under Y

Experiments

- We implemented our technique and tested it over the **NPD** benchmark *[Lanti et al.,EDBT 2015]* (approximated to *DL-Lite*$_A$)
- We specified a policy $\mathcal{P}$ with 6 denials, a priority relation $\succ$ specifying 6 priorities, and selected 9 queries from the benchmark
- We executed each query in six settings: the first with empty policy and priority relation $(\emptyset, \emptyset)$, the second with policy $\mathcal{P}$ and no priorities $(\mathcal{P}, \emptyset)$, the others with policy $\mathcal{P}$, priorities in $\succ$, and $k = 1, 3, 5, 7$
- For $k \leq 3$, time is only slightly affected, and for 6 out of 9 queries we get already the same answers we obtain for $k > 3$. For $k > 3$, the gain in the number of answers is limited, and for 4 queries performances decrease.

	q_3 [5]		q_4 [4]		q_5 [6]		q_9 [5]		q_{12} [10]		q_{13} [7]		q_{14} [5]		q_{18} [9]		q_{44} [6]	
Setting	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**	**#**	**time**
$\emptyset, \emptyset$	910	207	1558	168	17254	585	1566	320	96671	5665	22541	811	141439	2553	339	1525	5078	221
$\mathcal{P}, \emptyset$	910	278	252	295	14797	825	416	331	13028	2876	9374	2861	62255	12372	311	1804	325	153
$\mathcal{P}, \succ, 1$	910	221	252	179	17254	612	416	216	96671	5933	22541	914	125656	4145	311	1384	325	112
$\mathcal{P}, \succ, 3$	910	249	521	1445	17254	749	1252	1148	96671	5378	22541	716	131791	15873	311	1416	4630	1952
$\mathcal{P}, \succ, 5$	910	242	566	8942	17254	723	1456	7715	96671	5219	22541	732	132127	1625K	311	4733	4630	522K
$\mathcal{P}, \succ, 7$	910	472	–	t.o.	17254	993	–	t.o.	96671	7691	22541	912	–	t.o.	311	5464	–	t.o.

Summary of results

- Our experiments show the applicability of our technique in practice
- Our results hold also for general CQs (i.e., non-Boolean ones)
- Our FO rewritability result holds for CQs with comparison predicates (provided that the $DL\text{-}Lite_A$ TBox satisfies a *safeness* condition)
- But: k-DD censor is only a **sound approximation** of the real (intractable) semantics of preferences

Outline

dynCQE: A "longest honeymoon" approach to CQE

In [ISWC 2022] we have proposed a "dynamic" approach to CQE called **dynCQE**

In dynCQE, the censors are progressively selected according to the user queries

We prove that this is the only possible approach that satisfies the "**longest-honeymoon**" property (*be honest as long as you can*)

Such a cooperative behavior:

- takes into account the user's interests
- allows for revealing more information than all the previous approaches

dynCQE - Example

Example (cont'd)

- $\mathcal{T} = \emptyset$, $\mathcal{A} = \{\ A(o), B(o)\ \}$, $\mathcal{P} = \{\ \exists x\,(A(x) \land B(x)) \rightarrow \bot\ \}$
- $OptCens(\mathcal{E}) = \{\ \{A(o)\}, \{B(o)\}\ \}$
- Given $q = A(o)$, we have that:
  - $\mathcal{T} \cup \{A(o)\} \models q$ (honest)
  - $\mathcal{T} \cup \{B(o)\} \not\models q$ (liar)
- No good reason to lie, but we have to "**remember**" that q was asked!

