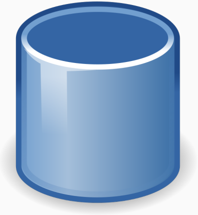
Reasoning about Disclosure in Data Integration in the Presence of Source Constraints

DIG Seminar 21/11/19

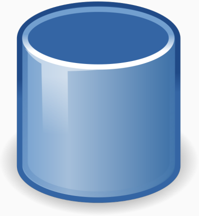
Michael Benedikt Pierre Bourhis Louis Jachiet Michaël Thomazo

Motivations

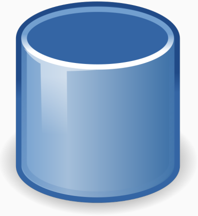


Motivations

Schema+Constraints



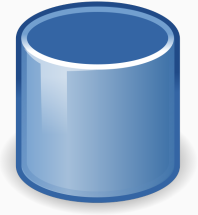
Motivations



$\hookrightarrow$

publication

Motivations

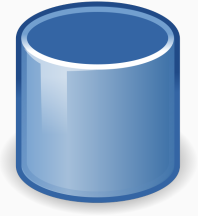


Secret

$\hookrightarrow$

publication

Motivations



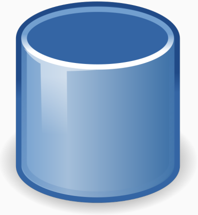
$\hookrightarrow$

publication

Secret

Secret leaked?

Motivations



Secret

$\hookrightarrow$

Safe publication?

Secret leaked?

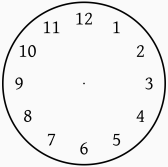
Example: Hospital setting

Patients

Doctors

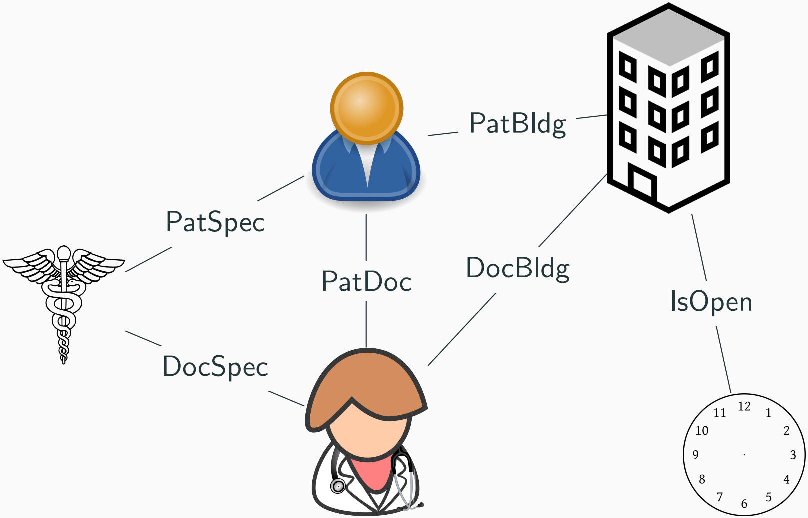
Buildings

Specialties



Open hours

Example: Hospital setting



Example

Database schema

Predicate	Meaning
IsOpen(b, t)	Building b is open on Date t
PatBdlg(p, b)	Patient p is present in Building b
PatSpec(p, s)	Patient p was treated for Specialty s
PatDoc(p, d)	Patient p was treated by Doctor d
DocBldg(d, b)	Doctor d is associated with Building b
DocSpec(d, s)	Doctor d is associated with Specialty s

Example

Views

$$
\begin{aligned}
\texttt{OpenHours}(b, t) &= \texttt{IsOpen}(b, t) \\
\texttt{VisitingHours}(p, t) &= \texttt{PatBdlg}(p, b) \wedge \texttt{IsOpen}(b, t) \\
\texttt{DocList}(d, s, b) &= \texttt{DocSpec}(d, s) \wedge \texttt{DocBldg}(d, b)
\end{aligned}
$$

