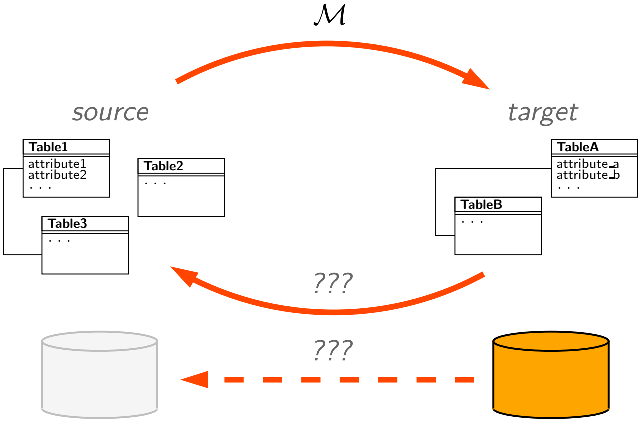
The Inverse

Jorge Pérez

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

DEIS'10, Schloss Dagstuhl

How do we *recover* exchanged data?
What is a good *inverse* mapping?



Inverting Schema Mappings

Research questions:

- What is a *good semantics* for inverting schema mappings?
- How can we *test invertibility* of schema mappings?
- Can we *compute* an inverse?
- What is the *language needed* to express an inverse?

Outline

- Fagin-inverse (PODS'06)
- Quasi-inverse (PODS'07)
- Maximum Recovery (PODS'08)
- Computing Inverses
- Query language-based inverses (VLDB'09)
- Dealing with nulls in source instances (PODS'09)

Preliminaries

A mapping $\mathcal{M}$ from **S** to **T** is a set of pairs (I, J) s.t.:

- I is an instance of **S** (source schema), and
- J is an instance of **T** (target schema)

Recall that $\text{Sol}_{\mathcal{M}}(I) = \{J \mid (I, J) \in \mathcal{M}\}$.

Mappings usually defined in terms of a set Σ of formulas:

- $\mathcal{M} = \{(I, J) \mid (I, J) \models \Sigma\}$

We assume that:

- source instances contain only *constant values*
- target instances may contain *null values*.

(we drop this assumption at the end of this talk)

How to define the inverse of a mapping?

Ron Fagin (PODS'06)

"A mapping composed with its inverse should equal the *identity*"

We know how to compose, but what is a natural identity?

- Let $\mathbf{S} = \{R, S, \ldots\}$, and $\hat{\mathbf{S}} = \{\hat{R}, \hat{S}, \ldots\}$ a *copy* of $\mathbf{S}$.
- Let $\overline{\mathsf{Id}}$ be the mapping from $\mathbf{S}$ to $\hat{\mathbf{S}}$ specified by

$$\Sigma_{\overline{\mathsf{Id}}} = \{\ R(\bar{x}) \rightarrow \hat{R}(\bar{x})\ \mid R \in \mathbf{S}\} \quad (\textit{copying setting})$$

- $\overline{\mathsf{Id}}$ is a very natural identity when one focuses on st-tgds.
  $\overline{\mathsf{Id}}$ is not exactly the identity for binary relations:

$$\overline{\mathsf{Id}} = \{(I, \hat{K}) \in \mathbf{S} \times \hat{\mathbf{S}} \mid I \subseteq K\}.$$

Fagin-inverse (Fagin, PODS'06)

Definition (F06)

Let $\mathcal{M}$ be a mapping from $\mathbf{S}$ to $\mathbf{T}$, and $\mathcal{M}'$ from $\mathbf{T}$ to $\hat{\mathbf{S}}$.
$\mathcal{M}'$ is a *Fagin-inverse* of $\mathcal{M}$ if

$$\mathcal{M} \circ \mathcal{M}' \quad = \quad \overline{\mathsf{Id}}$$

Example

$$\begin{array}{rccc} \mathcal{M}: & R(x,y) & \rightarrow & T(x,y) \\ \mathcal{M}': & T(x,y) & \rightarrow & \hat{R}(x,y) \\ & & & \\ \mathcal{M} \circ \mathcal{M}': & R(x,y) & \rightarrow & \hat{R}(x,y) \end{array}$$

$\mathcal{M}'$ is a Fagin-inverse of $\mathcal{M}$.