Example

Example

$$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Antiseizure}(y)) \to \bot,$$
$$\exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \to \bot\}$$
$$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$$
$$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$$

Example

Example

$$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Antiseizure}(y)) \rightarrow \bot,$$
$$\exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$$
$$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$$
$$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$$

$$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$$
$$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$$
$$\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$$
$$\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Antiseizure}(y)) \rightarrow \bot,$
$\quad \exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$
$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$
$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$
$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$
$\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$
$\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$q_1 = \mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1)$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Antiseizure}(y)) \to \bot,$
$\quad \exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \to \bot\}$
$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$
$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$
$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$
$\cancel{\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$
$\cancel{\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}}$

$q_1 = \mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1)$

$\Longrightarrow$ We have both that $\mathcal{T} \cup \mathcal{C}_1 \models q_1$ and $\mathcal{T} \cup \mathcal{C}_2 \models q_1$
The system answers $true$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Antiseizure}(y)) \rightarrow \bot,$
$\quad \exists x, y(\mathsf{Buy}(x, y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$
$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$
$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$
$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$
$\cancel{\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$
$\cancel{\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}}$

$q_2 = \mathsf{DrugA}(\mathsf{d}_1)$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x, y) \land \mathsf{Antiseizure}(y)) \rightarrow \bot,$
$\quad \exists x, y(\mathsf{Buy}(x, y) \land \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$

$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$

$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$

$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\cancel{\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$

$\cancel{\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}}$

$q_2 = \mathsf{DrugA}(\mathsf{d}_1)$

$\Longrightarrow$ No censor (among the surviving ones) entails q_2!
The system answers $false$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Antiseizure}(y)) \rightarrow \bot,$
$\quad \exists x, y(\mathsf{Buy}(x,y) \wedge \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$

$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$

$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}$

$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\cancel{\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$

$\cancel{\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}}$

$q_3 = \exists x\, \mathsf{Contains}(x, \mathsf{phenytoin})$

Example

Example

$\mathcal{P} = \{\exists x, y(\mathsf{Buy}(x, y) \land \mathsf{Antiseizure}(y)) \rightarrow \bot,$
$\quad \exists x, y(\mathsf{Buy}(x, y) \land \mathsf{Contains}(y, \mathsf{phenytoin})) \rightarrow \bot\}$

$\mathcal{T} = \{\mathsf{DrugA} \sqsubseteq \mathsf{Antiseizure}\}$

$\mathcal{A} = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{DrugA}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\cancel{\mathcal{C}_1 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$

$\mathcal{C}_2 = \{\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}$

$\cancel{\mathcal{C}_3 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Buy}(\mathsf{bob}, \mathsf{d}_2)\}}$

$\cancel{\mathcal{C}_4 = \{\mathsf{DrugA}(\mathsf{d}_1), \mathsf{Antiseizure}(\mathsf{d}_1), \mathsf{Contains}(\mathsf{d}_2, \mathsf{phenytoin})\}}$

$q_3 = \exists x\, \mathsf{Contains}(x, \mathsf{phenytoin})$

$\Longrightarrow$ We have that $\mathcal{C}_2 \models q_3$
The system answers $true$

Semantics

We keep track of the history of user queries through the notion of state:

State

A *state* of a CQE instance $\mathcal{E}$ is a pair $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ where $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of BUCQs

The set $StCens(\mathcal{S}_n)$ of ***censors of the state*** $\mathcal{S}_n = \langle \mathcal{E}, \mathcal{Q}_n \rangle$ is recursively defined as follows:

- $StCens(\mathcal{S}_0) = OptCens(\mathcal{E})$
- for $1 \leq i \leq n$:
  - $StCens(\mathcal{S}_i) = StCens(\mathcal{S}_{i-1})$, if $\nexists \mathcal{C} \in StCens(\mathcal{S}_{i-1})$ s.t. $\mathcal{T} \cup \mathcal{C} \models q_i$
  - $StCens(\mathcal{S}_i) = \{\mathcal{C} \in StCens(\mathcal{S}_{i-1}) \mid \mathcal{T} \cup \mathcal{C} \models q_i\}$, otherwise

We say that the state $\mathcal{S}_n$ ***entails*** q_i ($\mathcal{S}_n \models q_i$) if $\mathcal{T} \cup \mathcal{C} \models q_i$ for all the censors $\mathcal{C} \in StCens(\mathcal{S}_n)$