Example

Views

$$\begin{aligned} \texttt{OpenHours}(b,t) &= \texttt{IsOpen}(b,t) \\ \texttt{VisitingHours}(p,t) &= \texttt{PatBdlg}(p,b) \wedge \texttt{IsOpen}(b,t) \\ \texttt{DocList}(d,s,b) &= \texttt{DocSpec}(d,s) \wedge \texttt{DocBldg}(d,b) \end{aligned}$$

Constraints

$$\begin{aligned} \texttt{PatDoc}(p,d) &\rightarrow \exists s\ \texttt{PatSpec}(p,s) \wedge \texttt{DocSpec}(d,s) \\ \texttt{PatBdlg}(p,b) &\rightarrow \exists d\ \texttt{PatDoc}(p,d) \wedge \texttt{DocBldg}(d,b) \end{aligned}$$

Example

Views

$$
\begin{aligned}
\texttt{OpenHours}(b,t) &= \texttt{IsOpen}(b,t)\\
\texttt{VisitingHours}(p,t) &= \texttt{PatBdlg}(p,b) \wedge \texttt{IsOpen}(b,t)\\
\texttt{DocList}(d,s,b) &= \texttt{DocSpec}(d,s) \wedge \texttt{DocBldg}(d,b)
\end{aligned}
$$

Constraints

$$
\begin{aligned}
\texttt{PatDoc}(p,d) &\rightarrow \exists s\ \texttt{PatSpec}(p,s) \wedge \texttt{DocSpec}(d,s)\\
\texttt{PatBdlg}(p,b) &\rightarrow \exists d\ \texttt{PatDoc}(p,d) \wedge \texttt{DocBldg}(d,b)
\end{aligned}
$$

Secret

$$\exists p,s\ \ \texttt{PatSpec}(p,s)?$$

Example

OpenHour	
B_1	Tuesday
B_2	Every day 10-17h

VisitingHours	
Charline	Tuesday

DocList		
Alice	Cancer	B_1
Alice	Cancer	B_2
Bob	Radiology	B_2
Daniel	Cancer	B_1

Views

$$\begin{aligned}
\texttt{OpenHours}(b,t) &= \texttt{IsOpen}(b,t) \\
\texttt{VisitingHours}(p,t) &= \texttt{PatBdlg}(p,b) \wedge \texttt{IsOpen}(b,t) \\
\texttt{DocList}(d,s,b) &= \texttt{DocSpec}(d,s) \wedge \texttt{DocBldg}(d,b)
\end{aligned}$$

Example

OpenHour

B_1	Tuesday
B_2	Every day 10-17h

VisitingHours

Charline	Tuesday

DocList

Alice	Cancer	B_1
Alice	Cancer	B_2
Bob	Radiology	B_2
Daniel	Cancer	B_1

Views

$$\begin{aligned}\texttt{OpenHours}(b,t) &= \texttt{IsOpen}(b,t)\\ \texttt{VisitingHours}(p,t) &= \texttt{PatBdlg}(p,b) \wedge \texttt{IsOpen}(b,t)\\ \texttt{DocList}(d,s,b) &= \texttt{DocSpec}(d,s) \wedge \texttt{DocBldg}(d,b)\end{aligned}$$

Example

OpenHour

B_1	Tuesday
B_2	Every day 10-17h

VisitingHours

Charline	Tuesday

DocList

Alice	Cancer	B_1
Alice	Cancer	B_2
Bob	Radiology	B_2
Daniel	Cancer	B_1

Constraints

$$\texttt{PatBdlg}(p,b) \quad \rightarrow \quad \exists d\ \texttt{PatDoc}(p,d) \wedge \texttt{DocBldg}(d,b)$$

Views

$$\begin{aligned}
\texttt{OpenHours}(b,t) &= \texttt{IsOpen}(b,t) \\
\texttt{VisitingHours}(p,t) &= \texttt{PatBdlg}(p,b) \wedge \texttt{IsOpen}(b,t) \\
\texttt{DocList}(d,s,b) &= \texttt{DocSpec}(d,s) \wedge \texttt{DocBldg}(d,b)
\end{aligned}$$

Example

OpenHour	
B_1	Tuesday
B_2	Every day 10-17h

VisitingHours	
Charline	Tuesday

DocList		
Alice	Cancer	B_1
Alice	Cancer	B_2
Bob	Radiology	B_2
Daniel	Cancer	B_1

Constraints

$$\mathtt{PatBdlg}(p, b) \quad \rightarrow \quad \exists d\ \mathtt{PatDoc}(p, d) \wedge \mathtt{DocBldg}(d, b)$$