Fagin-inverse: Examples

Example

$$\begin{array}{rrcl}
\mathcal{M}: & R(x,y) & \rightarrow & T(x,x,y) \\
\\
\mathcal{M}_1: & T(x,x,y) & \rightarrow & \hat{R}(x,y) \\
\mathcal{M}_2: & T(x,u,y) & \rightarrow & \hat{R}(x,y) \\
\mathcal{M}_3: & T(u,x,y) & \rightarrow & \hat{R}(x,y) \\
\\
\mathcal{M} \circ \mathcal{M}_1: & R(x,y) & \rightarrow & \hat{R}(x,y) \\
\mathcal{M} \circ \mathcal{M}_2: & R(x,y) & \rightarrow & \hat{R}(x,y) \\
\mathcal{M} \circ \mathcal{M}_3: & R(x,y) & \rightarrow & \hat{R}(x,y)
\end{array}$$

They are all inverses of $\mathcal{M}$.

Fagin-inverse: More examples

Example

$$
\begin{array}{lrcl}
\mathcal{M}: & R(x) & \rightarrow & T(x) \\
 & R(x) & \rightarrow & S(x) \\
 & P(x) & \rightarrow & T(x) \\
 & P(x) & \rightarrow & U(x)
\end{array}
$$

$$
\begin{array}{lrcl}
\mathcal{M}': & S(x) & \rightarrow & \hat{R}(x) \\
 & U(x) & \rightarrow & \hat{P}(x)
\end{array}
$$

$\mathcal{M}'$ is a Fagin-inverse of $\mathcal{M}$.

Fagin-inverse: More examples

Example

$$\mathcal{M}: \quad \begin{array}{rcl} R(x) & \rightarrow & T(x) \\ R(x) & \rightarrow & S(x) \\ P(x) & \rightarrow & T(x) \\ P(x) & \rightarrow & U(x) \end{array}$$

$$\mathcal{M}': \quad \begin{array}{rcl} T(x) & \rightarrow & \hat{R}(x) \\ U(x) & \rightarrow & \hat{P}(x) \end{array}$$

$$\mathcal{M} \circ \mathcal{M}': \quad \begin{array}{rcl} R(x) & \rightarrow & \hat{R}(x) \\ P(x) & \rightarrow & \hat{R}(x) \\ P(x) & \rightarrow & \hat{P}(x) \end{array}$$

$\mathcal{M}'$ is not a Fagin-inverse of $\mathcal{M}$.

Fagin-inverse: More examples

Example

$$\mathcal{M}: \quad \begin{array}{rcl} R(x,y) & \rightarrow & T(x,y) \\ P(x) & \rightarrow & T(x,x) \wedge S(x) \\ R(x,x) & \rightarrow & U(x) \end{array}$$

$$\mathcal{M}': \quad \begin{array}{rcl} T(x,y) \wedge x \neq y & \rightarrow & \hat{R}(x,y) \\ U(x) & \rightarrow & \hat{R}(x,x) \\ S(x) & \rightarrow & \hat{P}(x) \end{array}$$

$\mathcal{M}'$ is a Fagin-inverse of $\mathcal{M}$.

Several st-tgds mappings do not have Fagin-inverses.

Example

$$
\begin{array}{lrcl}
\mathcal{M}_1: & R(x, y) & \rightarrow & S(x) \\
\mathcal{M}_2: & R(x, y) & \rightarrow & S(x) \wedge T(y) \\
\mathcal{M}_3: & R(x) & \rightarrow & S(x) \\
 & P(x) & \rightarrow & S(x)
\end{array}
$$

Do they have Fagin-inverse? intuitively, they do not.
How do we formally prove that a mapping is (not) Fagin-invertible?

The unique-solutions property

Definition (F06)

$\mathcal{M}$ has the *unique-solutions property* if for every I_1 and I_2

$$\mathsf{Sol}_{\mathcal{M}}(I_1) = \mathsf{Sol}_{\mathcal{M}}(I_2) \text{ implies } I_1 = I_2.$$

Theorem (F06)

Let $\mathcal{M}$ be specified by st-tgds. If $\mathcal{M}$ is Fagin-invertible then $\mathcal{M}$ has the unique-solutions property.

