Models for information integration: turning local-as-view into global-as-view

Andrea Calì, Giuseppe De Giacomo, Maurizio Lenzerini

Dipartimento di Informatica e Sistemistica
Università di Roma "La Sapienza"
Via Salaria 113, I-00198 Roma, Italy
lastname@dis.uniroma1.it,
http://www.dis.uniroma1.it/~*lastname*

Abstract. There are basically two approaches for designing a data integration system. In the global-as-view approach, one defines the concepts in the global schema as views over the sources, whereas in the local-as-view approach, one characterizes the sources as views over the global schema. The goal of this paper is to verify whether we can transform a data integration system built with the local-as-view approach into a system following the global-as-view approach. We study the problem in a setting where the global schema is expressed in the relational model with inclusion dependencies, and the queries used in the integration systems (both the queries on the global schema, and the views in the mapping) are expressed in the language of conjunctive queries. The result we present is that such a transformation exists: we can always transform a local-as-view system into a global-as-view system such that, for each query, the set of answers to the query wrt the former is the same as the set of answers wrt the latter.

1 Introduction

Data integration is the problem of combining the data residing at different sources, and providing the user with a unified view of these data, called global (or, mediated) schema [9,8]. The global schema is therefore a reconciled view of the information, which can be queried by the user. It is the task of the data integration system to free the user from the knowledge on where data are, how data are structured at the sources, and how data are to be merged and reconciled to fit into the global schema.

The interest in this kind of systems has been continuously growing in the last years. Many organizations face the problem of integrating data residing in several sources. Companies that build a Data Warehouse, a Data Mining, or an Enterprise Resource Planning system must address this problem. Also, integrating data in the World Wide Web is the subject of several investigations and projects nowadays. Finally, applications requiring accessing or re-engineering legacy systems must deal with the problem of integrating data stored in preexisting sources.

The design of a data integration system is a very complex task, which comprises several different issues. Here, we concentrate on the following issues:

1. Specifying the mapping between the global schema and the sources,
2. Processing queries expressed on the global schema.

With regard to issue (1), two basic approaches have been used to specify the mapping between the sources and the global schema [9–11]. The first approach, called *global-as-view* (or global-centric), requires that the global schema is expressed in terms of the data sources. More precisely, to every element of the global schema, a view over the data sources is associated, so that its meaning is specified in terms of the data residing at the sources. The second approach, called *local-as-view* (or source-centric), requires the global schema to be specified independently from the sources. In turn, the sources are defined as views over the global schema. The relationships between the global schema and the sources are thus established by specifying the information content of every source in terms of a view over the global schema.

Issue (2) is concerned with one of the most important problems in the design of a data integration system, namely, the method for computing the answer to queries. The basic assumption regarding query processing is that the queries posed to the system are expressed in terms of the global schema, and, therefore, the system should be able to re-express the query in terms of a suitable set of queries posed to the sources. In this reformulation process, the crucial step is deciding how to decompose the query on the global schema into a set of subqueries on the sources, based on the meaning of the mapping. The computed subqueries are then shipped to the sources, and the results are assembled into the final answer.

A comparison of the local-as-view and global-as-view approaches is reported in [14]. It is known that the former approach ensures an easier extensibility of the integration system, and provides a more appropriate setting for its maintenance. For example, adding a new source to the system requires only to provide the definition of the source, and does not necessarily involve changes in the global view. On the contrary, in the global-as-view approach, adding a new source may in principle require changing the definition of the concepts in the global schema.

It is also well known that processing queries in the local-as-view approach is a difficult task [13, 14, 7, 1, 6, 3, 4]. Indeed, in this approach, the only knowledge we have about the data in the global schema is through the views representing the sources, and such views provide only partial information about the data. Since the mapping associates to each source a view over the global schema, it is not immediate to infer how to use the sources in order to answer queries expressed over the global schema. Thus, extracting information from the data integration system is similar to query answering with incomplete information, which is a complex task [15]. On the other hand, query processing looks much easier in the global-as-view approach, where we can take advantage that the mapping directly specify which source queries corresponds to the elements of the global schema. Besides these intuitive considerations, a deep analysis of the differences/similarities of two approaches is still missing.