Views

$$\begin{aligned} \mathtt{OpenHours}(b, t) &= \mathtt{IsOpen}(b, t) \\ \mathtt{VisitingHours}(p, t) &= \mathtt{PatBdlg}(p, b) \wedge \mathtt{IsOpen}(b, t) \\ \mathtt{DocList}(d, s, b) &= \mathtt{DocSpec}(d, s) \wedge \mathtt{DocBldg}(d, b) \end{aligned}$$

Example

OpenHour	
B_1	Tuesday
B_2	Every day 10-17h

VisitingHours	
Charline	Tuesday

DocList		
Alice	Cancer	B_1
Alice	Cancer	B_2
Bob	Radiology	B_2
Daniel	Cancer	B_1

Constraints

$$\begin{aligned} \mathtt{PatBdlg}(p,b) &\rightarrow \exists d\ \mathtt{PatDoc}(p,d) \wedge \mathtt{DocBldg}(d,b) \\ \mathtt{PatDoc}(p,d) &\rightarrow \exists s\ \mathtt{PatSpec}(p,s) \wedge \mathtt{DocSpec}(d,s) \end{aligned}$$

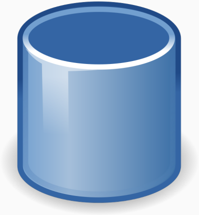
Views

$$\begin{aligned} \mathtt{OpenHours}(b,t) &= \mathtt{IsOpen}(b,t) \\ \mathtt{VisitingHours}(p,t) &= \mathtt{PatBdlg}(p,b) \wedge \mathtt{IsOpen}(b,t) \\ \mathtt{DocList}(d,s,b) &= \mathtt{DocSpec}(d,s) \wedge \mathtt{DocBldg}(d,b) \end{aligned}$$

Formalism

Data represented by databases

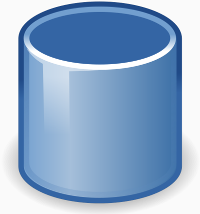
$R(1,17), R(2,42), S(23,45), \ldots$



Formalism

Data represented by databases

$R(1, 17), R(2, 42), S(23, 45), \ldots$



$\hookrightarrow$ + secret

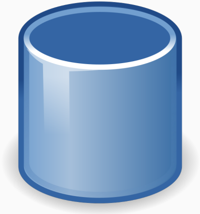
Mappings and secrets are CQ

$V(x, z) := R(x, y) \wedge S(y, z)$

Formalism

Data represented by databases

$R(1, 17), R(2, 42), S(23, 45), \ldots$



$\hookrightarrow$ + secret

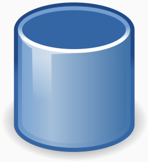
Mappings and secrets are CQ

$V(x, z) := R(x, y) \wedge S(y, z)$

Constraints are TGD

$R(x, y) \rightarrow \exists z, S(y, z)$

Safe publication



$\hookrightarrow$

Secret

Safe publication

View Problem

Given (schema, constraints $\mathcal{C}$, views $\mathcal{V}$, secret $\mathcal{S}$, visible ) do we have  such that $\mathcal{C}($$)$, $\mathcal{V}($$) =$  and $\neg\mathcal{S}($$)$?

Schema Problem

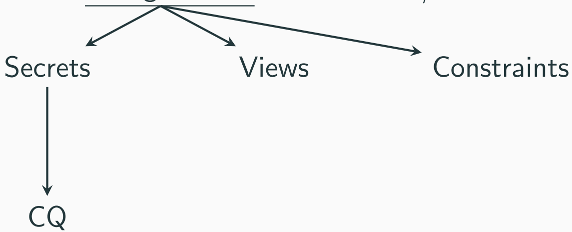
Given (schema, constraints $\mathcal{C}$, views $\mathcal{V}$, secret $\mathcal{S}$) do we have for all  an instance  such that $\mathcal{C}($$)$, $\mathcal{V}($$) = \mathcal{V}($$)$ and $\neg\mathcal{S}($$)$?