We have a very simple necessary condition!

Using the unique-solutions property

Example

$$\mathcal{M}_1: \quad R(x,y) \rightarrow S(x)$$

$$\mathcal{M}_2: \quad R(x,y) \rightarrow S(x) \wedge T(y)$$

$$\mathcal{M}_3: \quad \begin{aligned} R(x) &\rightarrow S(x) \\ P(x) &\rightarrow S(x) \end{aligned}$$

have no Fagin-inverse.

They do not satisfy the unique-solutions property.

- $\mathcal{M}_1$: $I_1 = \{R(1,2)\}$, $I_2 = \{R(1,3)\}$.
- $\mathcal{M}_2$: $I_1 = \{R(1,2), R(3,4)\}$, $I_2 = \{R(1,4), R(3,2)\}$.
- $\mathcal{M}_3$: $I_1 = \{R(1)\}$, $I_2 = \{P(1)\}$.

Unfortunately, the unique-solutions property is not sufficient.

How can we check Fagin-invertibility?

Definition (Fagin et al., PODS'07)

$\mathcal{M}$ has the *subset property* if for every I_1 and I_2

$$\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_2) \text{ implies } I_2 \subseteq I_1.$$

Theorem (FKPT07)

Let $\mathcal{M}$ be specified by st-tgds. $\mathcal{M}$ is Fagin-invertible if and only if *$\mathcal{M}$ has the subset property.*

What can we do if a Fagin-inverse does not exist?

Example

$$\begin{array}{lrcl} \mathcal{M}_1: & R(x,y) & \rightarrow & S(x) \\ \mathcal{M}_2: & R(x,y) & \rightarrow & S(x) \wedge T(y) \\ \mathcal{M}_3: & R(x) \wedge P(y) & \rightarrow & U(x,y) \end{array}$$

They are not Fagin-invertible, but
we still can find *good reverse mappings*

Example

$$\begin{array}{lrcl} \mathcal{M}_2': & S(x) & \rightarrow & \exists u\ R(x,u) \\ & T(y) & \rightarrow & \exists v\ R(v,y) \end{array}$$

Two main proposals for relaxed notions of inverse of mappings:

- Fagin et al., PODS'07: *Quasi-inverse*
- Arenas et al., PODS'08: *Maximum-recovery*

Outline

Fagin-inverse (PODS'06)

Quasi-inverse (PODS'07)

Maximum Recovery (PODS'08)

Computing Inverses

Query language-based inverses (VLDB'09)

Dealing with nulls in source instances (PODS'09)

Quasi-inverses of schema mappings

Fagin et al. (FKPT07)

"When inverting mappings, do not differentiate instances that has the same space of solutions"

Given a mapping $\mathcal{M}$ define the equivalence relation:

$$I_1 \sim_{\mathcal{M}} I_2 \iff \mathrm{Sol}_{\mathcal{M}}(I_1) = \mathrm{Sol}_{\mathcal{M}}(I_2)$$

Informaly:

$\mathcal{M}'$ is a *quasi-inverse* of $\mathcal{M}$ if the equation

$$\mathcal{M} \circ \mathcal{M}' = \overline{\mathrm{Id}}$$

holds *modulo* the equivalence relation $\sim_{\mathcal{M}}$.

Quasi-inverses of schema mappings

Definition

Let D be a binary relation on instances of a schema $\mathbf{S}$, and $\mathcal{M}$ a mapping with source schema $\mathbf{S}$. Define $D[\sim_{\mathcal{M}}]$ as

$$D[\sim_{\mathcal{M}}] \quad = \quad \{(I, J) \mid \text{exists } K \text{ and } L \text{ such that } I \sim_{\mathcal{M}} K,\ J \sim_{\mathcal{M}} L, \text{ and } (K, L) \in D\ \}$$

