Data Integration Using ID-Logic

Bert Van Nuffelen[1], Alvaro Cortés-Calabuig[1], Marc Denecker[1], Ofer Arieli[2], and Maurice Bruynooghe[1]

[1] Department of Computer Science, Katholieke Universiteit Leuven, Belgium
{Bert.VanNuffelen, Alvaro.Cortes, Marc.Denecker, Maurice.Bruynooghe}@cs.kuleuven.ac.be
[2] Department of Computer Science, The Academic College of Tel-Aviv, Israel
oarieli@mta.ac.il

Abstract. ID-Logic is a knowledge representation language that extends first-order logic with non-monotone inductive definitions. This paper introduces an ID-Logic based framework for database schema integration. It allows us to to uniformly represent and reason with independent source databases that contain information about a common domain, but may have different schemas. The ID-Logic theories that are obtained are called *mediator-based systems*. We show that these theories properly capture the common methods for data integration (i.e., global-as view and local-as-view with either exact or partial definitions), and apply on them a robust abductive inference technique for query answering.

1 Introduction

This work introduces a query answering system that mediates among several independent databases (sources) that contain information about a common domain, but where each source may have a different schema. These systems, called *information integration systems* (or *mediator-based systems*, see [24, 25, 31]) consist of an alphabet, called the *global schema* (representing the global information), and a structure that links the information of the sources with the global schema, such that a virtual knowledge-base in terms of the global schema is obtained. Each schema of a source, as well as the global schema, reflects its own view on the information domain. Therefore, the "intended meaning" of the different schemas should be related, and this is done by a logic theory. This requires appropriate definitions, in which the relations of one schema are expressed in terms of another. There are two common methods to define these relations: one, called *Global-as-View* (GAV) [31], expresses the relations of the global schema in terms of those of the sources. The other, called *Local-as-View* (LAV) [25], defines each source relation in terms of the relations of the global schema.

Both approaches have to deal with gaps between the amount of knowledge in the sources and in the intended global database. Even if a certain source contains a complete knowledge about its own relations, it might possess only partial information about the global relations, which means (in the LAV approach) incomplete definitions. It might happen, however, that the partial information is

complemented by other sources, so that together the sources possess complete knowledge about (part of) the intended global database. The possible relations between the amount of (extentional) knowledge contained in a source predicate and that of the (intended) global database predicates are designated in [14, 22] by different labels: a source predicate that contains a proper subset (respectively, superset) of the intended knowledge of the corresponding predicate at the global schema is labeled *open* (respectively, *closed*). A source predicate whose information is the same as that of the global one is labeled *clopen.*

In addition to its (global) schema and the mapping to the sources' schemas, a mediator-based system also defines an inference procedure that exploits the represented information in order to answer queries. Different semantics [1] have been used for query answers (in terms of the global schema): (1) According to the *certain answer* semantics a tuple $[t]$ is an answer to a global query $\mathcal{Q}$ if its true in all possible instances of the global schema. (2) According to the *possible answer* semantics a tuple $[t]$ is an answer of a query $\mathcal{Q}$ if $[t]$ holds in at least one instance of the global schema. Typically, answers are computed in two phases: first, the query is rewritten in a query in terms of the source relations. Next, the sources are queried.This is needed because the sources are not materialized at the global level. Among query answering algorithms are Bucket [25], Minicon [29] and Inverse-rule [20].

We present a mediator-based system that is based on ID-Logic [15, 16], an expressive knowledge representation language. Contrary to most systems, ours explicitly distinguishes between complete and incomplete knowledge. It also allows us to formalize the knowledge expressed by the labels of [14, 22] and offers a uniform treatment of the LAV and GAV approaches (as well as a mixture of both). Query answering is implemented by abductive reasoning [19] that simplifies the construction of the transformed queries and provides informative answers when sources are temporary unavailable.

2 Preliminaries

2.1 Mediator-based systems

A *source database* is a structure $\langle \mathcal{L}, I \rangle$ where $\mathcal{L}$ is a first-order language and I is a database instance, *i.e.*, a set of tuples representing all true instances of the predicates of $\mathcal{L}$. It is assumed that the unique names axioms hold.

Example 1. A source describing a set of students may have the following structure: $\mathcal{S}_1 = \langle \{student(\cdot)\}, \{student(john), student(mary)\} \rangle$

Definition 1 (A mediator-based system). *A* mediator-based system $\mathfrak{G}$ *is a triple* $\langle \mathcal{L}, S, M \rangle$*, where* $\mathcal{L}$ *is a first-order language of the integrated (or global) database,* $S = \{\mathcal{S}_1, \ldots, \mathcal{S}_n\}$ *is a set of source databases, and* M *is a set of sets of formulae in* $\mathcal{L}$ *(representing the relationships between the sources and the intended global database).*

To simplify the presentation, we assume that each predicate is uniquely defined by either a source or the global schema[3]. We also assume the unique domain assumption: all involved database languages share the same domain, *i.e.*, all constants and function symbols are shared and have the same interpretation everywhere. Hence, as in the above example, the language of a database is completely determined by its *vocabulary*, i.e., its set of predicates.