Semantics

We keep track of the history of user queries through the notion of state:

State

A *state* of a CQE instance $\mathcal{E}$ is a pair $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ where $\mathcal{Q} = \langle q_1, \ldots, q_n \rangle$ is a sequence of BUCQs

The set $StCens(\mathcal{S}_n)$ of ***censors of the state*** $\mathcal{S}_n = \langle \mathcal{E}, \mathcal{Q}_n \rangle$ is recursively defined as follows:

- $StCens(\mathcal{S}_0) = OptCens(\mathcal{E})$
- for $1 \leq i \leq n$:
  - $StCens(\mathcal{S}_i) = StCens(\mathcal{S}_{i-1})$, if $\nexists \mathcal{C} \in StCens(\mathcal{S}_{i-1})$ s.t. $\mathcal{T} \cup \mathcal{C} \models q_i$
  - $StCens(\mathcal{S}_i) = \{\mathcal{C} \in StCens(\mathcal{S}_{i-1}) \mid \mathcal{T} \cup \mathcal{C} \models q_i\}$, otherwise

We say that the state $\mathcal{S}_n$ ***entails*** q_i ($\mathcal{S}_n \models q_i$) if $\mathcal{T} \cup \mathcal{C} \models q_i$ for all the censors $\mathcal{C} \in StCens(\mathcal{S}_n)$

Maximal cooperativity

Maximal cooperativity

We say that $\mathcal{C}$ is ***maximally cooperative*** w.r.t. $\mathcal{Q}$ if there do not exist a censor $\mathcal{C}'$ and a number m s.t.:

- $\mathcal{T} \cup \mathcal{C} \models q_i \iff \mathcal{T} \cup \mathcal{C}' \models q_i$ for every $1 \leq i \leq m$, and
- $\mathcal{T} \cup \mathcal{C} \not\models q_{m+1}$ and $\mathcal{T} \cup \mathcal{C}' \models q_{m+1}$.

Theorem

Every censor for $\mathcal{E}$ is maximally cooperative w.r.t. $\mathcal{Q}$ iff $\mathcal{C} \in StCens(\langle \mathcal{E}, \mathcal{Q} \rangle)$

Maximal cooperativity

Maximal cooperativity

We say that $\mathcal{C}$ is ***maximally cooperative*** w.r.t. $\mathcal{Q}$ if there do not exist a censor $\mathcal{C}'$ and a number m s.t.:

- $\mathcal{T} \cup \mathcal{C} \models q_i \iff \mathcal{T} \cup \mathcal{C}' \models q_i$ for every $1 \leq i \leq m$, and
- $\mathcal{T} \cup \mathcal{C} \not\models q_{m+1}$ and $\mathcal{T} \cup \mathcal{C}' \models q_{m+1}$.

Theorem

Every censor for $\mathcal{E}$ is maximally cooperative w.r.t. $\mathcal{Q}$ iff $\mathcal{C} \in StCens(\langle \mathcal{E}, \mathcal{Q} \rangle)$

Query entailment problem

We study the data complexity of the following decision problem for OWL 2 QL ontologies:

Query entailment in a state

Given a state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ and a BUCQ $q \in \mathcal{Q}$, check whether $\mathcal{S} \models q$

We proved that the above problem is **FO-rewritable**, i.e. it is always possible to reformulate q into a FO query q_r such that $\mathcal{S} \models q$ iff $\mathcal{A} \models q_r$

- No need to materialize censors
- Problem belongs to AC^0 class in data complexity

Query entailment problem

We study the data complexity of the following decision problem for OWL 2 QL ontologies:

Query entailment in a state

Given a state $\mathcal{S} = \langle \mathcal{E}, \mathcal{Q} \rangle$ and a BUCQ $q \in \mathcal{Q}$, check whether $\mathcal{S} \models q$

We proved that the above problem is **FO-rewritable**, i.e. it is always possible to reformulate q into a FO query q_r such that $\mathcal{S} \models q$ iff $\mathcal{A} \models q_r$

