Exchanging More than Complete Data

Marcelo Arenas Jorge Pérez Juan Reutter

PUC Chile, Universidad de Chile, University of Edinburgh

Father	
Andy	Bob
Bob	Danny
Danny	Eddie

Mother	
Carrie	Bob

$$\begin{aligned} \text{Father}(x, y) &\rightarrow \text{Parent}(x, y) \\ \text{Mother}(x, y) &\rightarrow \text{Parent}(x, y) \\ \text{Parent}(x, y) \wedge \text{Parent}(y, z) &\rightarrow \text{Gr-Parent}(x, z) \end{aligned}$$

Father	
Andy	Bob
Bob	Danny
Danny	Eddie

Mother	
Carrie	Bob

$$\begin{aligned} \text{Father}(x, y) &\rightarrow \text{Parent}(x, y) \\ \text{Mother}(x, y) &\rightarrow \text{Parent}(x, y) \\ \text{Parent}(x, y) \wedge \text{Parent}(y, z) &\rightarrow \text{Gr-Parent}(x, z) \end{aligned}$$

Source

$$\begin{aligned} \text{Father}(x, y) &\rightarrow \text{Pateras}(x, y) \\ \text{Gr-Parent}(x, y) &\rightarrow \text{Pappoi}(x, y) \end{aligned}$$

Target

Father	
Andy	Bob
Bob	Danny
Danny	Eddie

Mother	
Carrie	Bob

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Mother}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Parent}(x,y) \wedge \text{Parent}(y,z) &\rightarrow \text{Gr-Parent}(x,z) \end{aligned}$$

Source

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Pateras}(x,y) \\ \text{Gr-Parent}(x,y) &\rightarrow \text{Pappoi}(x,y) \end{aligned}$$

Target

Pateras	
Andy	Bob
Bob	Danny
Danny	Eddie

Pappoi	
Andy	Danny
Carrie	Danny
Bob	Eddie

Father	
Andy	Bob
Bob	Danny
Danny	Eddie

Mother	
Carrie	Bob

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Mother}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Parent}(x,y) \wedge \text{Parent}(y,z) &\rightarrow \text{Gr-Parent}(x,z) \end{aligned}$$

Source

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Pateras}(x,y) \\ \text{Gr-Parent}(x,y) &\rightarrow \text{Pappoi}(x,y) \end{aligned}$$

Target

$$\text{Pateras}(x,y) \wedge \text{Pateras}(y,z) \rightarrow \text{Pappoi}(x,z)$$

Father	
Andy	Bob
Bob	Danny
Danny	Eddie

Mother	
Carrie	Bob

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Mother}(x,y) &\rightarrow \text{Parent}(x,y) \\ \text{Parent}(x,y) \wedge \text{Parent}(y,z) &\rightarrow \text{Gr-Parent}(x,z) \end{aligned}$$

Source

$$\begin{aligned} \text{Father}(x,y) &\rightarrow \text{Pateras}(x,y) \\ \text{Gr-Parent}(x,y) &\rightarrow \text{Pappoi}(x,y) \end{aligned}$$

Target

Pateras	
Andy	Bob
Bob	Danny
Danny	Eddie

Pappoi	
Carrie	Danny

$$\text{Pateras}(x,y) \wedge \text{Pateras}(y,z) \rightarrow \text{Pappoi}(x,z)$$

We can exchange more than complete data

- In data exchange we begin with a database instance
  we begin with complete data
- What if we have a representation of a set of possible instances?
- We propose a new general formalism to exchange representations of possible instances
  and apply it to incomplete instances and knowledge bases

Outline

Formalism for exchanging representations systems

Applications to incomplete instances

Applications to knowledge bases

- What is the complexity of testing kb-solutions?
- How can we compute a good kb-solution?