Example 2. Consider the following two data-sources:
$\mathcal{S}_1 = \langle \{student(\cdot)\}, \{student(john), student(mary)\}\rangle$,
$\mathcal{S}_2 = \langle \{enrolled(\cdot,\cdot)\}, \{enrolled(john, 1999), enrolled(mary, 2000)\}\rangle$.
A possible mediator-based system $\mathfrak{G}$ for these sources is $\langle \mathcal{L}, S, M\rangle$, where
$\mathcal{L} = \{st99(\cdot)\}$, $S = \{\mathcal{S}_1, \mathcal{S}_2\}$, and
$M = \{\{\forall\, x.st99(x) \leftarrow student(x) \wedge enrolled(x, 1999)\}\}$[4]
Queries w.r.t. $\mathfrak{G}$ will be first-order formulas over $\mathcal{L}$ (*e.g.* $\exists\, x.st99(x)$).

2.2 ID-Logic

ID-Logic [15–18] is a knowledge representation language that extends classical first-order logic with non-monotone inductive definitions. Formally:

Definition 2 (ID-Logic). *An* ID-Logic theory $\mathcal{T}$ *based on the first-order logic language $\mathcal{L}$ is a pair ($\mathcal{D}$,$\mathcal{F}$). $\mathcal{D}$ is a set of definitions D_i $(i = 1 \ldots n)$ and $\mathcal{F}$ is a set of first-order formulas. A* definition D *is a set of rules of the form $p(\overline{t}) \leftarrow B$ where $p(\overline{t})$ is an atom and B is any first-order formula.*

The predicates occurring in the heads of the rules of an inductive definition D are the *defined* predicates of D. All the other predicates belong to $Open(D)$, the set of *open* predicates of D. As will be exploited in Section 3, the same predicate can be defined in different definitions. Care must be taken that different definitions are equivalent.

Definition 3 (A model of a definition[5]). *A structure M is a* model *of a definition D iff there exists an interpretation I of $Open(D)$ such that M is the two-valued well-founded [33] model of D that extends I. A structure M is a model of $\mathcal{D}$ iff M is a model of each $D \in \mathcal{D}$.*

Definition 4 (Formal semantics of an ID-Logic theory). *A structure M is a model of the* ID*-Logic theory $\mathcal{T}$=($\mathcal{D}$,$\mathcal{F}$) iff M is a model of $\mathcal{D}$ and satisfies all formulas of $\mathcal{F}$. The collection of all models of $\mathcal{T}$ is denoted by $Mod(\mathcal{T})$.*

Note 1 (Relation with Description Logics (DL)). Several data integration systems use DL [6] as underlying language (see, *e.g.*, [13, 14]). As shown in [32], ID-Logic can be viewed as a very expressive DL. The main differences between both are the facility to specify *inductive definitions*[6], and the computational

[3] If needed, a simple renaming of predicates in sources can establish this property.
[4] M expresses that $st99(x)$ is the conjunction of two relations. See Section 2.2.
[5] This is also well defined for general non-monotone inductive definitions [16].
[6] Most DL only allow transitive closure as a special case of an inductive definition.

paradigm; whereas DL systems focus on deductive query answering, ID-Logic systems also make use of abductive reasoning, *i.e.*, computing (classes of) models (explanations) that support the query.

Definition 5 (Composition of ID-Logic theories). *For two* ID*-Logic theories $\mathcal{T}_1$ and $\mathcal{T}_2$ over the same language $\mathcal{L}$, the composed theory $\mathcal{T}_1 \circ \mathcal{T}_2$ is an* ID*-Logic theory $\mathcal{T}$ over $\mathcal{L}$, obtained via the pairwise union of both theories:*

$$\mathcal{T} = \mathcal{T}_1 \circ \mathcal{T}_2 = (\mathcal{D}_1, \mathcal{F}_1) \circ (\mathcal{D}_2, \mathcal{F}_2) = (\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{F}_1 \cup \mathcal{F}_2).$$

Proposition 1. *For two* ID*-Logic theories $\mathcal{T}_1$ and $\mathcal{T}_2$ over a language $\mathcal{L}$, it holds that $Mod(\mathcal{T}_1 \circ \mathcal{T}_2) = Mod(\mathcal{T}_1) \cap Mod(\mathcal{T}_2)$.*

2.3 Expressing partial knowledge

When designing mediator-based systems, the available information is often insufficient to define the ontological relations between the global database and the sources. We explain how open predicates can complete partial knowledge.

An incomplete set of rules. Suppose that the set of rules $\{p(\overline{t}) \leftarrow B_i | i = 1 \dots k\}$ only partially defines p. A complete definition can be obtained by adding a rule $p(\overline{s}) \leftarrow p^*(\overline{s})$, in which the auxiliary open predicate p^* represents all the tuples in p that are not defined by any of the bodies B_i. To ensure that the tuples in p^* do not overlap with the other tuples, the integrity constraint $\forall\ p^*(\overline{s}) \rightarrow \neg(B_1 \vee \dots \vee B_k)$ can be added.

An imprecise rule. Another type of incompleteness occurs when the body of a rule $p(\overline{t}) \leftarrow B$ is overly general, *i.e.*, includes tuples not intended to be in the relation p. This can be repaired by adding to the body an auxiliary open predicate p^s that filters the extraneous tuples. The completed rule in this case is $p(\overline{t}) \leftarrow B \wedge p^s(\overline{t})$.