The goal of this paper is to investigate on the relative expressive power of the two approaches. In particular, we address the problem of checking whether we can transform a data integration system built with the local-as-view approach into a system following the global-as-view approach. Obviously, we are interested in an equivalent transformation, in the sense that we want that queries posed to the latter have the same answers than queries posed to the former. We study the problem in a setting where the global schema is expressed in the relational model with inclusion dependencies, and the queries used in the integration systems (both the queries on the global schema, and the queries in the mapping) are expressed in the language of conjunctive queries. The result we present is that such a transformation exists: given a local-as-view system L, we can always transform it into a query-equivalent global-as-view system, i.e., a system G such that, for each query q, the set of answers to q wrt L is the same as the set of answers to q wrt G. We observe, however, that the presence of inclusion dependencies in the global schema is crucial for the transformation.

The paper is organized as follows. In Section 2 we describe the formal framework we use for data integration, by describing the main components of a data integration system, namely, the global schema, the sources, and the mapping between the two, and by specifying the precise semantics of the system. In Section 3 we present the result that shows that, in the presented framework, the global-as-view approach has at least the same expressive power than the local-as-view approach. Section 4 concludes the paper.

2 Framework for information integration

In this section we set up a formal framework for data integration. In particular, we describe the main components of a data integration system, namely, the global schema, the sources, and the mapping between the two. Finally, we provide the semantics both of the system, and of query answering.

The formal definition of a data integration system is given below.

Definition 1. *A* data integration system $\mathcal{I}$ *is a triple* $\langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$*, where* $\mathcal{G}$ *is the global schema,* $\mathcal{S}$ *is the source schema, and* $\mathcal{M}$ *is the mapping between* $\mathcal{G}$ *and* $\mathcal{S}$*.*

We denote with $\mathcal{A}_{\mathcal{G}}$ and $\mathcal{A}_{\mathcal{S}}$ the finite alphabets for the elements of the global schema and the elements of the sources, respectively. We consider the global schema constituted by the elements named with symbols of $\mathcal{A}_{\mathcal{G}}$, and by a set of constraints. The source schema describes the structure of the various data sources. The mapping $\mathcal{M}$ establishes a relationship between elements of the global schema $\mathcal{G}$ and those of the source schema $\mathcal{S}$. (i.e. the sources).

We assume that the databases involved in our framework (both global databases and source databases) are defined over a fixed (infinite) alphabet Γ of symbols. In order to assign semantics to a data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$, we start by considering a *source database* for $\mathcal{I}$, i.e., a database $\mathcal{D}$ for the source schema $\mathcal{S}$. Based on $\mathcal{D}$, we now specify which is the information content of the global schema $\mathcal{G}$. We call *global database* for $\mathcal{I}$ any database for $\mathcal{G}$. A global database $\mathcal{B}$ for $\mathcal{I}$ is said to be *legal for $\mathcal{I}$ with respect to $\mathcal{D}$*, if:

- $\mathcal{B}$ is legal with respect to $\mathcal{G}$, i.e. $\mathcal{B}$ satisfies all the constraints of $\mathcal{G}$;
- $\mathcal{B}$ satisfies the mapping $\mathcal{M}$ wrt $\mathcal{D}$.