Concluding remarks

Outline

Representation systems

A representation system is composed of:

- a set **W** of *representatives*
- a function rep from **W** to sets of instances

$$\mathcal{V} \quad \xrightarrow{\text{rep}} \quad \{I_1, I_2, I_3, \ldots\} = \text{rep}(\mathcal{V})$$

Representation systems

A representation system is composed of:

- a set $\mathbf{W}$ of *representatives*
- a function rep from $\mathbf{W}$ to sets of instances

$$\mathcal{V} \quad \xrightarrow{\text{rep}} \quad \{I_1, I_2, I_3, \ldots\} = \text{rep}(\mathcal{V})$$

- Incomplete instances and knowledge bases are representation systems

In classical data exchange
we consider only *complete* data

$\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ a schema mapping from $\mathbf{S}$ to $\mathbf{T}$
$I \in \mathsf{Inst}(\mathbf{S})$, $J \in \mathsf{Inst}(\mathbf{T})$

J is a *solution* for I under $\mathcal{M}$ iff $(I, J) \models \Sigma$

$$J \in \mathsf{Sol}_{\mathcal{M}}(I)$$

In classical data exchange
we consider only *complete* data

$\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ a schema mapping from $\mathbf{S}$ to $\mathbf{T}$
$I \in \mathsf{Inst}(\mathbf{S})$, $J \in \mathsf{Inst}(\mathbf{T})$

J is a *solution* for I under $\mathcal{M}$ iff $(I, J) \models \Sigma$

$$J \in \mathsf{Sol}_{\mathcal{M}}(I)$$

We can extend this to set of instances:
given a set $\mathcal{X} \subseteq \mathsf{Inst}(\mathbf{S})$

$$\mathsf{Sol}_{\mathcal{M}}(\mathcal{X}) = \bigcup_{I \in \mathcal{X}} \mathsf{Sol}_{\mathcal{M}}(I)$$

Extending the definition to representation systems

Given:

- a mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$
- a representation system $\mathcal{R} = (\mathbf{W}, \text{rep})$
- $\mathcal{U}, \mathcal{V} \in \mathbf{W}$

Extending the definition to representation systems

Given:

- a mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$
- a representation system $\mathcal{R} = (\mathbf{W}, \text{rep})$
- $\mathcal{U}, \mathcal{V} \in \mathbf{W}$

Definition

$\mathcal{U}$ is an *$\mathcal{R}$-solution* of $\mathcal{V}$ if

$$\text{rep}(\mathcal{U}) \subseteq \text{Sol}_{\mathcal{M}}(\text{rep}(\mathcal{V}))$$

Extending the definition to representation systems

Given:

- a mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$
- a representation system $\mathcal{R} = (\mathbf{W}, \text{rep})$
- $\mathcal{U}, \mathcal{V} \in \mathbf{W}$

Definition

$\mathcal{U}$ is an *$\mathcal{R}$-solution* of $\mathcal{V}$ if

$$\text{rep}(\mathcal{U}) \subseteq \text{Sol}_{\mathcal{M}}(\text{rep}(\mathcal{V}))$$

or equivalently,

$$\forall J \in \text{rep}(\mathcal{U}),\ \exists I \in \text{rep}(\mathcal{V}) \text{ such that } J \in \text{Sol}_{\mathcal{M}}(I)$$

Universal solutions and strong representation systems

Definition

$\mathcal{U}$ is a *universal $\mathcal{R}$-solution* of $\mathcal{V}$ if

$$\mathsf{rep}(\mathcal{U}) \;=\; \mathsf{Sol}_{\mathcal{M}}(\mathsf{rep}(\mathcal{V}))$$

Universal solutions and strong representation systems

Definition

$\mathcal{U}$ is a *universal $\mathcal{R}$-solution* of $\mathcal{V}$ if

$$\mathsf{rep}(\mathcal{U}) \;=\; \mathsf{Sol}_{\mathcal{M}}(\mathsf{rep}(\mathcal{V}))$$

Let $\mathcal{C}$ be a class of mappings,

Definition

$\mathcal{R} = (\mathbf{W}, \mathsf{rep})$ is a *strong representation system* for $\mathcal{C}$ if
for every $\mathcal{M} \in \mathcal{C}$, and $\mathcal{V} \in \mathbf{W}$, there exists a $\mathcal{U} \in \mathbf{W}$ such that:

$$\mathsf{rep}(\mathcal{U}) \;=\; \mathsf{Sol}_{\mathcal{M}}(\mathsf{rep}(\mathcal{V}))$$