Example 3. Let $st99(\cdot)$ be defined in terms of $student(\cdot)$. The rule $st99(x) \leftarrow student(x)$ is overly general since not all students did enroll in 1999. By adding an auxiliary predicate $st99^s(\cdot)$, denoting all persons enrolled during 1999, the revised rule $st99(x) \leftarrow student(x) \wedge st99^s(x)$ correctly defines $st99(\cdot)$.

3 An ID-Logic Mediator-based System

Definition 6 (An ID-Logic mediator-based system). *For a set of sources $\{\mathcal{S}_1, \dots, \mathcal{S}_n\}$ and a global schema $\mathcal{L}_G$, an* ID-Logic mediator-based system *is a triple $\mathfrak{G} = \langle \mathcal{L}_G, S, M \rangle$, where*

- *S is a set of* ID*-Logic theories $\{\mathcal{S}_1 \dots \mathcal{S}_n\}$ encoding the source databases.*
- *M is a set of* ID*-Logic theories $\{\mathcal{W}_1 \dots \mathcal{W}_n, \mathcal{K}\}$ encoding the relationships between the sources and the intended global database:*
  - *$\mathcal{W}_i$, $i = 1, \dots, n$, are source mappings for the source $\mathcal{S}_i$ w.r.t. $\mathcal{L}_G$,*
  - *$\mathcal{K}$ is an* ID*-Logic theory that describes how the information in the different sources complement each other.*

and the knowledge of the ID*-Logic mediator-based system $\mathfrak{G}$ is represented by the* ID*-Logic theory $\mathcal{T} = S_1 \circ \dots \circ S_n \circ \mathcal{W}_1 \circ \dots \circ \mathcal{W}_n \circ \mathcal{K}$.*

The sources. A source $\mathcal{S}_i = \langle \mathcal{L}_\mathcal{S}, I \rangle$ is encoded as $S_i = (\{\{\mathcal{I}\}\}, \emptyset)$ where $\mathcal{I}$ is obtained by interpreting the database instance I as an enumeration of facts.

Example 4. The source $\mathcal{S} = \langle \{student(\cdot)\}, \{student(john), student(mary)\} \rangle$ of Example 1 is interpreted as the following ID-Logic theory:

$$S = \big(\{\{\, student(john).\ \ student(mary).\,\}\}, \emptyset\big)$$

Relating one source with the global schema. This part defines the relationships between the relations of a source and of the global database. These relationships are expressed in the form of (inductive) definitions, taking into account the ontological relationships between the predicates and the actual knowledge of the source. The techniques described in Section 2.3 are used when there is a mismatch between the information in the source and in the global database.

Definition 7 (A source mapping). *A source mapping from a language $\mathcal{L}_2$ to a language $\mathcal{L}_1$ is an* ID*-Logic theory $\mathcal{W}$ defining the predicates of $\mathcal{L}_1$ in terms of the predicates of $\mathcal{L}_2$ and of the necessary auxiliary open predicates.*

Local-as-View (LAV) and Global-as-View (GAV) are particular instances of source mappings. For a source with vocabulary $\mathcal{L}_S$ and a global database with vocabulary $\mathcal{L}_G$, LAV (GAV) defines the predicates of $\mathcal{L}_S$ ($\mathcal{L}_G$) in terms of the predicates of $\mathcal{L}_G$ ($\mathcal{L}_S$).

Example 5. Consider the languages $\mathcal{L}_1 = \{st99(\cdot)\}$ and $\mathcal{L}_2 = \{student(\cdot)\}$, where $st99(\cdot)$ represents the students enrolled in 1999 and $student(\cdot)$ represents all the students. The possible source mappings are the following:

1. $\mathcal{W}_{1\to 2} = (\{\{st99(x) \leftarrow student(x) \wedge student^s(x).\}\}, \emptyset)$
2. $\mathcal{W}_{2\to 1} =$
   $(\{\{student(x) \leftarrow st99(x) \vee st99^*(x).\}\}, \{\forall\, x.st99^*(x) \rightarrow \neg st99(x).\})$

The meaning of the predicates allows only those two representations. When $\mathcal{L}_1$ is the source predicate, the first mapping is LAV and the second is GAV. Now, the auxiliary predicate $st99^*(\cdot)$ represents the students that are not known by the source, while $student^s(\cdot)$ represents the students known by the source.

Note 2 (GAV or LAV?). According to our definition GAV and LAV are equally good approaches.[7] However, as it is more natural to define abstract concepts in terms of more detailed notions, differences in abstraction levels of the source and the global languages imply that in practice one approach could be more appropriate than the other. Moreover, since the abstraction levels of the sources' languages may also be different (some of which may be more abstract than the global language and some may be less abstract than it), it makes sense to

[7] In the literature one finds arguments in favor of one or the other. For our representational point of view there is no difference. However, in the section on query answering, we give an argument in favor of GAV.

combine both approaches, i.e., to use GAV for the source mappings between certain sources and the global schema, and the LAV approach for the mappings between the other sources and the global schema. The fact that our framework supports such a combination may serve, therefore, as one of its advantages over other formalisms.

