Comparing the Expressive Power of Data Integration Systems: Work in Progress

Marcelo Arenas[1] Pablo Barceló[2] Juan Reutter[3]

[1]PUC Chile
[2]U. of Chile
[3]PUC Chile

Data integration

What is data integration?

The problem of combining data residing at different sources, and providing the user with a unified view of them.

Data integration

What is data integration?

The problem of combining data residing at different sources, and providing the user with a unified view of them.

Why data integration?

- Important in real-world applications.
- Characterized by issues that are interesting from a theoretical point of view.

Data integration setting

A data integration system $\mathcal{DI}$ consists of:

- A relational source schema $\mathcal{S} = \{S_1, \ldots, S_n\}$.
- A relational global schema $\mathcal{G} = \{G_1, \ldots, G_m\}$.
- A set Σ of mappings:

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

that specify the relationship between source and global schema.
Here $\phi_{\mathcal{S}}$ and $\psi_{\mathcal{G}}$ are logical formulas over $\mathcal{S}$ and $\mathcal{G}$, respectively.

GLAV, GAV, LAV

We restrict our attention to the following classes of data integration systems:

- Global-as-view (GAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow G(\bar{x})$$

where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$.

- Local-as-view (LAV): Each mapping is of the form

$$S(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

where $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

- Global/Local-as-view (GLAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$ and $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

GLAV, GAV, LAV

We restrict our attention to the following classes of data integration systems:

- Global-as-view (GAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow G(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$.

- Local-as-view (LAV): Each mapping is of the form

$$S(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

- Global/Local-as-view (GLAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$ and $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

GLAV, GAV, LAV

We restrict our attention to the following classes of data integration systems:

- Global-as-view (GAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow G(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$.

- Local-as-view (LAV): Each mapping is of the form

$$S(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

- Global/Local-as-view (GLAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$ and $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

GLAV, GAV, LAV

We restrict our attention to the following classes of data integration systems:

- Global-as-view (GAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow G(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$.

- Local-as-view (LAV): Each mapping is of the form

$$S(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

- Global/Local-as-view (GLAV): Each mapping is of the form

$$\phi_{\mathcal{S}}(\bar{x}) \rightarrow \psi_{\mathcal{G}}(\bar{x})$$

  where $\phi_{\mathcal{S}}$ is a conjunction of atoms over $\mathcal{S}$ and $\psi_{\mathcal{G}}$ is a conjunctive query over $\mathcal{G}$.

GAV v/s LAV

Why LAV and Why GAV?

- LAV approach is declarative as the content of the source is characterized by views over the global schema.
  It favors the extensibility of the system (a new source just needs a new view over the global schema).
  Computing certain answers is hard even for $\mathrm{CQ}^{\neq}$ [Abiteboul, Dushcka; PODS'98].
- The GAV approach is procedural as the mapping explicitly says how to retrieve data from the source to compute certain answers.
  Computing certain answers is easy for all monotone queries (simple unfolding of the query).
  However, in GAV it is not easy to extend the system with new sources (all views may have to be redefined).

GAV v/s LAV

Why LAV and Why GAV?

- LAV approach is declarative as the content of the source is characterized by views over the global schema.
  It favors the extensibility of the system (a new source just needs a new view over the global schema).
  Computing certain answers is hard even for $\mathrm{CQ}^{\neq}$ [Abiteboul, Dushcka; PODS'98].
- The GAV approach is procedural as the mapping explicitly says how to retrieve data from the source to compute certain answers.
  Computing certain answers is easy for all monotone queries (simple unfolding of the query).
  However, in GAV it is not easy to extend the system with new sources (all views may have to be redefined).

GAV v/s LAV

Why LAV and Why GAV?

- LAV approach is declarative as the content of the source is characterized by views over the global schema.
  It favors the extensibility of the system (a new source just needs a new view over the global schema).
  Computing certain answers is hard even for $\mathrm{CQ}^{\neq}$ [Abiteboul, Dushcka; PODS'98].
- The GAV approach is procedural as the mapping explicitly says how to retrieve data from the source to compute certain answers.
  Computing certain answers is easy for all monotone queries (simple unfolding of the query).
  However, in GAV it is not easy to extend the system with new sources (all views may have to be redefined).

Querying in data integration

In data integration queries are posed over the global schema (the reconciled, virtual view that is shown to the user).

However, real data belongs to the source.

Querying in data integration

In data integration queries are posed over the global schema (the reconciled, virtual view that is shown to the user).

However, real data belongs to the source.