Universal solutions and strong representation systems

Definition

$\mathcal{U}$ is a *universal $\mathcal{R}$-solution* of $\mathcal{V}$ if

$$\mathsf{rep}(\mathcal{U}) \;=\; \mathsf{Sol}_{\mathcal{M}}(\mathsf{rep}(\mathcal{V}))$$

Let $\mathcal{C}$ be a class of mappings,

Definition

$\mathcal{R} = (\mathbf{W}, \mathsf{rep})$ is a *strong representation system* for $\mathcal{C}$ if
for every $\mathcal{M} \in \mathcal{C}$, and $\mathcal{V} \in \mathbf{W}$, there exists a $\mathcal{U} \in \mathbf{W}$ such that:

$$\mathsf{rep}(\mathcal{U}) \;=\; \mathsf{Sol}_{\mathcal{M}}(\mathsf{rep}(\mathcal{V}))$$

Strong representation systems $\Rightarrow$ universal solutions for representatives in $\mathcal{R}$ can always be represented in the same system.

Outline

Incomplete Instances

We consider:

- Naive instances: instances with null values

Incomplete Instances

We consider:

- Naive instances: instances with null values
- Positive conditional instances: naive instances plus *conditions* for tuples
  - equalities between nulls, and between nulls and constants
  - conjunctions and disjunctions

$$\begin{array}{r|l} T(1, N_1) & \mathit{true} \\ R(1, N_1, N_2) & N_1 = N_2 \vee (N_1 = 2 \wedge N_2 = 3) \end{array}$$

Incomplete Instances

We consider:

- Naive instances: instances with null values
- Positive conditional instances: naive instances plus *conditions* for tuples
  - equalities between nulls, and between nulls and constants
  - conjunctions and disjunctions

$$\begin{array}{r|l} T(1, N_1) & \mathit{true} \\ R(1, N_1, N_2) & N_1 = N_2 \vee (N_1 = 2 \wedge N_2 = 3) \end{array}$$

Semantics given by *valuations* of nulls (+ open world):

$$\mathrm{rep}(\mathcal{I}) = \{K \mid \mu(\mathcal{I}) \subseteq K \text{ for some valuation } \mu\}$$

Strong representation systems for st-tgds

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be specified by source-to-target tgds (st-tgd: $\phi_{\mathbf{S}}(\bar{x}) \rightarrow \exists \bar{y} \psi_{\mathbf{T}}(\bar{x}, \bar{y})$, $\phi_{\mathbf{S}}$ and $\psi_{\mathbf{T}}$ are CQs)

(FKMP03)

For every complete instance I there exists a naive instance $\mathcal{J}$ s.t.

$$\mathsf{rep}(\mathcal{J}) \;=\; \mathsf{Sol}_{\mathcal{M}}(I)$$

Strong representation systems for st-tgds

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be specified by source-to-target tgds (st-tgd: $\phi_{\mathbf{S}}(\bar{x}) \rightarrow \exists \bar{y} \psi_{\mathbf{T}}(\bar{x}, \bar{y})$, $\phi_{\mathbf{S}}$ and $\psi_{\mathbf{T}}$ are CQs)

(FKMP03)

For every complete instance I there exists a naive instance $\mathcal{J}$ s.t.

$$\mathrm{rep}(\mathcal{J}) \;=\; \mathrm{Sol}_{\mathcal{M}}(I)$$

Proposition

Naive instances are not a strong representation system for st-tgds

Idea

$\mathtt{Mng}(N, \text{Alice})$

$$\begin{aligned} \mathtt{Mng}(x, y) &\rightarrow \mathtt{Reports}(x, y) \\ \mathtt{Mng}(x, x) &\rightarrow \mathtt{Self\text{-}Mng}(x) \end{aligned}$$

Strong representation systems for st-tgds

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be specified by source-to-target tgds (st-tgd: $\phi_{\mathbf{S}}(\bar{x}) \rightarrow \exists \bar{y} \psi_{\mathbf{T}}(\bar{x}, \bar{y})$, $\phi_{\mathbf{S}}$ and $\psi_{\mathbf{T}}$ are CQs)