As noted in the introduction, special labels are sometimes used to denote the amount of knowledge a source stores (see, e.g., [14, 22]). For the sake of presentation, we show this in the context of LAV mappings between one source relation and one global relation. In general, the labels express a relation between a query over the source and over the global schema [14]. The following example illustrates that our use of open auxiliary predicates exactly captures the meaning of such labels. It is a variant on the world cup example, considered in [22].

closed source: The source predicate contains **more** information than the mediator predicate needs.
Consider $\mathcal{L}_G = \{st99(\cdot)\}$ and $\mathcal{L}_{\mathcal{S}} = \{student(\cdot)\}$. Here, the mapping is:

$$\left(\begin{array}{l} \{\{\ student(x) \leftarrow st99(x) \vee student^*(x).\ \}\}, \\ \{\ \forall\ x.student^*(x) \rightarrow \neg st99(x).\ \} \end{array} \right)$$

$student^*(\cdot)$ models that there are other students than those listed by $st99(\cdot)$.

open source: The source predicate contains **less** information than the mediator predicate needs. Consider $\mathcal{L}_G = \{st99(\cdot)\}$ and $\mathcal{L}_{\mathcal{S}} = \{st99male(\cdot)\}$. Now, the mapping is:

$$\left(\{\{\ st99male(x) \leftarrow st99(x) \wedge st99male^s(x).\ \}\}, \emptyset\right)$$

$st99male^s(x)$ models the unknown subset of male students.

clopen source: The source predicate has **exact** information for the mediator predicate. Consider $\mathcal{L}_G = \{st99(\cdot)\}$ and $\mathcal{L}_{\mathcal{S}} = \{studentsOf1999(\cdot)\}$. The mapping is as follows:

$$\left(\{\{\ studentsOf1999(x) \leftarrow st99(x).\ \}\}, \emptyset\right)$$

Knowing what multiple sources know. The last component in the composition of the ID-Logic theories, introduced in Definition 6 for representing a mediator-based system, contains an ID-Logic theory that allows the designer to formulate additional meta-knowledge about how partial information of one source (regarding a certain predicate of the global schema) is completed by data of other sources. This ID-Logic theory is denoted by $\mathcal{K}$, and as shown below, its information may be vital for a proper schema integration.

Example 6. Consider the global schema $\{student(\cdot)\}$ and the sources
$\mathcal{S}_1 = \langle\{st99(\cdot)\}, \{st99(john)\}\rangle$ and $\mathcal{S}_2 = \langle\{st00(\cdot)\}, \{st00(mary)\}\rangle$ having the source mappings

$$\mathcal{W}_{1\rightarrow G} = \left(\{\{student(x) \leftarrow st99(x) \vee st99^*(x).\}\}, \{\forall\ x.st99^*(x) \rightarrow \neg st99^(x).\}\right)$$
$$\mathcal{W}_{2\rightarrow G} = \left(\{\{student(x) \leftarrow st00(x) \vee st00^*(x).\}\}, \{\forall\ x.st00^*(x) \rightarrow \neg st00^(x).\}\right)$$

Note that $\mathcal{W}_{1\to G} \circ \mathcal{W}_{2\to G}$ contains two (alternative and equivalent) definitions for the student relation. The statement that the relation $student(\cdot)$ is complete w.r.t. the set of sources $\{\mathcal{S}_1, \mathcal{S}_2\}$ can be formalized by the first-order assertion

$$\mathcal{K} = (\emptyset, \{\forall\ x.\neg(st99^*(x) \wedge st00^*(x)).\})$$

Obviously, no general rules for expressing meta-knowledge exist. It depends on the representation choices of the source mappings, the information content of the sources and the intended information content of the global database.

An elaborated example. We conclude this section with an elaboration of Example 2. It shows, in particular, that a certain data integration problem can be described by many different mediator-based systems.

Example 7. Consider two sources, each one has a complete knowledge about its relations. Source $\mathcal{S}_1$ stores all full-time students, and source $\mathcal{S}_2$ contains data about the year of enrollment of all students (both part-time and full-time):

$$\begin{array}{ll} \mathcal{S}_1 = \langle \{student(\cdot)\}, & \{student(john), student(mary), student(bob)\}\rangle, \\ \mathcal{S}_2 = \langle \{enrolled(\cdot,\cdot)\}, & \{enrolled(john, 1999), enrolled(eve, 1999), \\ & enrolled(mary, 2000), enrolled(alice, 2003)\}\rangle \end{array}$$

A mediator-based system that extracts lists of full-time students enrolled at the years 1999 and 2000 looks as follows: $\mathfrak{G} = \langle \mathcal{L}_G, \{S_1, S_2\}, M\rangle$, where:

- $\mathcal{L}_G = \{st99(\cdot), st00(\cdot)\}$
- $S_1 = \left(\left\{ \left\{ \begin{array}{l} student(john). \\ student(mary). \\ student(bob). \end{array} \right\} \right\}, \emptyset \right)$
- $S_2 = \left(\left\{ \left\{ \begin{array}{l} enrolled(john, 1999). \\ enrolled(eve, 1999). \\ enrolled(mary, 2000). \\ enrolled(alice, 2003). \end{array} \right\} \right\}, \emptyset \right)$

Three possible mappings are presented below. In these mapping, we have additionally assumed that the sources contain all information about the global relations.