Thus, semantics of query Q over $\mathcal{G}$ in data integration system $\mathcal{DI}$ is given by source instance I:

- A solution for I on $\mathcal{DI}$ is a global instance J such that $(I, J) \models \Sigma$.
- Since there are multiple solutions for I on $\mathcal{DI}$ the semantics for Q is:

$$\text{certain}(Q, \mathcal{DI}, I) = \bigcap_{J \text{ is a solution for } I} Q(J)$$

Example

Consider the following LAV system:

$$\begin{aligned} \text{male}(x) &\rightarrow \exists y\, \text{couple}(x, y) \\ \text{female}(x) &\rightarrow \exists y\, \text{couple}(y, x) \end{aligned}$$

If source instance I is such that $I^{\text{male}} = \{\text{Pablo}\}$ and $I^{\text{female}} = \{\text{Magdalena}\}$ then:

- The set of certain answers to query $\text{couple}(x, y)$ is $\emptyset$.
- The set of certain answers to query $\exists y\, \text{couple}(x, y)$ is $\{\text{Pablo}\}$.
- The set of certain answers to query $\exists y\, \text{couple}(y, x)$ is $\{\text{Magdalena}\}$.

This talk

This talk is about:

- Comparison of the expressive power of data integration settings: GLAV, GAV, LAV.
- Query rewriting in data integration: Expressing a query over the global schema in terms of the source.

Query-preserving transformations: Motivation

The study of the compared expressive power of data integration systems was started in [Calì, Calvanese, De Giacomo, Lenzerini; ER'02].

Motivation: To study when a declarative approach could be transformed into a procedural one, and viceversa.

Query-preserving transformations: Definition

They used the following definition:

Given $\mathcal{C}, \mathcal{C}'$ classes of data integration systems. We say that $\mathcal{C}$ is query-reducible to $\mathcal{C}'$, if for every setting $\mathcal{DI} \in \mathcal{C}$ there exists setting $\mathcal{DI}' \in \mathcal{C}'$ (with same source and global schema) such that

$$\text{certain}(Q, \mathcal{DI}, I) \;=\; \text{certain}(Q, \mathcal{DI}', I)$$

for every source instance I and conjunctive query Q over the global schema.

Results on query-preserving transformations

Proposition: [CCGL, '03] The class of GAV settings is not query-reducible to the class of LAV settings, and viceversa.

More positive results can be obtained if constraints on the global schema are allowed.

Results on query-preserving transformations

Proposition: [CCGL, '03] The class of GAV settings is not query-reducible to the class of LAV settings, and viceversa.

More positive results can be obtained if constraints on the global schema are allowed.

We think that this approach has several drawbacks:

- It only deals with conjunctive queries.
- Queries are preserved, i.e. no capability to re-express the query is allowed.
- Target constraints quickly lead to undecidability.

Non-query-preserving transformations

Thus, we have designed a new (more liberal) setting:

Given $\mathcal{C}, \mathcal{C}'$ classes of data integration systems, and $\mathcal{L}, \mathcal{L}'$ fragments of FO logic.

We say that $\mathcal{C}$ is reducible to $\mathcal{C}'$ from $\mathcal{L}$ to $\mathcal{L}'$, if for every setting $\mathcal{DI} \in \mathcal{C}$ there exists setting $\mathcal{DI}' \in \mathcal{C}'$ (with same source schema than $\mathcal{DI}$) such that for every Q in $\mathcal{L}$ there exists query Q' in $\mathcal{L}'$ satisfying that

$$\text{certain}(Q, \mathcal{DI}, I) \;=\; \text{certain}(Q', \mathcal{DI}', I)$$

for every source instance I.

Non-query-preserving transformations

Thus, we have designed a new (more liberal) setting:

Given $\mathcal{C}, \mathcal{C}'$ classes of data integration systems, and $\mathcal{L}, \mathcal{L}'$ fragments of FO logic.

We say that $\mathcal{C}$ is reducible to $\mathcal{C}'$ from $\mathcal{L}$ to $\mathcal{L}'$, if for every setting $\mathcal{DI} \in \mathcal{C}$ there exists setting $\mathcal{DI}' \in \mathcal{C}'$ (with same source schema than $\mathcal{DI}$) such that for every Q in $\mathcal{L}$ there exists query Q' in $\mathcal{L}'$ satisfying that

$$\text{certain}(Q, \mathcal{DI}, I) \;=\; \text{certain}(Q', \mathcal{DI}', I)$$

for every source instance I.

This allows us to obtain a bigger number of positive results, as well as stronger negative results.

Results on non-query-preserving transformations

In the following we put together our own results with previous results in the DI literature to study this notion.

From now on $\mathcal{C} \preceq_{\mathcal{L},\mathcal{L}'} \mathcal{C}'$ if $\mathcal{C}$ is reducible to $\mathcal{C}'$ from $\mathcal{L}$ to $\mathcal{L}'$.

Results on non-query-preserving transformations

In the following we put together our own results with previous results in the DI literature to study this notion.

