Graph Data Exchange with Target Constraints

Radu Ciucanu

joint work with
Iovka Boneva and Angela Bonifati

University of Lille, France
INRIA Lille – Nord Europe
Links Project

GraphQ’15
March 27, 2015
Data exchange is the problem of translating data structured under a source schema according to a target schema and a set of constraints\(^1\).

The source and target schema are usually of the same data model e.g., relational, graph, XML.

We study a heterogeneous setting with relational source and graph target.

We additionally consider target constraints, previously ignored in graph data exchange\(^2\).

\(^1\)Fagin, Kolaitis, Miller, Popa. Data exchange: semantics and query answering. *TCS’05*.

\(^2\)Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT’13*.
Table of contents

1 Problem setting

2 Particular cases

3 Results
Problem setting

\( S \) (source schema) – a finite collection of relations.
\( T \) (target schema) – a finite alphabet \( \Sigma \).

- A graph database is a directed, edge-labeled graph.
- In graph data exchange\(^1\), the source and target schema are alphabets.

source: \( \text{Flight}(\text{fid}, \text{src}, \text{dest}) \) and \( \text{Hotel}(\text{fid}, \text{hid}) \)

target: \( \Sigma = \{ f, h \} \)

\(^1\)Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT’13.*
Problem setting

\( S \) (source schema) – a finite collection of relations.
\( T \) (target schema) – a finite alphabet \( \Sigma \).

- A graph database is a **directed, edge-labeled graph**.
- In **graph data exchange**\(^1\), the source and target schema are alphabets.

source: \( \text{Flight}(\text{fid, src, dest}) \) and \( \text{Hotel}(\text{fid, hid}) \)
target: \( \Sigma = \{f, h\} \)

\( M_{st} \) (s-t tgds) – a specification on how relations are translated to graphs.

\[ \text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*), y) \land (y, h, x_4) \land (y, (f \cdot f^*), x_3) \]

\(^1\)Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT’13.*
Problem setting

\( S \) (source schema) – a finite collection of relations.

\( T \) (target schema) – a finite alphabet \( \Sigma \).

- A graph database is a directed, edge-labeled graph.
- In graph data exchange\(^1\), the source and target schema are alphabets.

source: \( \text{Flight}(\text{fid, src, dest}) \) and \( \text{Hotel}(\text{fid, hid}) \)

target: \( \Sigma = \{f, h\} \)

\( M_{st} \) (s-t tgds) – a specification on how relations are translated to graphs.

\[
\text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*), y) \land (y, h, x_4) \land (y, (f \cdot f^*), x_3)
\]

\( M_t \) (target constraints) – previously ignored in graph data exchange\(^1\).

- target egds

\[
(x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1 = x_2)
\]

- sameAs constraints

\[
(x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1, \text{sameAs}, x_2)
\]

\(^1\)Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT’13.*
Solution = a graph satisfying the constraints

\begin{align*}
\text{Flight} &= \begin{array}{ccc}
\text{fid} & \text{src} & \text{dest} \\
01 & c_1 & c_2 \\
02 & c_3 & c_2
\end{array} \\
\text{Hotel} &= \begin{array}{cc}
\text{fid} & \text{hid} \\
01 & hx \\
01 & hy \\
02 & hx
\end{array}
\end{align*}

\begin{align*}
\mathcal{M}_{st} \quad &\text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*), y) \land (y, h, x_4) \land (y, (f \cdot f^*), x_3) \\
\mathcal{M}_t \quad & (x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1 = x_2)
\end{align*}

Extra assumption:

all hotels in the same NULL.
Solution = a graph satisfying the constraints

\[
\begin{array}{ccc}
\text{Flight} &=& \begin{array}{ccc}
\text{fid} & \text{src} & \text{dest} \\
01 & c_1 & c_2 \\
02 & c_3 & c_2 \\
\end{array}
\text{ and Hotel} &=& \begin{array}{cc}
\text{fid} & \text{hid} \\
01 & hx \\
02 & hx \\
\end{array}
\end{array}
\]

\[\begin{array}{c}
\mathcal{M}_{st} \quad \text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*), y) \land (y, h, x_4) \land (y, (f \cdot f^*), x_3) \\
\mathcal{M}_t \quad (x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1 = x_2)
\end{array}\]

Extra assumption:
all hotels in the same NULL.

Extra assumption:
the stop at \( hx \) before the stop at \( hy \).
Solution = a graph satisfying the constraints

\[
\begin{array}{c|c|c}
\text{Flight} & \text{fid} & \text{src} \quad \text{dest} \\
01 & c_1 & c_2 \\
02 & c_3 & c_2 \\
\end{array} \quad \text{and} \quad \begin{array}{c|c}
\text{Hotel} & \text{fid} \quad \text{hid} \\
01 & hx \\
01 & hy \\
02 & hx \\
\end{array}
\]

\[\mathcal{M}_{st} \quad \text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2,(f \cdot f^*),y) \land (y,h,x_4) \land (y,(f \cdot f^*),x_3)\]

\[\mathcal{M}'_t \quad (x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1, \text{sameAs}, x_2)\]

The dotted edges are labeled \text{sameAs}.

Extra assumption:
the stop at \text{hx} before the stop at \text{hy}.