- The GAV approach where each source is individually related with the global schema. Here, $M = \{\mathcal{W}_{G\to 1}, \mathcal{W}_{G\to 2}, \mathcal{K}\}$, where

$$\mathcal{W}_{G\to 1} = \left(\left\{ \left\{ \begin{array}{l} st99(x) \leftarrow student(x) \wedge st99^s_{\mathcal{S}_1}(x). \\ st00(x) \leftarrow student(x) \wedge st00^s_{\mathcal{S}_1}(x). \end{array} \right\} \right\}, \emptyset \right)$$

$$\mathcal{W}_{G\to 2} = \left(\left\{ \left\{ \begin{array}{l} st99(x) \leftarrow enrolled(x, 1999) \wedge st99^s_{\mathcal{S}_2}(x). \\ st00(x) \leftarrow enrolled(x, 2000) \wedge st00^s_{\mathcal{S}_2}(x). \end{array} \right\} \right\}, \emptyset \right)$$

$$\mathcal{K} = \left(\emptyset, \left\{ \begin{array}{l} \forall\ x. st99^s_{\mathcal{S}_1}(x) \leftrightarrow enrolled(x, 1999). \\ \forall\ x. st00^s_{\mathcal{S}_1}(x) \leftrightarrow enrolled(x, 2000). \\ \forall\ x. st99^s_{\mathcal{S}_2}(x) \leftrightarrow student(x). \\ \forall\ x. st00^s_{\mathcal{S}_2}(x) \leftrightarrow student(x). \end{array} \right\} \right)$$

– An alternative GAV approach that treats the two sources as if there are one. This time, $M = \{\mathcal{W}_{G\rightarrow\{1,2\}}\}$, where:

$$\mathcal{W}_{G\rightarrow\{1,2\}} = \left(\left\{ \left\{ \begin{array}{l} st99(x) \leftarrow student(x) \wedge enrolled(x, 1999). \\ st00(x) \leftarrow student(x) \wedge enrolled(x, 2000). \end{array} \right\} \right\}, \emptyset \right)$$

– The LAV approach: $M = \{\mathcal{W}_{1\rightarrow G}, \mathcal{W}_{2\rightarrow G}, \mathcal{K}\}$.

$$\mathcal{W}_{1\rightarrow G} = \left(\begin{array}{l} \{\{ student(x) \leftarrow st99(x) \vee st00(x) \vee student^*(x). \}\}, \\ \{\forall\ x.student^*(x) \rightarrow \neg(st99(x) \vee st00(x)).\} \end{array} \right)$$

$$\mathcal{W}_{2\rightarrow G} = \left(\begin{array}{l} \left\{ \left\{ \begin{array}{l} enrolled(x, y) \leftarrow st99(x) \wedge y = 1999. \\ enrolled(x, y) \leftarrow st00(x) \wedge y = 2000. \\ enrolled(x, y) \leftarrow enrolled^*(x, y) \wedge (y \neq 1999 \vee y \neq 2000). \end{array} \right\} \right\}, \\ \qquad \left\{ \begin{array}{l} \forall\ x, y.enrolled^*(x, y) \rightarrow \neg(st99(x) \wedge y = 1999). \\ \forall\ x, y.enrolled^*(x, y) \rightarrow \neg(st00(x) \wedge y = 2000). \end{array} \right\} \end{array} \right)$$

$$\mathcal{K} \quad = \left(\emptyset, \left\{ \begin{array}{l} \forall\ x.st99(x) \rightarrow student(x). \\ \forall\ x.st00(x) \rightarrow student(x). \\ \forall\ x.st99(x) \rightarrow enrolled(x, 1999). \\ \forall\ x.st00(x) \rightarrow enrolled(x, 2000). \end{array} \right\} \right)$$

According to any one of the representations above, the unique model of $\mathfrak{G}$ (restricted to $\mathcal{L}_G$) is $\{st99(john), st00(mary)\}$.

4 Query Answering

In the previous sections we have shown how to set up correctly an ID-Logic mediator-based system. This section discusses how queries can be answered with respect to such system. First, we consider the general context of ID-Logic theories and then concentrate on abductive inference as a general technique of computing answers posed to mediator-based systems. We also show that the answers generated by the abductive process are more informative than answers that are produced by other techniques.

Definition 8 (Types of queries). *Let $\mathcal{T}$ be an* ID*-Logic theory and $\mathcal{Q}$ a query.*

a) $\mathcal{Q}$ is skeptically true *iff it is entailed by every model of $\mathcal{T}$; i.e.,*

$$\mathcal{T} \models_{skep} \mathcal{Q} \text{ iff } \text{for each model } M \text{ of } \mathcal{T} : M \models \mathcal{Q}.$$

b) $\mathcal{Q}$ is credulously true *iff it is entailed by at least one model of $\mathcal{T}$; i.e.,*

$$\mathcal{T} \models_{cred} \mathcal{Q} \text{ iff } \text{there exists a model } M \text{ of } \mathcal{T} : M \models \mathcal{Q}.$$

In a mediator-based system $\mathfrak{G}$, the sources contain fixed information. Thus, answers to queries that are supported by the sources will be always skeptically true, while answers to queries for which the sources have no information might be either skeptically false or credulously true.

As mediator-based system does not materialize the knowledge of the sources in its own schema, the process of answering a global query $\mathcal{Q}$ is two-phased:

a) Compute for $\mathcal{Q}$ an equivalent (if possible) query $\mathcal{Q}_s$ expressed in terms of the source languages.
b) Query the sources with $\mathcal{Q}_s$.