Which configurations are decidable/tractable for the schema problem?

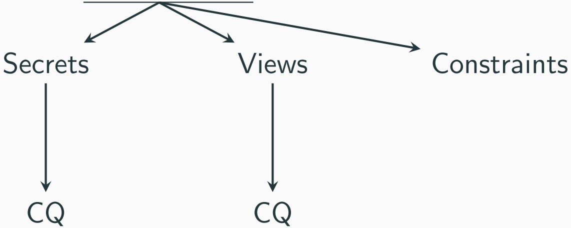
Which configurations are decidable/tractable for the schema problem?

Secrets Views Constraints

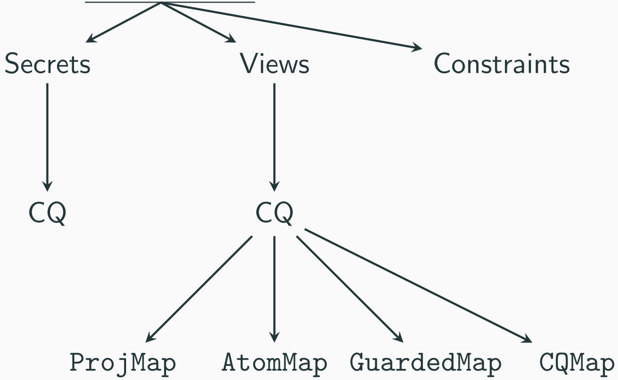
Which configurations are decidable/tractable for the schema problem?



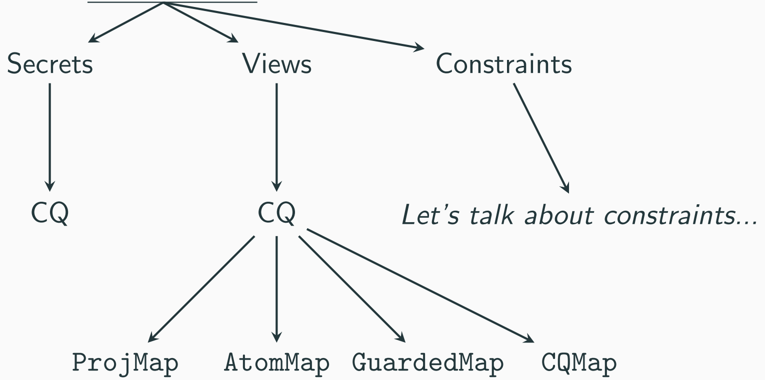
Which configurations are decidable/tractable for the schema problem?



Which configurations are decidable/tractable for the schema problem?



Which configurations are decidable/tractable for the schema problem?



Ontologies

Ontologies in a few words

An ontology represents entities and their relationship to each other.

Ontologies in a few words

An ontology represents entities and their relationship to each other.

Ontologies can be seen as a sort of expressive schema.

Ontologies in a few words

An ontology represents entities and their relationship to each other.

Ontologies can be seen as a sort of expressive schema.

Ontologies allows to enrich data by inferring new facts from existing ones.

An example of ontology

With plain words:

All cats are mammals.

All mammals are animals.

An example of ontology

With plain words:

All cats are mammals.

All mammals are animals.

With Tuple Generating Dependencies:

$$CAT(x) \rightarrow MAMMAL(x)$$

$$MAMMAL(x) \rightarrow ANIMAL(x)$$

An example of ontology

With plain words:

All cats are mammals.

All mammals are animals.

Database: $\{CAT($  $)\}$

An example of ontology

With plain words:

All cats are mammals.

All mammals are animals.

Database: $\{CAT($  $)\}$
Query: Are there animals? $(\exists X, ANIMAL(X)?)$

An example of ontology

With plain words:

All cats are mammals.

All mammals are animals.