From now on, we do not differentiate between $\mathbf{S}$ and $\hat{\mathbf{S}}$, thus we redefine $\overline{\mathsf{Id}}$ as

$$\overline{\mathsf{Id}} = \{(I, J) \mid I \text{ and } J \text{ are instances of } \mathbf{S} \text{ and } I \subseteq J\}$$

Definition (FKPT07)

$\mathcal{M}'$ is a *quasi-inverse* of $\mathcal{M}$ if

$$(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}] = \overline{\mathsf{Id}}[\sim_{\mathcal{M}}]$$

Non Fagin-invertible mappings can have quasi-inverses

Example

$$\mathcal{M}: \quad R(x,y) \rightarrow S(x)$$

$$\mathcal{M}': \quad S(x) \rightarrow \exists u\, R(x,u)$$

$\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$.

Consider $I_1 = \{R(1,2)\}$ and $I_2 = \{R(1,3)\}$

- $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$,
- $(I_1, I_2) \notin \overline{\text{Id}}$, thus $\mathcal{M}'$ is not a Fagin-inverse of $\mathcal{M}$,
- $(I_1, I_2) \in \overline{\text{Id}}[\sim_{\mathcal{M}}]$, since $I_1 \sim_{\mathcal{M}} I_2$ and $(I_1, I_1) \in \overline{\text{Id}}$.

Non Fagin-invertible mappings can have quasi-inverses

Example

$$\begin{array}{llcl} \mathcal{M}: & R(x) & \rightarrow & S(x) \\ & P(x) & \rightarrow & S(x) \\ \\ \mathcal{M}_1: & S(x) & \rightarrow & R(x) \vee P(x) \end{array}$$

$\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$.

Consider $I_1 = \{R(1)\}$ and $I_2 = \{P(1)\}$

- $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$,
- $(I_1, I_2) \in \overline{\mathrm{Id}}[\sim_{\mathcal{M}}]$, since $I_1 \sim_{\mathcal{M}} I_2$ and $(I_1, I_1) \in \overline{\mathrm{Id}}$.

Necessary and sufficient condition for quasi-inverses

(FKPT07) define the $\sim_{\mathcal{M}}$*-subset property*, as a relaxation of the subset property.

Theorem (FKPT07)

Let $\mathcal{M}$ be specified by st-tgds. $\mathcal{M}$ is quasi-invertible if and only if *$\mathcal{M}$ has the $\sim_{\mathcal{M}}$-subset property.*

If $\mathcal{M}$ is Fagin-invertible, then $\sim_{\mathcal{M}}$ coincides with $=$, thus:

Theorem (FKPT07)

If $\mathcal{M}$ is Fagin-invertible, then

quasi-inverses and Fagin-inverses coincide.

Not every st-tgd mapping is quasi-invertible

Example

$$\mathcal{M}: \quad E(x,z) \wedge E(z,y) \rightarrow F(x,y) \wedge M(z)$$

Does not satisfy the $\sim_{\mathcal{M}}$-subset property $\Rightarrow$ is not quasi-invertible.

But we have a *natural reverse mapping* in this case:

$$\mathcal{M}': \quad \begin{aligned} F(x,y) &\rightarrow \exists u\ E(x,u) \wedge E(u,y) \\ M(z) &\rightarrow \exists v \exists w\ E(v,z) \wedge E(z,w) \end{aligned}$$

- This was the main motivation of Arenas et al. (APR08) to propose a new notion of inverse.

Outline

- Fagin-inverse (PODS'06)
- Quasi-inverse (PODS'07)
- Maximum Recovery (PODS'08)
- Computing Inverses
- Query language-based inverses (VLDB'09)
- Dealing with nulls in source instances (PODS'09)

Recovery: specifies how to recover *sound* information.

Idea 1: (Arenas et al., PODS'08)

- data *may be lost* in the exchange through $\mathcal{M}$.
- we want an $\mathcal{M}'$ that *at least* recovers *sound* data w.r.t. $\mathcal{M}$.

$\mathcal{M}'$ is called a *recovery* of $\mathcal{M}$.