Abductive Inference. Abductive reasoning is related to credulous query answering, i.e., finding a model that satisfies the query. Recall that each model of an ID-Logic theory is uniquely determined by an interpretation of the open predicates of the theory. Abductive reasoning is the inference process that constructs an explanation formula $\mathcal{E}$ in terms of the open predicates that entails the query $\mathcal{Q}$. Formally, for an ID-Logic theory $\mathcal{T}$ and a query $\mathcal{Q}$, $\mathcal{E}$ is an *abductive solution* iff $\exists\ (\mathcal{E})$ is satisfiable w.r.t. $\mathcal{T}$ and $\mathcal{T} \models \forall\ (\mathcal{E} \rightarrow \mathcal{Q})$.

Abductive reasoning for ID-Logic theories is provided by the $\mathcal{A}$system [3, 5, 23]. A preprocessing step transforms an ID-Logic theory into an equivalent ID-Logic theory consisting of one definition. Such theories correspond to abductive normal logic programs that form the input for the $\mathcal{A}$system. The open predicates of this ID-Logic theory are mapped into the abducibles in the abductive logic programming framework. We refer the reader to [19] for more details about abduction and its relation with ID-Logic.

In the case of the $\mathcal{A}$system the computed explanation formula $\mathcal{E}$ describes a class of models of $\mathcal{T}$. When $\mathcal{E}$ is **true**, the query is satisfiable w.r.t. to all models. When the $\mathcal{A}$system is unable to find an abductive solution for $\mathcal{Q}$, then $\mathcal{T} \models \forall\ (\neg\mathcal{Q})$[8].

Answers for Queries. Consider $\mathfrak{G} = \langle \mathcal{L}_G, \{\mathcal{S}_1, \ldots, \mathcal{S}_n\}, \{\mathcal{W}_1, \ldots, \mathcal{W}_l, \mathcal{K}\}\rangle$, a mediator-based system, and $\mathcal{T}_q = \mathcal{W}_1 \circ \cdots \circ \mathcal{W}_l \circ \mathcal{K}$, the derived ID-Logic theory. This theory describes only the relationship between the languages. Then a new query $\mathcal{Q}_s$ expressed in terms of $\mathcal{L}_G$ is *derivable* from an abductive solution for the query $\mathcal{Q}$ w.r.t. $\mathcal{T}_q$. According to the mapping style, the abductive solution forms the new query $\mathcal{Q}_s$ or the basis to compute it. In GAV the open predicates are the source predicates, and thus, an abductive solution $\mathcal{E}$ is an expression in terms of the source predicates. $\mathcal{E}$ is then a representation of $\mathcal{Q}_s$. The LAV case is different: the open predicates are those of the global schema $\mathcal{L}_G$. An abductive solution $\mathcal{E}$ does not encode directly $\mathcal{Q}_s$. However, all models satisfying $\mathcal{E}$ correspond to answers for $\mathcal{Q}_s$. Hence one has to design an extra procedure to compute answers form $\mathcal{Q}_s$ out of the abductive solution for LAV mappings. In the literature one finds approaches that can form the basis for that procedure. For example, the inverse-rule algorithm for Datalog [20]. The availability of a general computational engine (an abductive solver) for GAV and the absence of such one for LAV, is in our opinion an argument in favor of GAV.

Example 8. Consider $\mathfrak{G} = \langle \{student(\cdot)\}, S, \{\mathcal{W}_{G\rightarrow\{1,2\}}\}\rangle$, a mediator-based system in which

$$S = \{\ \mathcal{S}_1 = \langle \{st99(\cdot)\}, \{st99(john)\}\rangle,\ \mathcal{S}_2 = \langle \{st00(\cdot)\}, \{st00(mary)\}\rangle\ \},$$

[8] This is called the *duality property* of abductive systems.

$\mathcal{W}_{G\to\{1,2\}} = (\{student(x) \leftarrow st99(x) \vee st00(x) \vee student^*(x)\}, \emptyset)$.

Now suppose we pose the global query $\mathcal{Q} : \exists\ student(john)$ to $\mathfrak{G}$. In order to answer this query with the data from the sources, an abductive explanation for $\mathcal{Q}$ w.r.t. $\mathcal{T}_q = \mathcal{W}_{G\to\{1,2\}}$ is computed by the $\mathcal{A}$system:

$$\mathcal{Q}_s : st99(john) \vee st00(john) \vee student^*(john).$$

This explanation is exactly the expected reformulated query $\mathcal{Q}_s$. Because $\mathcal{Q}_s$ is expressed in terms of the sources and the auxiliary predicates, we can evaluate this query over the sources. When the first two predicates fail, the last one always succeeds, denoting the fact that the sources lack the knowledge to decide whether John is a student. In that case the answer is credulously true. In our case, since $\mathcal{S}_1$ contains the information that John is enrolled in 1999, it follows that he is a student. This is a skeptically true answer.

Supporting dynamics of a mediator-based system. Abductive inference is particularly useful when the mediator-based system acts in a dynamic environment, i.e., the sources are dynamically added or removed. In such a situation, the produced abductive answer contains certain information that justifies the result, and helps to understand it.

Consider, for example, the following scenario: in the university restaurant one gets only a student reduction if he or she is registered in the university database as a student. When the source that contains all part-time students falls out, it might be that none of these students can get its reduction. If the mediator-based system removes each piece of knowledge from the unavailable source and it notifies the restaurant only that one of its sources is down, the restaurant is unable to grant the student reduction to all part-time students. Only when the restaurant is informed with the precise information that the list of part-time students is unavailable, it can question every person that is not recognized as a student if he or she is a part-time student.