Database: $\{CAT($  $)\}$

Query: Are there animals? $(\exists X, ANIMAL(X)?)$

Answer: Yes: $CAT($  $) \Rightarrow MAMMAL($  $) \Rightarrow ANIMAL($  $)$

More complex ontological rules

Foreign key constraint:

$$SEMINAR(team, speaker, room, date) \rightarrow$$

$$\exists pers, RESERVED(room, date, pers)$$

More complex ontological rules

Foreign key constraint:

$$SEMINAR(team, speaker, room, date) \rightarrow$$
$$\exists pers, RESERVED(room, date, pers)$$

Even more complex constraints:

$$SEMINAR(team, speaker, room, date) \rightarrow$$
$$\exists pers, RESERVED(room, date, pers) \wedge MEMBER(pers, team)$$

More complex ontological rules

Foreign key constraint:

$$SEMINAR(team, speaker, room, date) \rightarrow$$
$$\exists pers, RESERVED(room, date, pers)$$

Even more complex constraints:

$$SEMINAR(team, speaker, room, date) \rightarrow$$
$$\exists pers, RESERVED(room, date, pers) \wedge MEMBER(pers, team)$$

$$MEMBER(person, team) \rightarrow$$
$$\exists date, room,\ SEMINAR(team, person, room, date)$$

Tuple Generating Dependencies

$$\forall \vec{X}, \vec{Y} \quad \varphi(\vec{X}, \vec{Y}) \quad \Rightarrow \quad \exists \vec{Z} \quad \psi(\vec{Y}, \vec{Z})$$

Tuple Generating Dependencies

$$\forall \vec{X}, \vec{Y} \quad \varphi(\vec{X}, \vec{Y}) \quad \Rightarrow \quad \exists \vec{Z} \quad \psi(\vec{Y}, \vec{Z})$$

Often Omitted

Tuple Generating Dependencies

Body $\downarrow$ Head $\downarrow$

$$\varphi(\vec{X}, \vec{Y}) \quad \Rightarrow \quad \exists \vec{Z} \quad \psi(\vec{Y}, \vec{Z})$$

Tuple Generating Dependencies

$$\varphi(\vec{X}, \vec{Y}) \quad \Rightarrow \quad \exists \vec{Z} \quad \psi(\vec{Y}, \vec{Z})$$

Frontier

Open World Query Answering

Open World Query Answering

- A set of facts $\mathcal{F}$
- A set of TGD constraints $\mathcal{C}$
- A conjunctive query $\mathcal{Q}$

Do we have, for all :

$(\mathcal{F} \subseteq$  $\wedge\, \mathcal{C}($  $)) \Rightarrow \mathcal{Q}($  $)$?

Open World Query Answering

OWQA($\mathcal{F}, \mathcal{C}, \mathcal{Q}$) asks if $\mathcal{Q}$ is true in all completions of $\mathcal{F}$ (respecting $\mathcal{C}$).

Open World Query Answering

OWQA($\mathcal{F}, \mathcal{C}, \mathcal{Q}$) asks if $\mathcal{Q}$ is true in all completions of $\mathcal{F}$ (respecting $\mathcal{C}$).

The Chase algorithms build a universal model.

Open World Query Answering

OWQA($\mathcal{F}, \mathcal{C}, \mathcal{Q}$) asks if $\mathcal{Q}$ is true in all completions of $\mathcal{F}$ (respecting $\mathcal{C}$).

The Chase algorithms build a universal model.

$$\mathit{OWQA}(\mathcal{F}, \mathcal{C}, \mathcal{Q}) \Leftrightarrow \mathit{Chase}(\mathcal{F}, \mathcal{C}) \vDash \mathcal{Q}$$

The Chase

Intuitively the chase simply “applies” the constraints.

The Chase

Intuitively the chase simply "applies" the constraints.

With:

- $\mathcal{F} = CAT($  $)$
- $\mathcal{C} = \{CAT(X) \rightarrow MAMMAL(X), MAMMAL(X) \rightarrow ANIMAL(X)\}$

We obtain:

1. $\mathcal{F}_1 = \{CAT($  $)\}$

The Chase

Intuitively the chase simply "applies" the constraints.

With:

- $\mathcal{F} = CAT($  $)$
- $\mathcal{C} = \{CAT(X) \rightarrow MAMMAL(X), MAMMAL(X) \rightarrow ANIMAL(X)\}$

We obtain:

1. $\mathcal{F}_1 = \{CAT($  $)\}$
2. $\mathcal{F}_2 = \{CAT($  $), MAMMAL($  $)\}$

Intuitively the chase simply “applies” the constraints.