Example

Emp(*name*, *lives_in*, *works_in*)   Shuttle(*name*, *destination*)

$$\mathcal{M}: \quad \text{Emp}(x, y, z) \wedge y \neq z \longrightarrow \text{Shuttle}(x, z)$$

$$\mathcal{M}_1: \quad \text{Shuttle}(x, z) \longrightarrow \exists U \exists V \; \text{Emp}(x, U, V) \quad \checkmark$$
$$\mathcal{M}_2: \quad \text{Shuttle}(x, z) \longrightarrow \exists U \; \text{Emp}(x, U, z) \quad \checkmark$$
$$\mathcal{M}_3: \quad \text{Shuttle}(x, z) \longrightarrow \exists V \; \text{Emp}(x, z, V) \quad \times$$

Maximum recovery, the *most informative* recovery

Can we compare alternative recoveries?

Example

$$\mathcal{M}: \quad \text{Emp}(x, y, z) \wedge y \neq z \quad \longrightarrow \quad \text{Shuttle}(x, z)$$

$$\begin{array}{lllll} \mathcal{M}_1: & \text{Shuttle}(x, z) & \longrightarrow & \exists U \exists V & \text{Emp}(x, U, V) \\ \mathcal{M}_2: & \text{Shuttle}(x, z) & \longrightarrow & \exists U & \text{Emp}(x, U, z) \\ \mathcal{M}_4: & \text{Shuttle}(x, z) & \longrightarrow & \exists U & \text{Emp}(x, U, z) \wedge U \neq z \end{array}$$

$\mathcal{M}_2$ *is better than* $\mathcal{M}_1$

$\mathcal{M}_4$ *is better than* $\mathcal{M}_2$ and $\mathcal{M}_1$

Idea 2: (APR08)

- Choose a recovery $\mathcal{M}'$ of $\mathcal{M}$ that is *better than every other*.

$\mathcal{M}'$ is a *maximum recovery* of $\mathcal{M}$.

Recovery: formalization

- Let Id be the identity over a schema $\mathbf{S}$, that is

$$\text{Id} = \{(I, I) \mid I \text{ is an instance of } \mathbf{S}\}$$

- Notice the difference between Id and $\overline{\text{Id}}$.

Definition (APR08)

$$\mathcal{M}' \text{ is a } \textit{recovery} \text{ of } \mathcal{M} \quad \text{iff} \quad \text{Id} \subseteq \mathcal{M} \circ \mathcal{M}'$$

Intuitively: $\mathcal{M}'$ is a recovery of $\mathcal{M}$ if for every instance I
I is *a possible solution for itself* under $\mathcal{M} \circ \mathcal{M}'$.

Maximum recovery: formalization

Being a recovery is just a *sound* condition.

Definition (APR08)

$\mathcal{M}'$ is a *maximum recovery* of $\mathcal{M}$ iff

- $\mathcal{M}'$ is a recovery of $\mathcal{M}$, and
- *for every possible recovery* $\mathcal{M}''$ of $\mathcal{M}$ we have

$$\mathsf{Id} \ \subseteq \ \mathcal{M} \circ \mathcal{M}' \ \subseteq \ \mathcal{M} \circ \mathcal{M}''$$

Intuitively:

We want $\mathcal{M} \circ \mathcal{M}'$ to be
as close as possible to the identity mapping.

Characterizing maximum recoveries

How can we check that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$?
The definition implies a quantification over *all possible* recoveries!

Theorem (APR08)

$\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ iff

$$\mathcal{M} \circ \mathcal{M}' \circ \mathcal{M} \;\; = \;\; \mathcal{M}$$

Example

$$\mathcal{M}: \qquad E(x,z) \wedge E(z,y) \quad \rightarrow \quad F(x,y) \wedge M(z)$$

$$\begin{aligned} \mathcal{M}': \qquad F(x,y) \quad &\rightarrow \quad \exists u \; E(x,u) \wedge E(u,y) \\ M(z) \quad &\rightarrow \quad \exists v \exists w \; E(v,z) \wedge E(z,w) \end{aligned}$$

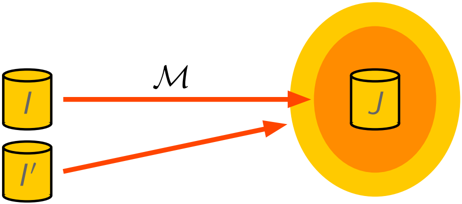
it can be checked that $\mathcal{M} \circ \mathcal{M}' \circ \mathcal{M} = \mathcal{M}$,
thus $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

How can we check if a mapping has a maximum recovery?