The notion of $\mathcal{B}$ satisfying the mapping $\mathcal{M}$ wrt $\mathcal{D}$ depends on the type of the mapping. As we said in the introduction, two basic approaches, namely global-as-view (GAV) and local-as-view (LAV), have been proposed for specifying the mapping. In the GAV approach, the mapping $\mathcal{M}$ associates to each element r in $\mathcal{G}$ a query over $\mathcal{S}$, denoted by $\rho(r)$. We say that $\mathcal{B}$ satisfies $\mathcal{M}$ wrt $\mathcal{D}$ if, for each element r of $\mathcal{G}$, the set of tuples $r^{\mathcal{B}}$ that $\mathcal{B}$ assigns to r contains the set of tuples $\rho(r)^{\mathcal{D}}$ that satisfy the query $\rho(r)$ in $\mathcal{D}$, i.e.

$$\rho(r)^{\mathcal{D}} \subseteq r^{\mathcal{B}}.$$

Note that this means that the view associated to r is *sound*: the data provided by the sources satisfy the element of global schema, but are not necessarily complete.

In the LAV approach, instead, the mapping $\mathcal{M}$ associates to each source s in $\mathcal{S}$ a query over $\mathcal{G}$, denoted by $\rho(s)$. In this case, we say that $\mathcal{B}$ satisfies $\mathcal{M}$ wrt $\mathcal{D}$, if for each source s of $\mathcal{S}$, the set of tuples $s^{\mathcal{D}}$ that $\mathcal{D}$ assigns to s is contained in the set of tuples $\rho(s)^{\mathcal{B}}$ that satisfy the query $\rho(s)$ in $\mathcal{B}$, i.e.

$$s^{\mathcal{D}} \subseteq \rho(s)^{\mathcal{B}}.$$

Note that, analogously to the previous case, this means that the view associated to s is *sound*.

Queries posed to a data integration system $\mathcal{I}$ are expressed in terms of a query language $\mathcal{Q}_{\mathcal{G}}$ over the alphabet $\mathcal{A}_{\mathcal{G}}$, i.e., over the global schema. Given a source database $\mathcal{D}$ for $\mathcal{I}$, the answer $q^{\mathcal{I},\mathcal{D}}$ to a query q to $\mathcal{I}$ wrt $\mathcal{D}$, is the set of tuples $(c_1, \ldots, c_n) \in \Gamma^n$ such that $(c_1, \ldots, c_n) \in q^{\mathcal{B}}$ for each global database $\mathcal{B}$ legal for $\mathcal{I}$ wrt $\mathcal{D}$, where $q^{\mathcal{DB}}$ denotes the result of evaluating the query q over the database $\mathcal{DB}$. Since, in general, several global databases exist that are legal for $\mathcal{I}$ wrt $\mathcal{D}$, in the terminology of data integration, $q^{\mathcal{I},\mathcal{D}}$ is often called the set of *certain answers* of q wrt $\mathcal{D}$.

In the rest of this paper, we deal with data integration systems with the following characteristics.

- The global schema $\mathcal{G}$ is expressed in the *relational model with inclusion dependencies*. A global database $\mathcal{B}$ is legal wrt $\mathcal{G}$ if it respects all inclusion dependencies in $\mathcal{G}$. To each symbol of $\mathcal{A}_{\mathcal{G}}$, which denotes a relation, we associate an *arity*, which is the arity of the corresponding relation. For the sake of simplicity, we denote the attributes of a relation of arity n with the natural numbers $1 \ldots n$. We remind the reader that, given two sequences of distinct attributes $\mathbf{A} = \langle A_1, \ldots, A_n \rangle$ and $\mathbf{B} = \langle B_1, \ldots, B_n \rangle$, belonging to relations r_1, r_2 respectively, and denoting with $r_1[\mathbf{A}]$ and $r_2[\mathbf{B}]$ the projection of r_1 and r_2 over $\mathbf{A}$ and $\mathbf{B}$ respectively, an inclusion dependency between

r_1 and r_2 is denoted by $r_1[\mathbf{A}] \subseteq r_2[\mathbf{B}]$. Such a dependency is satisfied in a database $\mathcal{DB}$ if for each tuple $t \in r_1^{\mathcal{DB}}$ there exists a tuple $t' \in r_2^{\mathcal{DB}}$ such that $t[A_i] = t'[B_i]$ for each $i \in \{1, \dots, n\}$.