With:

- $\mathcal{F} = CAT($  $)$
- $\mathcal{C} = \{CAT(X) \rightarrow MAMMAL(X), MAMMAL(X) \rightarrow ANIMAL(X)\}$

We obtain:

1. $\mathcal{F}_1 = \{CAT($  $)\}$
2. $\mathcal{F}_2 = \{CAT($  $), MAMMAL($  $)\}$
3. $\mathcal{F}_3 = \{CAT($  $), MAMMAL($  $), ANIMAL($  $)\}$

The Chase

Intuitively the chase simply “applies” the constraints.

The Chase

Intuitively the chase simply "applies" the constraints.

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$

The Chase

Intuitively the chase simply “applies” the constraints.

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$

The Chase

Intuitively the chase simply "applies" the constraints.

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$

The Chase

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)$
  $PARENT(X, Y) \rightarrow \exists PERSON(Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$

The Chase

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)$
  $PARENT(X, Y) \rightarrow \exists PERSON(Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)$
  $PARENT(X, Y) \rightarrow \exists PERSON(Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$
3. $\mathcal{F}_3 = \{PERSON(alice), PARENT(alice, Y), PERSON(Y)\}$

The Chase

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)$
  $PARENT(X, Y) \rightarrow \exists PERSON(Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$
3. $\mathcal{F}_3 = \{PERSON(alice), PARENT(alice, Y), PERSON(Y)\}$
4. $\mathcal{F}_4 = \{PERSON(alice), PARENT(alice, Y), PERSON(Y), PARENT(Y, Y')\}$

With:

- $\mathcal{F} = \{PERSON(alice)\}$
- $\mathcal{C} = \{PERSON(X) \rightarrow \exists Y, PARENT(X, Y)$
  $PARENT(X, Y) \rightarrow \exists PERSON(Y)\}$

We obtain:

1. $\mathcal{F}_1 = \{PERSON(alice)\}$
2. $\mathcal{F}_2 = \{PERSON(alice), PARENT(alice, Y)\}$
3. $\mathcal{F}_3 = \{PERSON(alice), PARENT(alice, Y), PERSON(Y)\}$
4. $\mathcal{F}_4 = \{PERSON(alice), PARENT(alice, Y), PERSON(Y), PARENT(Y, Y')\}$

   ...

The Chase

The Chase model is not always finite.

The Chase model is not always finite.

When it is not finite it sometimes has a regularity that allows for decidable OWQA.

The Chase

The Chase model is not always finite.

When it is not finite it sometimes has a regularity that allows for decidable OWQA.

And in general the OWQA is undecidable. . .