Definition

J is a *witness solution* for I under $\mathcal{M}$ if for every other instance I',

$$J \in \mathsf{Sol}_{\mathcal{M}}(I') \implies \mathsf{Sol}_{\mathcal{M}}(I) \subseteq \mathsf{Sol}_{\mathcal{M}}(I').$$



Theorem

$\mathcal{M}$ has a maximum recovery iff every source instance has a witness solution.

Every st-tgd mapping has a maximum recovery

Theorem (APR08)

Every mapping specified by st-tgds has a maximum recovery.

Proof idea

For st-tgds, every *universal solution* is a witness solution.

Relationship with previous notions

Theorem (APR08)

If $\mathcal{M}$ is specified by st-tgds and is Fagin-invertible then

$$\mathcal{M}' \text{ is a Fagin-inverse of } \mathcal{M} \text{ iff}$$
$$\mathcal{M}' \text{ is a maximum recovery of } \mathcal{M}.$$

For quasi-inverses:

- there are quasi-inverses that are not recoveries.

Outline

Fagin-inverse (PODS'06)

Quasi-inverse (PODS'07)

Maximum Recovery (PODS'08)

Computing Inverses

Query language-based inverses (VLDB'09)

Dealing with nulls in source instances (PODS'09)

How do we compute an inverse? we need some tools first

Source rewriting

Consider a mapping $\mathcal{M}$ from $\mathbf{S}$ to $\mathbf{T}$, and a target query $Q_\mathbf{T}$.

- $Q_\mathbf{S}$ is a *source rewriting* of $Q_\mathbf{T}$ if

$$\underline{\text{certain}}_{\mathcal{M}}(Q_\mathbf{T}, I) \;=\; Q_\mathbf{S}(I)$$

Well-known fact:

For mappings specified by st-tgds and target queries in **CQ**, a source rewriting always exists and can be expressed in $\mathbf{UCQ}^{=}$.

$$\mathcal{M}: \quad \begin{array}{rcl} P(x) & \rightarrow & T(x,x) \\ R(x,y) & \rightarrow & T(x,y) \end{array}$$

$$Q_\mathbf{T}(x,y):\; T(x,y)$$
$$Q_\mathbf{S}(x,y):\; (P(x) \wedge x = y) \vee R(x,y)$$

An algorithm for computing inverses

Algorithm

Let $\mathcal{M}$ be a mapping from **S** to **T** specified by a set Σ of st-tgds:

- Let $\Sigma' = \emptyset$.
- For every dependency $\varphi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \psi(\bar{x}, \bar{z})$ in Σ:
  - Compute a source rewriting $\alpha(\bar{x})$ of $\exists \bar{z}\ \psi(\bar{x}, \bar{z})$.
  - Add to Σ' the dependency

$$\psi(\bar{x}, \bar{z}) \wedge \mathbf{Const}(\bar{x}) \rightarrow \alpha(\bar{x}).$$

- Return the mapping $\mathcal{M}'$ from **T** to **S** specified by Σ'.

Theorem (APR08)

The algorithm produces a maximum recovery of $\mathcal{M}$.
It produces Fagin(quasi)-inverses if $\mathcal{M}$ is Fagin(quasi)-invertible.

What is the language needed to specify inverses?

The output of the algorithm uses:

- $\mathbf{UCQ}^{=}$ in the right-hand side of dependencies
- predicate $\mathbf{Const}(\cdot)$ in the left-hand side

Are these features strictly necessary?

Theorem (FKPT07)

Predicate $\mathbf{Const}(\cdot)$ *is necessary for Fagin-inverses of st-tgds:*

Example

$$\mathcal{M}: \quad \begin{array}{rcl} P(x) & \rightarrow & \exists y\ T(y) \land S(x) \\ R(x) & \rightarrow & T(x) \end{array}$$

$$\mathcal{M}': \quad \begin{array}{rcl} T(x) \land \mathbf{Const}(x) & \rightarrow & R(x) \\ S(x) & \rightarrow & P(x) \end{array}$$

$\mathcal{M}$ does not have a Fagin-inverse without $\mathbf{Const}(\cdot)$.