- The source schema $\mathcal{S}$ is also expressed in the *relational model*. To each source we associate a symbol of $\mathcal{A}_\mathcal{S}$. Each symbol has an associated *arity*, which is the arity of the corresponding source relation.
- The language of the views used in the mapping is that of *conjunctive queries* (CQs). We remind that a conjunctive query q of arity n over a set $\mathcal{R}$ of relations is written in the form

$$\{ \langle X_1, \dots, X_n \rangle \mid conj(X_1, \dots, X_n, Y_1, \dots, Y_m) \}$$

  where $conj(X_1, \dots, X_n, Y_1, \dots, Y_m)$ is a conjunction of atoms involving constants and the variables $X_1, \dots, X_n, Y_1, \dots, Y_m$, and the predicate symbols of the atoms are in $\mathcal{R}$. Given a database $\mathcal{DB}$, the answer $q^{\mathcal{DB}}$ of q over $\mathcal{DB}$ is the set of tuples $\langle c_1, \dots, c_n \rangle$ of constants in $\mathcal{DB}$ such that for some constants $d_1, \dots, d_m$ in $\mathcal{DB}$, every atom in $conj(c_1, \dots, c_n, d_1, \dots, d_m)$ is true in $\mathcal{DB}$.
- The language of the user queries, posed over the global schema, is again that of conjunctive queries.

3 From LAV to GAV

We show that we can reformulate a LAV integration system as a GAV integration system on the same sources that is equivalent wrt query processing to the original one. First, let us be precise on what kind of equivalence we are after.

Definition 2. *Let $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ be an integration system, having $\mathcal{A}_\mathcal{G}$ as the alphabet of the global schema. An integration system $\mathcal{I}'$ on $\mathcal{S}$ is* query-equivalent *to $\mathcal{I}$ if for every query q using only the symbols in $\mathcal{A}_\mathcal{G}$, and for every source databases $\mathcal{D}$:*

$$q^{\mathcal{I},\mathcal{D}} = q^{\mathcal{I}',\mathcal{D}}$$

In other words, we say that $\mathcal{I}'$ is query-equivalent to $\mathcal{I}$ if given any query on the alphabet of the global schema of $\mathcal{I}$, the certain answers we get for the query on the two integration systems is identical.

Let $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ be a LAV integration system. From $\mathcal{I}$ we define a correspondent GAV integration system $\mathcal{I}' = \langle \mathcal{G}', \mathcal{S}, \mathcal{M}' \rangle$ as follows. First, for the sake of simplicity, we assume that in all queries in the mapping $\mathcal{M}$, each variable appears at most once in each atom. Our reformulation technique can be easily extended to the general case, by considering a slightly extended class of constraints in the global schema [2].

Then we define $\mathcal{I}'$ as follows.

- The set of sources $\mathcal{S}$ remains unchanged.
- The global schema $\mathcal{G}'$ is obtained from $\mathcal{G}$ by introducing:
  - a new relation $\mathsf{sourceImage}/n$ (source image) for each relation source/n in $\mathcal{S}$;