- No need to materialize censors
- Problem belongs to AC^0 class in data complexity

First-order rewriting example

Example (cont'd)

The query $q_3 = \exists x\, \mathsf{Contains}(x, \mathsf{phenytoin})$ can be rewritten as:

$$
\begin{aligned}
q_r = {} & \neg \mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1) \wedge \neg \mathsf{DrugA}(\mathsf{d}_1) \wedge \exists x\, \mathsf{Contains}(x, \mathsf{phenytoin}) \vee \\
& \neg(\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1) \wedge \mathsf{DrugA}(\mathsf{d}_1) \wedge \neg(\top)) \wedge \exists x\, (\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1) \wedge \\
& \qquad \mathsf{Contains}(x, \mathsf{phenytoin}) \wedge \neg(\mathsf{d}_1 = x)) \vee \\
& \neg \mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1) \wedge (\exists x\, (\mathsf{DrugA}(\mathsf{d}_1) \wedge \mathsf{Contains}(x, \mathsf{phenytoin}))) \vee \\
& \exists x (\mathsf{Buy}(\mathsf{ann}, \mathsf{d}_1) \wedge \mathsf{DrugA}(\mathsf{d}_1) \wedge \mathsf{Contains}(x, \mathsf{phenytoin})) \wedge \neg(\top \vee \mathsf{d}_1 = x))
\end{aligned}
$$

Due to the second disjunct, we have that q_r evaluates to true in $\mathcal{A}$ (indeed we had that $\mathcal{S} \models q_3$)

Outline

1. The Controlled Query Evaluation approach in Ontologies and Description Logics
   - CQE in Description Logics through GA censors
   - Computational problems
2. Towards tractability 1: Intersecting the censors
   - IGA censors
   - Expressive limitations of IGA censors
3. Towards tractability 2: Adding preferences
   - Globally optimal and Pareto-optimal censors
   - DD and k-DD censors
   - Experimental results
4. Towards tractability 3: Maximally cooperative approach
   - The dynCQE approach
   - Complexity of dynCQE
5. Conclusions

Main open problems

- Extend the results to **different knowledge bases/databases** (beyond DLs)
- Improve the **user/attacker model**
- Increase the **policy language**
- Increase the **query language**
- Optimized practical **algorithms**
- More powerful notions of **censor** (not only tuple-deletion based)

Our ongoing work

- Extend the dynCQE framework to **non-Boolean UCQs**
- **Implement** dynCQE (as we did for previous frameworks)
- Extend the **policy language** (as done for static CQE)
- Refine the current framework for allowing to **delete specific cells** of a table instead of full tuples

Publications

- D. Lembo, R. Rosati and D. F. Savo. Revisiting Controlled Query Evaluation in Description Logics. *International Joint Conference on Artificial Intelligence (IJCAI), 2019*.
- G. Cima, D. Lembo, R. Rosati and D. F. Savo. Controlled Query Evaluation in Description Logics Through Instance Indistinguishability. *International Joint Conference on Artificial Intelligence (IJCAI), 2020*.

Publications (cont'd)

- G. Cima, D. Lembo, L. Marconi, R. Rosati and D. F. Savo. Controlled Query Evaluation over Prioritized Ontologies with Expressive Data Protection Policies. Controlled Query Evaluation in Ontology-Based Data Access. *International Semantic Web Conference (ISWC), 2020*.
- G. Cima, D. Lembo, L. Marconi, R. Rosati and D. F. Savo. Controlled Query Evaluation over Prioritized Ontologies with Expressive Data Protection Policies. In Fourth IEEE International Conference on Artificial Intelligence and Knowledge Engineering, *International Semantic Web Conference (ISWC), 2021*.
- P. Bonatti, G. Cima, D. Lembo, L. Marconi, R. Rosati, L. Sauro and D. F. Savo. Controlled Query Evaluation in OWL 2 QL: A "Longest Honeymoon" Approach. *International Semantic Web Conference (ISWC), 2022*.