(FKMP03)

For every complete instance I there exists a naive instance $\mathcal{J}$ s.t.

$$\mathrm{rep}(\mathcal{J}) \;=\; \mathrm{Sol}_{\mathcal{M}}(I)$$

Proposition

Naive instances are not a strong representation system for st-tgds

Idea

$\mathtt{Mng}(N, \mathrm{Alice})$

$$\begin{array}{lcll} \mathtt{Mng}(x,y) & \rightarrow & \mathtt{Reports}(x,y) & ? \\ \mathtt{Mng}(x,x) & \rightarrow & \mathtt{Self\text{-}Mng}(x) & \end{array}$$

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

$\texttt{Mng}(N, \text{Alice})$

$$\begin{aligned} \texttt{Mng}(x, y) &\rightarrow \texttt{Reports}(x, y) \\ \texttt{Mng}(x, x) &\rightarrow \texttt{Self-Mng}(x) \end{aligned}$$

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

$\texttt{Mng}(N, \text{Alice})$	$\texttt{Mng}(x, y)$	$\longrightarrow$	$\texttt{Reports}(x, y)$	$\texttt{Reports}(N, \text{Alice})$	*true*
	$\texttt{Mng}(x, x)$	$\longrightarrow$	$\texttt{Self-Mng}(x)$		

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

$\texttt{Mng}(N, \text{Alice})$	$\texttt{Mng}(x, y)$	$\rightarrow$	$\texttt{Reports}(x, y)$	$\texttt{Reports}(N, \text{Alice})$	*true*
	$\texttt{Mng}(x, x)$	$\rightarrow$	$\texttt{Self-Mng}(x)$	$\texttt{Self-Mng}(\text{Alice})$	

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

$$\begin{array}{lcll} \texttt{Mng}(N, \text{Alice}) & \texttt{Mng}(x, y) & \rightarrow & \texttt{Reports}(x, y) \\ & \texttt{Mng}(x, x) & \rightarrow & \texttt{Self-Mng}(x) \end{array} \qquad \begin{array}{r|l} \texttt{Reports}(N, \text{Alice}) & \textit{true} \\ \texttt{Self-Mng}(\text{Alice}) & N = \text{Alice} \end{array}$$

Positive conditional instances are enough

Theorem

Positive conditional instances are a strong representation system for st-tgds.

$$\begin{array}{lcll} \texttt{Mng}(N, \text{Alice}) & \texttt{Mng}(x,y) & \rightarrow & \texttt{Reports}(x,y) \\ & \texttt{Mng}(x,x) & \rightarrow & \texttt{Self-Mng}(x) \end{array} \qquad \begin{array}{r|l} \texttt{Reports}(N, \text{Alice}) & \textit{true} \\ \texttt{Self-Mng}(\text{Alice}) & N = \text{Alice} \end{array}$$

Corollary

Positive conditional instances are a strong representation system for SO-tgds.

Positive conditional instances are *exactly* the needed representation system

Theorem

Positive conditional instances are minimal*:*

- *equalities between nulls*
- *equalities between constant and nulls*
- *conjunctions and disjunctions*

are all needed for a strong representation system of st-tgds

Positive conditional instances match the complexity bounds of classical data exchange

Theorem

For positive conditional instances and (fixed) st-tgds

- *Universal solutions can be constructed in* polynomial time.
- *Certain answers of unions of conjunctive queries can be computed in* polynomial time.
- *Certain answers of unions of conjunctive queries with one inequality per disjunct can be computed in* polynomial time.

Outline

The semantics of *knowledge bases* is given by sets of instances

Knowledge base over $\mathbf{S}$: (I, Γ) s.t.