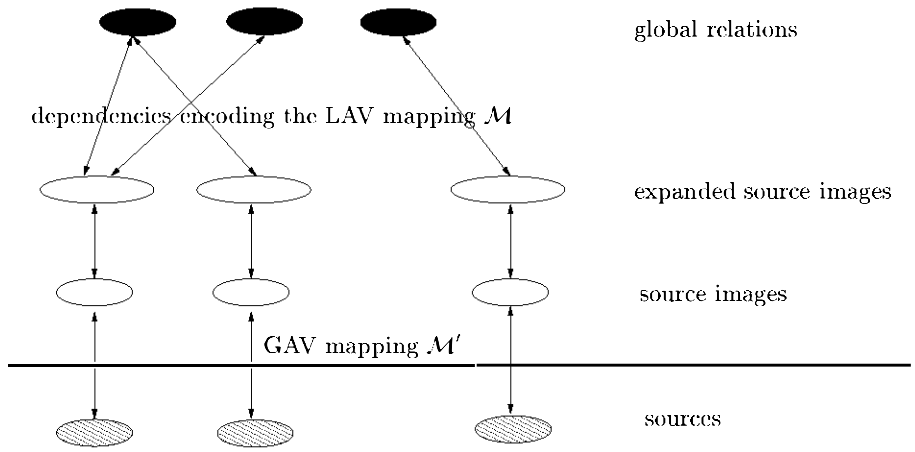

Fig. 1. Representation of the GAV system obtained after the transformation

- a new relation sourceImageExp/$(n+m)$ (expanded source image) for each relation source/n in $\mathcal{S}$, where m is the number of non-distinguished variables appearing in the query ρ(source); we assume variables in ρ(source) to be enumerated as $Z_1, \ldots, Z_{n+m}$, with $Z_1, \ldots, Z_n$ being the distinguished variables;

and by adding the following dependencies:

- for each relation source/n in $\mathcal{S}$ we add the inclusion dependency

$$\mathsf{sourceImage}[1, \ldots, n] \subseteq \mathsf{sourceImageExp}[1, \ldots, n]$$

- for each relation source/n in $\mathcal{S}$ and for each atom $g(Z_{i_1}, \ldots, Z_{i_k})$ occurring in ρ(source), we add the inclusion dependency

$$\mathsf{sourceImageExp}[i_1, \ldots, i_k] \subseteq \mathsf{g}[1, \ldots, k]$$

– The GAV mapping $\mathcal{M}'$ associates to each global relation sourceImage the query

$$\rho'(\mathsf{sourceImage}) = \{\langle X_1, \ldots, X_n \rangle \mid \mathsf{source}(X_1, \ldots, X_n)\}$$

and to the remaining global relations the empty query.

It is immediate to verify that given a LAV integration system $\mathcal{I}$, and being $\mathcal{I}'$ the correspondent GAV integration system defined as above, the size of $\mathcal{I}'$ is linearly related to the size of $\mathcal{I}$.

In Figure 1 a graphic representation of the obtained GAV system is shown. Above the horizontal line is the global schema, while below is the source schema. The role of the source imagesis to make the data at the sources available in the global schema, in order to express the original mapping $\mathcal{M}$ in the global schema itself. Their relationship with the sources is encoded in the GAV mapping. The original LAV mapping is encoded in the inclusion dependencies in $\mathcal{G}'$, together with the relationship among the source images and the expanded source images.

Let us illustrate the transformation technique we have presented with an example.

Example 1. Consider a LAV integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ where:

- The global schema $\mathcal{G}$ is simply constituted by the relations $\mathsf{cites}/2$, expressing that a paper cites another paper, and $\mathsf{sameTopic}/2$, expressing that two papers are on the same topic.
- The set of sources $\mathcal{S}$ is constituted by two relations: source_1 containing pairs of papers that mutually cite each other; and source_2 containing pairs of papers on the same topic, each with at least one citation.
- The mapping $\mathcal{M}$ between the sources and the global view is:

$$\begin{aligned} \rho(\mathsf{source}_1) &= \{\langle X, Y \rangle \mid \mathsf{cites}(X, Y) \wedge \mathsf{cites}(Y, X)\} \\ \rho(\mathsf{source}_2) &= \{\langle X, Y \rangle \mid \mathsf{sameTopic}(X, Y) \wedge \mathsf{cites}(X, Z) \wedge \mathsf{cites}(Y, W)\} \end{aligned}$$

Then the correspondent GAV integration system $\mathcal{I}' = \langle \mathcal{G}', \mathcal{S}, \mathcal{M}' \rangle$ is as follows:

- The set of sources $\mathcal{S}$ remains unchanged.
- The global schema $\mathcal{G}'$ is constituted by the relations $\mathsf{cites}/2$, $\mathsf{sameTopic}/2$ as before, and the additional relations $\mathsf{sourceImage}_1/2$, $\mathsf{sourceImage}_2/2$, $\mathsf{sourceImageExp}_1/2$, and $\mathsf{sourceImageExp}_2/4$. On such relations are defined the following inclusion dependencies:

$$\begin{aligned} \mathsf{sourceImage}_1[1,2] &\subseteq \mathsf{sourceImageExp}_1[1,2] \\ \mathsf{sourceImage}_2[1,2] &\subseteq \mathsf{sourceImageExp}_2[1,2] \\ \mathsf{sourceImageExp}_1[1,2] &\subseteq \mathsf{cites}[1,2] \\ \mathsf{sourceImageExp}_1[2,1] &\subseteq \mathsf{cites}[1,2] \\ \mathsf{sourceImageExp}_2[1,3] &\subseteq \mathsf{cites}[1,2] \\ \mathsf{sourceImageExp}_2[2,4] &\subseteq \mathsf{cites}[1,2] \\ \mathsf{sourceImageExp}_2[1,2] &\subseteq \mathsf{sameTopic}[1,2] \end{aligned}$$

- The mapping $\mathcal{M}'$ is:

$$\begin{aligned} \rho'(\mathsf{sourceImage}_1) &= \{\langle X, Y \rangle \mid \mathsf{source}_1(X, Y)\} \\ \rho'(\mathsf{sourceImage}_2) &= \{\langle X, Y \rangle \mid \mathsf{source}_2(X, Y)\} \end{aligned}$$

■

Next we show that the LAV integration system $\mathcal{I}$ and the correspondent GAV integration system $\mathcal{I}'$ obtained as above are indeed query-equivalent. The claim is based on the observation that the semantics of any integration system $\mathcal{I}$, either GAV or LAV, can be captured by a suitable *logic program* $\mathcal{P}_\mathcal{I}$ [12].

We first concentrate on GAV systems. If $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ is a GAV integration system, we can associate a logic program $\mathcal{P}_\mathcal{I}$ is defined as follows:

– For each inclusion dependency $g_1[\mathbf{A}] \subseteq g_2[\mathbf{B}]$ in $\mathcal{G}$ where $\mathbf{A}$ and $\mathbf{B}$ are sets of attributes, we have a rule of the form (assuming for simplicity that the attributes in $\mathbf{A}$ and $\mathbf{B}$ are the first h in g_1 and g_2, respectively):

$$g_2(X_1, \ldots, X_h, f_1(X_1, \ldots, X_h), \ldots, f_{n-h}(X_1, \ldots, X_h)) \leftarrow \\ g_1(X_1, \ldots, X_h, \ldots, X_m)$$

where f_i are fresh Skolem functions.

– For each query $\rho(g) = \{\langle X_1, \ldots, X_n \rangle \mid conj(X_1, \ldots, X_n, Y_1, \ldots, Y_n)\}$ in the mapping $\mathcal{M}$, we have a rule of the form:

$$g(X_1, \ldots, X_n) \leftarrow conj(X_1, \ldots, X_n, Y_1, \ldots, Y_n)$$

In addition, the relations in $\mathcal{S}$ can be seen as predicates that are given extensionally. That is, a source database $\mathcal{D}$ for $\mathcal{I}$ can be seen as a finite set of ground facts in logic programming terms.

By applying results from the logic programming theory [12], it is not difficult to show the following lemma.