- `UID`, the foreign key constraint with one variable

$$A(x, \vec{Y}) \rightarrow \exists Z, B(x, \vec{Z})$$

Decidable Classes of TGD

- `UID`, the foreign key constraint with one variable

$$A(x, \vec{Y}) \rightarrow \exists Z, B(x, \vec{Z})$$

- `IncDep`, the foreign key constraint

$$A(\vec{X}, \vec{Y}) \rightarrow \exists Z, B(\vec{X}, \vec{Z})$$

Decidable Classes of TGD

- `UID`, the foreign key constraint with one variable

$$A(x, \vec{Y}) \to \exists Z, B(x, \vec{Z})$$

- `IncDep`, the foreign key constraint

$$A(\vec{X}, \vec{Y}) \to \exists Z, B(\vec{X}, \vec{Z})$$

- `LTGD`, the foreign key constraint with repetition of atoms

$$A(x, x, y) \to \exists Z, B(x, y, y, z)$$

Decidable Classes of TGD

- `UID`, the foreign key constraint with one variable

$$A(x, \vec{Y}) \to \exists Z, B(x, \vec{Z})$$

- `IncDep`, the foreign key constraint

$$A(\vec{X}, \vec{Y}) \to \exists Z, B(\vec{X}, \vec{Z})$$

- `LTGD`, the foreign key constraint with repetition of atoms

$$A(x, x, y) \to \exists Z, B(x, y, y, z)$$

- `GTGD`, one atom in the body guards all variables

$$A(x, y, z) \wedge B(x) \wedge C(y, z) \to \exists w, D(x, y, w)$$

Decidable Classes of TGD

- `UID`, the foreign key constraint with one variable
$$A(x, \vec{Y}) \rightarrow \exists Z, B(x, \vec{Z})$$
- `IncDep`, the foreign key constraint
$$A(\vec{X}, \vec{Y}) \rightarrow \exists Z, B(\vec{X}, \vec{Z})$$
- `LTGD`, the foreign key constraint with repetition of atoms
$$A(x, x, y) \rightarrow \exists Z, B(x, y, y, z)$$
- `GTGD`, one atom in the body guards all variables
$$A(x, y, z) \wedge B(x) \wedge C(y, z) \rightarrow \exists w, D(x, y, w)$$
- `FGTGD`, one atom in the body guards all the frontier variables
$$A(w, y, y) \wedge B(x) \wedge C(y, z) \rightarrow \exists u, D(x, y, u)$$

Decidable Classes of TGD

- `UID`, the foreign key constraint with one variable

$$A(x, \vec{Y}) \to \exists Z, B(x, \vec{Z})$$

- `IncDep`, the foreign key constraint

$$A(\vec{X}, \vec{Y}) \to \exists Z, B(\vec{X}, \vec{Z})$$

- `LTGD`, the foreign key constraint with repetition of atoms

$$A(x, x, y) \to \exists Z, B(x, y, y, z)$$

- `GTGD`, one atom in the body guards all variables

$$A(x, y, z) \wedge B(x) \wedge C(y, z) \to \exists w, D(x, y, w)$$

- `FGTGD`, one atom in the body guards all the frontier variables

$$A(w, y, y) \wedge B(x) \wedge C(y, z) \to \exists u, D(x, y, u)$$

- `Fr1LTGD`, one atom in the body guards the only frontier variable

$$A(w, y, y) \wedge B(x) \wedge C(y, z) \to \exists u, D(x, u)$$

Another approach: query rewriting

With

- $MAMMAL(X) \rightarrow ANIMAL(X)$
- $CAT(X) \rightarrow MAMMAL(X)$

And the query $\exists X, ANIMAL(X)$, we obtain:

- $ANIMAL(X)$

Another approach: query rewriting

With

- $MAMMAL(X) \to ANIMAL(X)$
- $CAT(X) \to MAMMAL(X)$

And the query $\exists X, ANIMAL(X)$, we obtain:

- $ANIMAL(X)$
- $ANIMAL(X) \vee MAMMAL(X)$

Another approach: query rewriting

With

- $MAMMAL(X) \rightarrow ANIMAL(X)$
- $CAT(X) \rightarrow MAMMAL(X)$

And the query $\exists X, ANIMAL(X)$, we obtain:

- $ANIMAL(X)$
- $ANIMAL(X) \vee MAMMAL(X)$
- $ANIMAL(X) \vee MAMMAL(X) \vee CAT(X)$

Solving our problem

Back to our problems

View Problem

Given (schema, constraints $\mathcal{C}$, views $\mathcal{V}$, secret $\mathcal{S}$, visible) do we have such that $\mathcal{C}($ $)$, $\mathcal{V}($ $) =$ and $\neg\mathcal{S}($ $)$?

Schema Problem

Given (schema, constraints $\mathcal{C}$, views $\mathcal{V}$, secret $\mathcal{S}$) do we have for all an instance such that $\mathcal{C}($ $)$, $\mathcal{V}($ $) = \mathcal{V}($ $)$ and $\neg\mathcal{S}($ $)$?

The critical instance

The instance  $_{\mathtt{C}}$ contains one fact per relation, with one constant: C.

The critical instance

The instance $_{\mathtt{c}}$ contains one fact per relation, with one constant: $\mathtt{c}$. We note $_{\mathtt{c}} = \mathcal{V}($$_{\mathtt{c}})$ its view image.

Solving the schema problem

The critical instance

The instance $_{\texttt{C}}$ contains one fact per relation, with one constant: C. We note $_{\texttt{C}} = \mathcal{V}($$_{\texttt{C}})$ its view image.