What is the language needed to specify inverses?

Theorem (FKPT07, APR08)

Disjunctions in the right-hand side are necessary for quasi-inverses and maximum recoveries.

For Fagin-inverses we can do better:

Theorem (FKPT07)

Fagin-inverses do not need disjunctions in the right-hand side.

Proof idea

(FKPT07) provide an algorithm that produces a Fagin-inverse specified by tgds + $\mathbf{Const}(\cdot)$ + inequalities in the left-hand side.

The language of inverses is not suitable for data exchange

The language for quasi-inverses and maximum recoveries is not suitable for data exchange.

- how can we chase with disjunctions to materialize an instance?

We would like a natural notion of inverse for st-tgds
that can be expressed in a language with good properties.

Fagin-inverses have this last property, but rarely exists...
Do we have an alternative?

Outline

Fagin-inverse (PODS'06)

Quasi-inverse (PODS'07)

Maximum Recovery (PODS'08)

Computing Inverses

Query language-based inverses (VLDB'09)

Dealing with nulls in source instances (PODS'09)

Relaxation w.r.t. a query language, Arenas et al. VLDB'09

Let **L** be a query language

Definition (APRR09)

$\mathcal{M}'$ is an **L**-recovery of $\mathcal{M}$ iff

$$\underline{\text{certain}}_{\mathcal{M} \circ \mathcal{M}'}(Q, I) \subseteq Q(I)$$

for every source query $Q \in \mathbf{L}$ and instance I.

Definition (APRR09)

$\mathcal{M}'$ is an **L**-maximum recovery of $\mathcal{M}$ iff
for every **L**-recovery $\mathcal{M}''$ of $\mathcal{M}$ we have

$$\underline{\text{certain}}_{\mathcal{M} \circ \mathcal{M}''}(Q, I) \subseteq \underline{\text{certain}}_{\mathcal{M} \circ \mathcal{M}'}(Q, I) \subseteq Q(I)$$

for every source query $Q \in \mathbf{L}$ and instance I.

CQ-maximum recovery

Example

$$\mathcal{M}: \quad \begin{array}{rcl} P(x,y) & \rightarrow & T(x,y) \\ R(x) & \rightarrow & T(x,x) \end{array}$$

$$\mathcal{M}': \quad T(x,y) \wedge x \neq y \rightarrow P(x,y)$$

$\mathcal{M}'$ is a **CQ**-maximum recovery of $\mathcal{M}$.

CQ-maximum recoveries has good properties

Theorem (APRR09)

Every mapping specified by st-tgds $+ \neq$, has a **CQ***-maximum recovery specified by ts-tgds $+ \neq +$* **Const**$(\cdot)$.

Proof idea

Eliminate the disjunctions in maximum recoveries:

- (APRR09) introduce the notion of *product of queries*.
- Then replace $(\psi_1(\bar{x}) \vee \psi_2(\bar{x}))$ by $(\psi_1(\bar{x}) \times \psi_2(\bar{x}))$.

- "Sort of" *closure property*
- The language of **CQ** is *maximal* for the above result.

Outline

- Fagin-inverse (PODS'06)
- Quasi-inverse (PODS'07)
- Maximum Recovery (PODS'08)
- Computing Inverses
- Query language-based inverses (VLDB'09)
- Dealing with nulls in source instances (PODS'09)

What if source instances contain null values?

Do the technical results still hold whit nulls in the source?

- For st-tgds, the existence of max-recoveries is guaranteed since every universal solution is a witness solution.
- If we do not have a *clear distinction* between constant and nulls, universal solutions are no longer witness solutions.

Example

$$\mathcal{M}: \quad \begin{array}{rcl} P(x) & \rightarrow & \exists y \; T(y) \\ R(x) & \rightarrow & T(x) \end{array}$$

For $I = \{P(1)\}$ the instance $J = \{T(n)\}$ is no longer a witness solution:

- J is a solution also for $I' = \{R(n)\}$, but $\mathrm{Sol}_{\mathcal{M}}(I) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I')$.
- $\mathcal{M}$ does not have a maximum recovery when nulls are considered in the source.