Lemma 1. *Let $\mathcal{I}$ be a GAV integration system, $\mathcal{D}$ a source database for $\mathcal{I}$, $\mathcal{P}_\mathcal{I}$ the corresponding logic program defined above, and M_{min} the minimal model of $\mathcal{P}_I \cup \mathcal{D}$. Then given a query q over $\mathcal{G}$, for every tuple $\langle c_1, \ldots, c_n \rangle$ in Γ^n:*

$$\langle c_1, \ldots, c_n \rangle \in q^{\mathcal{I},\mathcal{D}} \textit{ iff } \langle c_1, \ldots, c_n \rangle \in q^{M_{min}}.$$

In other words, for GAV integration systems, the tuples of constants in the certain answer to a query q are equal to those that satisfy q in the minimal model of the corresponding logic program.

Let us turn to LAV integration systems. If $\mathcal{I}$ is a LAV integration system, we can define an associated logic program $\mathcal{P}_\mathcal{I}$ by introducing rules for inclusion dependency as before, and by treating queries in the mapping as done in [5]. In particular, given the following query associated to source s (we assume s to be a unary relation and the relations in the query to be binary for simplicity):

$$\rho(s) \;=\; \{\langle X \rangle \mid g_1(X, Y_1) \wedge \cdots \wedge g_k(X, Y_k)\}$$

we can apply skolemization and get

$$\rho(s) \;=\; \{\langle X \rangle \mid g_1(X, f_1(X)) \wedge \cdots \wedge g_k(X, f_k(X))\}.$$

Then, to the skolemized query then we can associate the following rules:

$$\begin{array}{rcl} g_1(X, f_1(X)) & \leftarrow & s(X) \\ & \cdots & \\ g_k(X, f_k(X)) & \leftarrow & s(X) \end{array}$$

As a result, also for LAV integration systems, we can prove the lemma analogous to Lemma 1.

Lemma 2. *Let $\mathcal{I}$ be a LAV integration system, $\mathcal{D}$ a source database for $\mathcal{I}$, $\mathcal{P}_{\mathcal{I}}$ the corresponding logic program defined above, and M_{min} the minimal model of $\mathcal{P}_I \cup \mathcal{D}$. Then given a query q over $\mathcal{G}$, for every tuple $\langle c_1, \ldots, c_n \rangle$ in Γ^n:*

$$\langle c_1, \ldots, c_n \rangle \in q^{\mathcal{I},\mathcal{D}} \text{ iff } \langle c_1, \ldots, c_n \rangle \in q^{M_{min}}.$$

In other words, also for LAV integration systems, the tuples of constants in the certain answer to a query q are equal to those that satisfy q in the minimal model of the corresponding logic program.

With these lemmas in place we can prove our main result.

Theorem 1. *Let $\mathcal{I}$ be a LAV integration system, and $\mathcal{I}'$ the correspondent GAV integration system defined as above. Then $\mathcal{I}'$ is query-equivalent to $\mathcal{I}$.*

Proof (sketch). Let $\mathcal{P}_{\mathcal{I}}$ be the logic program capturing $\mathcal{I}$ and $\mathcal{P}_{\mathcal{I}'}$ the logic program capturing $\mathcal{I}'$. Then it is possible to show that for every source database $\mathcal{D}$ for $\mathcal{I}$ (which is, by definition, also a source database for $\mathcal{I}'$), and every global relation g of the global schema $\mathcal{G}$ of $\mathcal{I}$ we have (modulo renaming of the Skolem functions):

$$\mathsf{g}^{M_{min}} = \mathsf{g}^{M'_{min}}$$

where M_{min} and M'_{min} are the minimal model of $\mathcal{P}_{\mathcal{I}} \cup \mathcal{D}$ and of $\mathcal{P}_{\mathcal{I}'} \cup \mathcal{D}$, respectively. Hence, by considering Lemma 1 and Lemma 2, we get the thesis. □

4 Conclusions

We have studied the relative expressive power of the two main approaches to data integration, namely, the local-as-view and the global-as-view approaches. The question addressed in this paper is whether we can transform a data integration system built with the local-as-view approach into an equivalent system following the global-as-view approach. We have shown that the answer to this question is positive: given a local-as-view system L, we can indeed transform it into a global-as-view system G such that, for each query q, the set of answers to q wrt L is the same as the set of answers to q wrt G.