Reduction for schema problem

SchemaProblem$(\mathcal{C}, \mathcal{V}, \mathcal{S})$ reduces to *ViewProblem*$($$_{\texttt{C}}, \mathcal{C}, \mathcal{V}, \mathcal{S})$

From Querying Visible and Invisible Information. LICS 2016

Open World Query Answering

- A set of facts $\mathcal{F}$
- A set of TGD constraints $\mathcal{C}$
- A query $\mathcal{Q}$

Do we have, for all :

$(\mathcal{F} \subseteq$  $\wedge \mathcal{C}($  $)) \Rightarrow \mathcal{Q}($  $)$?

Encoding *ViewProblem*($_{\mathtt{C}}, \mathcal{C}, \mathcal{V}, \mathcal{S}$) **as OWQA**

Reduction to Open World Query Answering

Encoding *ViewProblem*($_{\mathrm{C}}, \mathcal{C}, \mathcal{V}, \mathcal{S}$) **as OWQA**

- The query is $\mathcal{S}$

Encoding *ViewProblem*($_C, \mathcal{C}, \mathcal{V}, \mathcal{S})$ **as OWQA**

- The query is $\mathcal{S}$

- The initial facts encode the forward constraints

$_C \subseteq \mathcal{V}($$)$

Reduction to Open World Query Answering

Encoding *ViewProblem*($_{\mathrm{C}}, \mathcal{C}, \mathcal{V}, \mathcal{S}$) **as OWQA**

- The query is $\mathcal{S}$

- The initial facts encode the forward constraints

 $_{\mathcal{C}} \subseteq \mathcal{V}($$)$

- The constraints are the original constraints $\mathcal{C}$.

Reduction to Open World Query Answering

Encoding *ViewProblem*($_{\mathrm{C}}, \mathcal{C}, \mathcal{V}, \mathcal{S}$) **as OWQA**

- The query is $\mathcal{S}$
- The initial facts encode the forward constraints

 $_{C} \subseteq \mathcal{V}($  $)$

- The constraints are the original constraints $\mathcal{C}$.

But we also need to encode the backward constraints
$\mathcal{V}($  $) \subseteq$  $_{\mathrm{C}}$!

Reduction to Open World Query Answering

Encoding *ViewProblem*($_{\mathrm{c}}, \mathcal{C}, \mathcal{V}, \mathcal{S}$) **as OWQA**

- The query is $\mathcal{S}$
- The initial facts encode the forward constraints

$_{c} \subseteq \mathcal{V}($$)$

- The constraints are the original constraints $\mathcal{C}$.

But we also need to encode the backward constraints
$\mathcal{V}($$) \subseteq$ $_{\mathrm{c}}$!

For this we use that adom$(\mathcal{V}($$)) = \{\mathcal{C}\}$

Lower bounds

Various reductions from:

- OWQA

Lower bounds

Various reductions from:

- OWQA
- Query evaluation

Lower bounds

Various reductions from:

- OWQA
- Query evaluation
- Alternating Turing Machines

Complexity results

Constraints \ Views	ProjMap	AtomMap	GuardedMap	CQMap
IncDep	PSpace	ExpTime	2ExpTime	2ExpTime
LTGD	ExpTime	ExpTime	2ExpTime	2ExpTime
GTGD	2ExpTime	2ExpTime	2ExpTime	2ExpTime
FGTGD	2ExpTime	2ExpTime	2ExpTime	2ExpTime

Table 1: Complexity of disclosure

$\Rightarrow$ all bounds are *tight*!

Complexity results

Constraints \ Views	ProjMap	AtomMap	GuardedMap	CQMap
IncDep	NP	NP	ExpTime	2ExpTime
LTGD	NP	NP	ExpTime	2ExpTime
GTGD	ExpTime	ExpTime	ExpTime	2ExpTime
FGTGD	2ExpTime	2ExpTime	2ExpTime	2ExpTime

Table 2: Complexity of disclosure in bounded arity

⇒ all bounds are *tight*!

Complexity results

In PTIME:

- CQ secret, foreign keys constraints and projection views

Complexity results

In PTime:

- CQ secret, foreign keys constraints and projection views
- bounded CQ secret, `ProjMap`, `LTGD`

Future Works

- Implement model checker for publication methods.

Future Works

- Implement model checker for publication methods.
- How to synthesize publications automatically?

Thank you!

Questions?