From now on $\mathcal{C} \preceq_{\mathcal{L},\mathcal{L}'} \mathcal{C}'$ if $\mathcal{C}$ is reducible to $\mathcal{C}'$ from $\mathcal{L}$ to $\mathcal{L}'$.

Query languages we will consider (among others):

- FO (First-order logic)
- CQs (Conjunctive queries).
- UCQs (Unions of conjunctive queries)
- $\mathrm{CQ}^{\neq}$ (Conjunctive queries with inequalities).
- $\mathrm{UCQ}^{\neq}$ (Unions of conjunctive queries with inequalities.

Results on ...

We start with a fairly general result:

Proposition: GLAV $\preceq_{\mathrm{FO},\mathrm{FO}}$ GAV and GLAV $\preceq_{\mathrm{FO},\mathrm{FO}}$ LAV.

Proof idea: Copy source to target, and codify mappings of the GLAV setting into the new query.

Results on ...

We start with a fairly general result:

Proposition: GLAV $\preceq_{\mathrm{FO,FO}}$ GAV and GLAV $\preceq_{\mathrm{FO,FO}}$ LAV.

Proof idea: Copy source to target, and codify mappings of the GLAV setting into the new query.

Not really useful as computing certain answers for arbitrary FO queries is undecidable.

Results on ...

What if we start from a decidable fragment, e.g. CQs?

Results on ...

What if we start from a decidable fragment, e.g. CQs?

Then we can refine our result, but not much:

Proposition: GLAV $\preceq_{\mathrm{CQ},\exists^*\forall^*}$ GAV and GLAV $\preceq_{\mathrm{CQ},\exists^*\forall^*}$ LAV
(In general, GLAV $\preceq_{\exists^*\forall^*,\exists^*\forall^*}$ GAV and GLAV $\preceq_{\exists^*\forall^*,\exists^*\forall^*}$ LAV).

Again, not really useful: Fragment $\exists^*\forall^*$ undecidable in terms of certain answers.

Results on ...

What if we start from a decidable fragment, e.g. CQs?

Then we can refine our result, but not much:

Proposition: GLAV $\preceq_{\mathrm{CQ},\exists^*\forall^*}$ GAV and GLAV $\preceq_{\mathrm{CQ},\exists^*\forall^*}$ LAV
(In general, GLAV $\preceq_{\exists^*\forall^*,\exists^*\forall^*}$ GAV and GLAV $\preceq_{\exists^*\forall^*,\exists^*\forall^*}$ LAV).

Again, not really useful: Fragment $\exists^*\forall^*$ undecidable in terms of certain answers.

Question: How much can we refine the results if we restrict both query languages and settings, e.g. from LAV to GAV?

GAV to LAV

This is not really the most interesting direction: procedural to declarative.

Results are fairly good:

Proposition: GAV $\preceq_{\mathrm{CQ},\mathrm{UCQ}}$ LAV and GAV $\preceq_{\mathrm{CQ}^{\neq},\mathrm{UCQ}^{\neq}}$ LAV.

Proof: Simple unfolding.

But, on the negative side:

Proposition: GAV $\not\preceq_{\mathrm{CQ},\mathrm{CQ}^{\neq}}$ LAV.

Proof idea: Construct the right query.

LAV to GAV

This is really the most interesting case: declarative to procedural.

The following holds from [Halevy, Mendelzon, Sagiv, Srivastava; PODS'05].

Proposition: LAV $\preceq_{\mathrm{CQ},\mathrm{UCQ}}$ GAV.

However, on the negative side:

Proposition: LAV $\not\preceq_{\mathrm{CQ}^{\neq},\mathrm{UCQ}^{\neq}}$ GAV
(Indeed, not even LAV $\preceq_{\mathrm{CQ}^{\neq},\mathrm{monotone}}$ GAV).

LAV to GAV and rewriting over the source

The last negative result follows from [Fagin, Kolaitis, Miller, Popa; ICDT'03]:

There is a LAV system and a conjunctive query Q with one inequality such that the certain answers to Q cannot be recovered by an FO rewriting Q' over the source.

LAV to GAV and rewriting over the source

The last negative result follows from [Fagin, Kolaitis, Miller, Popa; ICDT'03]:

There is a LAV system and a conjunctive query Q with one inequality such that the certain answers to Q cannot be recovered by an FO rewriting Q' over the source.

A natural question is then: When is it possible to rewrite a conjunctive query with inequalities over the source?

Problems we are currently studying

Two interesting issues related to the last question:

- Decidability of the following problem: Given a LAV setting and a $\mathrm{CQ}^{\neq}$ Q, is there a FO rewriting Q' of Q over the source?
- Find sufficient conditions for a $\mathrm{CQ}^{\neq}$ Q over a LAV system to have an FO rewriting over the source.