The result holds in the setting where the global schema is expressed in the relational model with inclusion dependencies, and the queries used in the integration systems (both the queries on the global schema, and the queries in the mapping) are expressed in the language of conjunctive queries. Note that the inclusion dependencies in the global schema play an important role in the transformation. Indeed, it can be shown that, in the case of global schema without dependencies, the result does not hold any more.

It would be interesting to check whether the transformation from LAV to GAV is feasible in the case of different data models for expressing the global schema, and different query languages. Another interesting question is whether there exists an equivalent transformation in the reverse direction, i.e., from the global-as-view to the local-as-view approach. This will the subject of our future investigations.

Acknowledgments: This paper was supported by a MURST (Italian Ministry of University and Scientific Research) grant under the project "D2I". We would like to thank Diego Calvanese and Domenico Lembo for their valuable suggestions on the subject of this paper.

References

1. Serge Abiteboul and Oliver Duschka. Complexity of answering queries using materialized views. In *Proc. of the 17th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'98)*, pages 254–265, 1998.
2. Andrea Calì, Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. On the expressive power of data integration systems, 2002. Submitted for pubblication.
3. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. Answering regular path queries using views. In *Proc. of the 16th IEEE Int. Conf. on Data Engineering (ICDE 2000)*, pages 389–398, 2000.
4. Diego Calvanese, Giuseppe De Giacomo, Maurizio Lenzerini, and Moshe Y. Vardi. View-based query processing and constraint satisfaction. In *Proc. of the 15th IEEE Symp. on Logic in Computer Science (LICS 2000)*, pages 361–371, 2000.
5. Oliver M. Duschka and Michael R. Genesereth. Answering recursive queries using views. In *Proc. of the 16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'97)*, pages 109–116, 1997.
6. Gösta Grahne and Alberto O. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proc. of the 7th Int. Conf. on Database Theory (ICDT'99)*, volume 1540 of *Lecture Notes in Computer Science*, pages 332–347. Springer-Verlag, 1999.
7. Jarek Gryz. Query folding with inclusion dependencies. In *Proc. of the 14th IEEE Int. Conf. on Data Engineering (ICDE'98)*, pages 126–133, 1998.
8. Richard Hull. Managing semantic heterogeneity in databases: A theoretical perspective. In *Proc. of the 16th ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS'97)*, 1997.
9. Alon Y. Levy. Answering queries using views: A survey. Technical report, University of Washinghton, 1999.
10. Alon Y. Levy. Logic-based techniques in data integration. In Jack Minker, editor, *Logic Based Artificial Intelligence*. Kluwer Academic Publisher, 2000.
11. Chen Li and Edward Chang. Query planning with limited source capabilities. In *Proc. of the 16th IEEE Int. Conf. on Data Engineering (ICDE 2000)*, pages 401–412, 2000.
12. John W. Lloyd. *Foundations of Logic Programming (Second, Extended Edition)*. Springer-Verlag, Berlin, Heidelberg, 1987.
13. Xiaolei Qian. Query folding. In *Proc. of the 12th IEEE Int. Conf. on Data Engineering (ICDE'96)*, pages 48–55, 1996.
14. Jeffrey D. Ullman. Information integration using logical views. In *Proc. of the 6th Int. Conf. on Database Theory (ICDT'97)*, volume 1186 of *Lecture Notes in Computer Science*, pages 19–40. Springer-Verlag, 1997.
15. Ron van der Meyden. Logical approaches to incomplete information. In Jan Chomicki and Günter Saake, editors, *Logics for Databases and Information Systems*, pages 307–356. Kluwer Academic Publisher, 1998.