Extended mappings

Fagin et al., PODS'09 propose an alternative way to manage mappings with nulls in source instances.

Fagin et al. (FKPT09)

"Do not use nulls in source as constants, but as replaceable values"

Write $I_1 \to I_2$ to state that there is a homomorphism from I_1 to I_2.

(FKPT09): Given a mapping $\mathcal{M}$ with nulls in source and target, define the *extended mapping* $e(\mathcal{M})$ as

$$e(\mathcal{M}) = \{(I, J) \mid \text{ there exists } I' \text{ and } J' \text{ such that } I \to I', (I', J') \in \mathcal{M}, \text{ and } J' \to J\}$$

Maximum extended recovery

Definition (FKPT09)

- $\mathcal{M}'$ is an extended-recovery of $\mathcal{M}$ if

$$\mathsf{Id} \;\subseteq\; e(\mathcal{M}) \circ e(\mathcal{M}')$$

- $\mathcal{M}'$ is a maximum extended-recovery of $\mathcal{M}$ if for every extended recovery $\mathcal{M}''$ of $\mathcal{M}$ we have

$$\mathsf{Id} \;\subseteq\; e(\mathcal{M}) \circ e(\mathcal{M}') \;\subseteq\; e(\mathcal{M}) \circ e(\mathcal{M}'')$$

Theorem (FKPT09)

Every mapping specified by st-tgds considering nulls in source instances has a maximum extended recovery.

Maximum extended recovery

Example

$$\mathcal{M}: \quad P(x,y) \rightarrow \exists z\, S(x,z) \wedge S(z,y)$$

$$\mathcal{M}': \quad S(x,z) \wedge S(z,y) \rightarrow P(x,y)$$

$\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$,
but not a maximum recovery of $\mathcal{M}$

The language of maximum extended recoveries

Theorem (FKPT09)

Mappings specified by full st-tgds always have a maximum extended recovery specified by tgds + $\neq$ + disjunctions

Proof idea

(FKPT09) show that the algorithm in (FKTP07) for computing quasi-inverses of full st-tgds also works in this case.

It is an open problem to identify the exact language needed to express maximum extended recoveries of (general) st-tgds.

Concluding Remarks

- The research on inverting mappings has uncovered an interesting theory
- Challenging theoretical problems
  - Complexity and decidability
  - Algebraic properties, interplay with composition
  - Is there a language closed under inversion?
  - What about different data formats? Inverse for XML-mappings?
- Several issues remain, most importantly *practical issues*

Ron Fagin PODS'06

"The first step in a fascinating journey!"

The Inverse

Jorge Pérez

Departamento de Ciencia de la Computación
Pontificia Universidad Católica de Chile

DEIS'10, Schloss Dagstuhl

References

- *Inverting Schema Mappings* Fagin PODS'06 (also in TODS'07)
- *Quasi-Inverse of Schema Mappings*
  Fagin, Kolaitis, Popa, Tan, PODS'07 (also in TODS'08)
- *The Recovery of a Schema Mapping: Bringing the Exchanged Data Back* Arenas, Pérez, Riveros, PODS'08 (also in TODS'09)
- *Reverse Data Exchange: Copying with Nulls*
  Fagin, Kolaitis, Popa, Tan, PODS'09
- *Inverting Schema Mappings: Bridging the Gap Between Theory and Practice* Arenas, Pérez, Riveros, Reutter, VLDB'09

More on inverses:

- *Composition and Inversion of Schema Mappings*
  Arenas, Pérez, Riveros, Reutter, SIGMOD Record'09
- *The Structure of Inverses in Schema Mappings*
  Fagin, Nash, to appear in JACM

Outline

- Fagin-inverse (PODS'06)
- Quasi-inverse (PODS'07)
- Maximum Recovery (PODS'08)
- Computing Inverses
- Query language-based inverses (VLDB'09)
- Dealing with nulls in source instances (PODS'09)