The intended behavior is obtained by a source removal operation. Given a mediator-based system $\langle \mathcal{L}_G, \{\mathcal{S}_1, \ldots, \mathcal{S}_n\}, \{\mathcal{W}_1, \ldots, \mathcal{W}_l, \mathcal{K}\}\rangle$ and a corresponding ID-Logic theory

$$\mathcal{T} = \mathcal{S}_1 \circ \cdots \circ \mathcal{S}_n \circ \mathcal{W}_1 \circ \cdots \circ \mathcal{W}_l \circ \mathcal{K},$$

a removal of a source $\mathcal{S}_k$ $(1 \leq k \leq n)$ yields the following theory:

$$\mathcal{T}' = \mathcal{S}_1 \circ \cdots \circ \mathcal{S}_{k-1} \circ \mathcal{S}_{k+1} \circ \cdots \circ \mathcal{S}_n \circ \mathcal{W}_1 \circ \cdots \circ \mathcal{W}_l \circ \mathcal{K},$$

in which all predicates of $\mathcal{S}_k$ are open predicates.[9] Note that this update has non-monotonic characteristics, and that our framework correctly handles this. When the mediator-based system uses GAV, abductive reasoning can determine precisely the information the restaurant needs to react properly on the fall-out of the part-time student source.

[9] Many mediator-based systems remove all knowledge of $\mathcal{S}_k$, as shown by the first possibility in the restaurant scenario. We can simulate this by replacing the source by the empty source where all predicates are false.

Example 9 (Example 8 continued). If source $\mathcal{S}_1$ drops out, the abductive answer $\{st99(john)\}$ will have no source to be queried. The system can report to the user that in order to answer skeptically the query it is necessary to wait until $\mathcal{S}_1$ is available again, and the information $st99(john)$ can be verified.

Adding a new source is slightly more complex, because the mappings might have to be reconsidered. The amount of work in this case depends on the amount and type of information that the new source contributes. For example, a new source might require that the completeness assertions imposed in $\mathcal{K}$ must be reconsidered. In any data integration system the addition of a source requires this reconsideration. (Except when strong preconditions are imposed on the new sources.) In the worst case each mapping has to be updated. Fortunately, in most practical cases this is unlikely to happen. Moreover the modularity of ID-Logic enforces a strong locality of the changes since the changes must only be applied on the definitions that contain involved predicates.

5 Generalizations

5.1 Lifting the unique domain assumption

Often two sources use different domain elements to denote the same object in the world (e.g., a client number and a social security number for the same person). By introducing an auxiliary mapping, these differences can be taken into account. Let HU($\mathcal{L}$) denote the Herbrand Universe of the language $\mathcal{L}$.

Definition 9. *A mapping between the domains of two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ is the bijection $map_{\mathcal{L}_1 \to \mathcal{L}_2} = \{map(t_1, t_2) | t_1 \in HU(\mathcal{L}_1)$ and $t_2 \in HU(\mathcal{L}_2)\}$. $map_{\mathcal{L}_2 \leftarrow \mathcal{L}_1}$ denotes the inverse of the mapping $map_{\mathcal{L}_1 \to \mathcal{L}_2}$.*[10]

It is sufficient to define a mapping between each source and the global schema. A mapping $map(Name, ID)$ that maps names in identity numbers can be defined by the following theory: $\mathcal{W} = (\{\{st99(x) \to map(x, y) \wedge student(y).\}\}, \emptyset)$.

5.2 Reasoning with inconsistent knowledge

Up to now, we assumed that all information in the sources was consistent w.r.t. the intended global database, and so integrity constraints at the global level were not considered. In case that the global schema does contain such constraints, inconsistencies can arise[11]. We are aware of two approaches for handling this:

Computing repairs [3, 4, 9]: A repair is a set of atoms that must be added or retracted from the knowledge base in order to make it consistent. Repairs may advise the user about changes needed to restore consistency. In our context the repairs are computed at the level of each source.

[10] The mapping is the identity when both languages share the same Herbrand universe.

[11] Not to be confused with the notion of consistency in [22] where the concept is applied to conflicts that can arise when integrating complete sources.

Consistent query answering [2, 8]: This approach avoids the computation of the repairs. It transforms a query such that the answers of the transformed query are consistent w.r.t. to all repaired knowledge bases.

6 Comparison with Related Works

In the previous sections we introduced an ID-Logic mediator-based system for data integration and argued in favor of its expressive power. In this section we discuss how our work is related to some well-known existing formalisms.

GAV, LAV, and their combinations ([25, 29, 31]). A decade of active research in the topic of data integration has resulted in several implemented mediator-based systems. Some of them apply LAV, others are based on GAV (for a review, see [31]). Our ID-Logic framework takes into consideration both approaches, so the data integration system designer can select one or another, or work with both (see Note 2).

Generalized methods for data integration. At least two extensions have been proposed to increase the expressive power of LAV and GAV paradigms.

GLAV approach ([21]). This is an extension of LAV that allows to map a conjunctive query expression ϕ_G over the global schema into a conjunctive query expression ϕ_S over the sources [12]. This variant can be simulated in our framework by the introduction of an auxiliary predicate, say p, which has the view definition $\{p(\overline{t}) \leftarrow \phi_S\}$ w.r.t. the sources (ϕ_S). Using p, a LAV mapping can be constructed by $\{p(\overline{t}) \leftarrow \phi_G\}$.
Both-As-View (BAV) ([28]). McBrien and Poulovassilis present a novel method that combines the advantages LAV and GAV in the presence of dynamic schemas. They show how LAV and GAV view definitions can be fully derived from BAV transformation sequences, and that BAV transformation sequences can be partially derived for LAV or GAV view definitions. We believe that these transformation sequences (or extensions of them) could be applied to translate BAV mapping into ID-Logic mappings.