- The \text{sameAs} constraints\(^1\) correspond to a special case of target tgdss\(^2\).

\(^1\)Inspired by from OWL and RDF – \url{http://www.w3.org/wiki/WebSchemas/sameAs}.
\(^2\)Fagin, Kolaitis, Miller, Popa. Data exchange: semantics and query answering. \textit{TCS’05}.
Particular cases
1. Only regular expressions of the form “a” (a ∈ Σ)

- Reduces to a special case of relational data exchange\(^1\).
- The target has only binary relations.
- One can chase a universal solution and query it.

\[ M_{st} \quad Flight(x_1, x_2, x_3) \land Hotel(x_1, x_4) \rightarrow \exists y. (x_2, f, y) \land (y, h, x_4) \land (y, f, x_3) \]

\[ M_t \quad (x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1 = x_2) \]

\(\begin{array}{c|c|c}
\text{Flight} & \text{fid} & \text{src} & \text{dest} \\
\hline
01 & c_1 & c_2 \\
02 & c_3 & c_2 \\
\end{array}\)

\(\begin{array}{c|c}
\text{Hotel} & \text{fid} & \text{hid} \\
\hline
01 & hx \\
01 & hy \\
02 & hx \\
\end{array}\)

Problem: cannot express that \(N_1\) and \(N_2\) are on the same path from \(c_1\) to \(c_2\).

\(^1\)Fagin, Kolaitis, Miller, Popa. Data exchange: semantics and query answering. \textit{TCS'05}.
2. Only s-t tgds (no target constraints)

- Reduces to graph data exchange\(^1\).
- One can chase a universal representative – a graph pattern \(\pi\) s.t. all solutions are graphs where there exists homomorphism from \(\pi\).
- Query answering reduces to querying graph patterns.

\[ M_{st} \quad \text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*), y) \land (y, h, x_4) \land (y, (f \cdot f^*), x_3) \]

\(\begin{array}{c|c|c}
\text{Flight} = & \text{fid} & \text{src} \quad \text{dest} \\
\hline
01 & c_1 & c_2 \\
02 & c_3 & c_2 \\
\end{array}\)

\(\begin{array}{c|c}
\text{Hotel} = & \text{fid} \quad \text{hid} \\
\hline
01 & hx \\
01 & hy \\
02 & hx \\
\end{array}\)

\(^1\)Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT'13.*
Context

1. Only regular expressions of the form “a” (a ∈ Σ):
   - The problem reduces to relational data exchange\(^1\).
   - Our setting is more expressive because we allow recursion (s-t tgds with regular expressions).

2. Only s-t tgds:
   - The problem reduces to graph data exchange\(^2\).
   - Our setting is more expressive because we allow target constraints.

We study the classical problems of interest:
- Existence of solutions.
- Query answering (certain answers).

\(^1\) Fagin, Kolaitis, Miller, Popa. Data exchange: semantics and query answering. *TCS’05.*
\(^2\) Barceló, Pérez, Reutter. Schema mappings and data exchange for graph databases. *ICDT’13.*
Results
Intractability of the problems of interest

<table>
<thead>
<tr>
<th></th>
<th>Target egds</th>
<th>sameAs constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existence of solutions</td>
<td>NP-hard</td>
<td>PTIME</td>
</tr>
<tr>
<td>Query answering</td>
<td>coNP-hard</td>
<td>coNP-hard</td>
</tr>
</tbody>
</table>

Intractability for very **restricted classes**:

- Only regular expressions “a” or “a + b” in the s-t tgds.
- Only regular expressions “a₁ · ... · aₙ” in the target constraints.
- Only regular expressions “a” or “a · a” for query answering.
- Fixed source schema (two unary relations).
- Fixed source instance (hence **query complexity**).

The results for sameAs constraints transfer to general target tgds.
Graph patterns alone cannot be universal representatives

Homomorphisms do not give necessarily a solution

\[ \text{Flight} = \begin{array}{ccc} \text{fid} & \text{src} & \text{dest} \\ 01 & c_1 & c_2 \\ 02 & c_3 & c_2 \end{array} \quad \text{and} \quad \text{Hotel} = \begin{array}{ccc} \text{fid} & \text{hid} \\ 01 & \text{hx} \\ 01 & \text{hy} \\ 02 & \text{hx} \end{array} \]

\[ \mathcal{M}_{st} \quad \text{Flight}(x_1, x_2, x_3) \land \text{Hotel}(x_1, x_4) \rightarrow \exists y. (x_2, (f \cdot f^*) , y) \land (y, h, x_4) \land (y, (f \cdot f^*) , x_3) \]

\[ \mathcal{M}_t \quad (x_1, h, x_3) \land (x_2, h, x_3) \rightarrow (x_1 = x_2) \]

Problem:
Graph G is not a solution even though there is a homomorphism from the chased pattern \( \pi \) to \( G \).

We cannot use the same definition of universal representatives as without target constraints.
Conclusions

Problem setting:
- **Data exchange** with relational source and **graph** target.
- **Target constraints**.

Results:
- The problems of interest are **intractable** (query complexity).
- Graph patterns alone cannot be **universal representatives**.

Open questions:
- **Upper bounds** for the problems of interest.
- Definition and **query** mechanisms for universal representatives.