$I \in \mathsf{Inst}(\mathbf{S})$

Γ a set of rules over $\mathbf{S}$

Semantics: finite models

$$\mathsf{Mod}(I, \Gamma) = \{K \in \mathsf{Inst}(\mathbf{S}) \mid I \subseteq K \text{ and } K \models \Gamma\}$$

We can apply our formalism to knowledge bases

(I_2, Γ_2) is a *kb-solution* for (I_1, Γ_1) under $\mathcal{M}$ iff

$$\mathsf{Mod}(I_2, \Gamma_2) \subseteq \mathsf{Sol}_{\mathcal{M}}(\mathsf{Mod}(I_1, \Gamma_1))$$

Outline

We study the following decision problem

CHECK-KB-SOL:

Input: $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with Σ st-tgds
(I_1, Γ_1) knowledge base over $\mathbf{S}$ with Γ_1 tgds
(I_2, Γ_2) knowledge base over $\mathbf{T}$ with Γ_2 tgds

Output: Is (I_2, Γ_2) a kb-solution of (I_1, Γ_1) under $\mathcal{M}$?

We study the following decision problem

CHECK-KB-SOL:

Input: $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with Σ st-tgds
(I_1, Γ_1) knowledge base over $\mathbf{S}$ with Γ_1 tgds
(I_2, Γ_2) knowledge base over $\mathbf{T}$ with Γ_2 tgds

Output: Is (I_2, Γ_2) a kb-solution of (I_1, Γ_1) under $\mathcal{M}$?

Theorem

CHECK-KB-SOL *is undecidable (even for fixed* $\mathcal{M}$*).*

We study the following decision problem

CHECK-KB-SOL:

Input: $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with Σ st-tgds
(I_1, Γ_1) knowledge base over $\mathbf{S}$ with Γ_1 tgds
(I_2, Γ_2) knowledge base over $\mathbf{T}$ with Γ_2 tgds

Output: Is (I_2, Γ_2) a kb-solution of (I_1, Γ_1) under $\mathcal{M}$?

Theorem

CHECK-KB-SOL *is undecidable (even for fixed* $\mathcal{M}$*).*

Undecidability is a consequence of using $\exists$ in knowledge bases

CHECK-FULL-KB-SOL: Γ_1, Γ_2 full-tgds (no $\exists$)

The complexity of Check-Full-KB-Sol

Theorem

Check-Full-KB-Sol *is EXPTIME-complete.*

The complexity of Check-Full-KB-Sol

Theorem

Check-Full-KB-Sol *is EXPTIME-complete.*

Theorem

If $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *is fixed*

Check-Full-KB-Sol *is* $\Delta_2^P[O(\log n)]$*-complete.*

$\Delta_2^P[O(\log n)] \equiv P^{NP}$ with only a *logarithmic number of calls* to the *NP* oracle.

Outline

Minimal knowledge bases as good kb-solutions

What are good knowledge-base solutions?

- first choice: universal kb-solutions, but
- there are some other kb-solutions desirable to materialize

Minimal knowledge bases as good kb-solutions

What are good knowledge-base solutions?

- first choice: universal kb-solutions, but
- there are some other kb-solutions desirable to materialize

$\mathcal{X} \equiv_{\mathsf{min}} \mathcal{Y}$: $\mathcal{X}$ and $\mathcal{Y}$ coincide in the *minimal instances*

Minimal knowledge bases as good kb-solutions

What are good knowledge-base solutions?

- first choice: universal kb-solutions, but
- there are some other kb-solutions desirable to materialize

$\mathcal{X} \equiv_{\text{min}} \mathcal{Y}$: $\mathcal{X}$ and $\mathcal{Y}$ coincide in the *minimal instances*

(I_2, Γ_2) is a *minimal kb-solution* of (I_1, Γ_1) under $\mathcal{M}$ if

$$\text{Mod}(I_2, \Gamma_2) \equiv_{\text{min}} \text{Sol}_{\mathcal{M}}(\text{Mod}(I_1, \Gamma_1))$$

We consider minimal kb-solutions as the good solutions

Two requirements to construct minimal knowledge-base solutions

Given (I_1, Γ_1) and Σ, when constructing a minimal-knowledge base solution (I_2, Γ_2) we would want:

1. Γ_2 to only depend on Γ_1 and Σ

Two requirements to construct minimal knowledge-base solutions

Given (I_1, Γ_1) and Σ, when constructing a minimal-knowledge base solution (I_2, Γ_2) we would want:

1. Γ_2 to only depend on Γ_1 and Σ

Γ_2 is *safe* for Γ_1 and Σ

Two requirements to construct minimal knowledge-base solutions

Given (I_1, Γ_1) and Σ, when constructing a minimal-knowledge base solution (I_2, Γ_2) we would want:

1. Γ_2 to only depend on Γ_1 and Σ

Γ_2 is *safe* for Γ_1 and Σ

2. Γ_2 to be as informative as possible

thus minimizing the size of I_2

Two requirements to construct minimal knowledge-base solutions

Given (I_1, Γ_1) and Σ, when constructing a minimal-knowledge base solution (I_2, Γ_2) we would want:

1. Γ_2 to only depend on Γ_1 and Σ

Γ_2 is *safe* for Γ_1 and Σ

2. Γ_2 to be as informative as possible

thus minimizing the size of I_2

Γ_2 is *optimum-safe* if for every other safe set Γ'

$$\Gamma_2 \models \Gamma'$$

Safe sets: data independent target knowledge bases

Unfortunately, optimum-safe sets are not (always) FO-expressible

Theorem

There exist sets Γ_1 (full & acyclic) and Σ (full) s.t.

no FO sentence is optimum-safe for Γ_1 and Σ

Safe sets: data independent target knowledge bases

Unfortunately, optimum-safe sets are not (always) FO-expressible

Theorem

There exist sets Γ_1 (full & acyclic) and Σ (full) s.t.

no FO sentence is optimum-safe for Γ_1 and Σ

In the paper: an algorithm to compute *good* safe sets

- works for full & acyclic Γ_1 and full Σ
- we use, unfolding, rewriting, and composition of mappings

In the paper: a complete strategy for computing minimal knowledge-base solutions

Given Γ_1 and Σ:

- We compute Γ_2 safe for Γ_1 and Σ
- We then compute minimal set Γ^* (implied by Γ_1) such that for every I_1

In the paper: a complete strategy for computing minimal knowledge-base solutions

Given Γ_1 and Σ:

- We compute Γ_2 safe for Γ_1 and Σ
- We then compute minimal set Γ^* (implied by Γ_1) such that

  for every I_1

$$\text{chase}_{\Gamma^*}(I_1)$$

In the paper: a complete strategy for computing minimal knowledge-base solutions

Given Γ_1 and Σ:

- We compute Γ_2 safe for Γ_1 and Σ
- We then compute minimal set Γ^* (implied by Γ_1) such that

  for every I_1

$$\text{chase}_{\Sigma}(\,\text{chase}_{\Gamma^*}(I_1))$$

In the paper: a complete strategy for computing minimal knowledge-base solutions

Given Γ_1 and Σ:

- We compute Γ_2 safe for Γ_1 and Σ
- We then compute minimal set Γ^* (implied by Γ_1) such that

  for every I_1

$$\left(\text{chase}_{\Sigma}(\text{chase}_{\Gamma^*}(I_1)), \Gamma_2 \right)$$

In the paper: a complete strategy for computing minimal knowledge-base solutions

Given Γ_1 and Σ:

- We compute Γ_2 safe for Γ_1 and Σ
- We then compute minimal set Γ^* (implied by Γ_1) such that

  for every I_1

  $$\Big(\text{chase}_\Sigma(\text{chase}_{\Gamma^*}(I_1)), \Gamma_2 \Big)$$

  is a minimal kb-solution for (I_1, Γ_1).

Outline

We can exchange more than complete data

We propose a general formalism
to exchange *representation systems*

- Applications to incomplete instances
- Applications to knowledge bases

We can exchange more than complete data

We propose a general formalism
to exchange *representation systems*

- Applications to incomplete instances
- Applications to knowledge bases

Next step: apply our general setting to the Semantic Web

- Semantic Web data have *nulls* (blank nodes)
- Semantic Web specifications have rules (RDFS, OWL)

Exchanging More than Complete Data

Marcelo Arenas Jorge Pérez Juan Reutter

PUC Chile, Universidad de Chile, University of Edinburgh

Outline

Formalism for exchanging representations systems

Applications to incomplete instances

Applications to knowledge bases

- What is the complexity of testing kb-solutions?
- How can we compute a good kb-solution?

Concluding remarks