Data integration by Description Logics ([14]). Calvanese et al. present a general framework for data integration. Tt turns out that our framework encodes in a nice way this framework. In particular, they define labels (similar to [22]) to denote the amount of knowledge a mapping rule contributes to the global database. As shown before, this can be captured by the use of (auxiliary) open predicates.

A remarkable statement in [14] concerns the appropriateness of Description Logics for data integration. Due to the limitations on the use of variables, Description Logics are poor as query languages. Calvanese et al. argue that for the

[12] An equivalent extension for the GAV approach is straightforward.

data integration problem the modeling language has to be good in that respect. Since ID-Logic imposes no restriction on the use of variables, and can be regarded as a very expressive Description Logic, it is not surprising that the use of ID-Logic leads to a general approach, enclosing many of the existing approaches.

Data integration by abductive methods ([10, 11]). The power of abduction for data integration has already been recognized in the COntext INterchange project, where abductive logic programs are used to represent the mappings according to the GAV approach. In the same spirit, our answering procedure is based on abductive reasoning, that may be supported by existing abductive computational tools [3].

Query answering with incomplete knowledge ([1, 22, 26]). The inherent connection between incomplete knowledge and mediator-based systems has been broadly identified in the literature. Applying an expressive knowledge representation language with explicit denotation of incompleteness, a better understanding of the nature of the whole data integration problem is gained. Moreover, we argued how this knowledge can be used by the inference mechanism to compute more informative answers, for example, when a source is dropped.

Relation with composition of knowledge bases The knowledge representation origins of ID-Logic relate the data integration problem considered here with merging of knowledge bases [7, 34]. The former can be viewed as a particular case of the latter, where some knowledge bases contain actual data and others do not.

7 Conclusions and Future Work

This ID-Logic framework supports both the GAV and the LAV approaches to the data integration problem, any combination of these approaches, as well as various generalized methods (such as GLAV and BAV). Our framework provides a general method for expressing the relationships between a global schema and those of the sources. Specifically, the state of knowledge of each source w.r.t. the intended global database can be explicitly represented in the ID-Logic theories themselves. This allows to capture precisely the information of labels in [14, 22], and it clearly shows the strong relation of the data integration problem with incomplete knowledge [12], no matter which mapping approach is taken.

Since ID-Logic may be regarded as an expressive Description Logic [32], our approach generalizes all approaches that use Description Logics provided they take equal assumptions on the problem context.

For the future, we plan to implement the ID-Logic mediator-based system together with inconsistency repairing techniques, and test their behavior in realistic situations.

References

1. S. Abiteboul and O.M. Duschka. Complexity of answering queries using materialized views. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS'98*, pages 591–596, 1998.
2. M. Arenas, L. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Systems, PODS'99*, pages 68–79, 1999.
3. O. Arieli, M. Denecker, B. Van Nuffelen, and M. Bruynooghe. Coherent integration of databases by abductive logic programs. Accepted to the *Journal of Artificial Intelligence Research*, 2004. See http://www.cs.kuleuven.ac.be/ dtai/.
4. O. Arieli, B. Van Nuffelen, M. Denecker, and M. Bruynooghe. Database repair by signed formulae. In *Foundations of Information and Knowledge Systems, FoIKS 2004*, pages 14–30. LNCS 2942, Springer, 2004.
5. The $\mathcal{A}$system. Obtainable via www.cs.kuleuven.ac.be/ dtai/kt/systems-E.shtml.
6. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook. Theory, Implementation and Applications*. Cambridge University Press, 2003.
7. C. Baral, J. Minker, and S. Kraus. Combining multiple knowledge bases. *IEEE Transactions on Knowledge and Data Engineering*, 3(2):208–221, June 1991.
8. L. Bertossi, J. Chomicki, A. Cortes, and C. Gutierrez. Consistent answers from integrated data sources. In *Flexible Query Answering Systems, 5th International Conference, FQAS 2002*, pages 71–85. LNCS 2522, Springer, 2002.
9. L. Bertossi and L. Bravo. Logic programs for consistently querying data integration systems. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI 2003*, 2003.
10. S. Bressan, C. H. Goh, K. Fynn, M. Jakobisiak, K. Hussein, H. Kon, T. Lee, S. Madnick, T. Pena, J. Qu, A. Shum, and M. Siegel. The Context Interchange mediator prototype. In *Proc. of ACM SIGMOD'97 Conf.*, pages 525–527, 1997.
11. S. Bressan, C.H. Goh, T. Lee, S.E. Madnick, and M. Siegel. A procedure for mediation of queries to sources in disparate contexts. In *Proceedings of International Logic Programming Symposium, ILPS'97*, pages 213–227, 1997.
12. A. Calí, D. Calvanese, G. De Giacomo, and M. Lenzerini. Data integration under integrity constraints. In *Int. Conf. on Advanced Information Systems Engineering, CAiSE 2002*, pages 262–279. LNCS 2348, Springer, 2002.
13. D. Calvanese, G. De Giacomo, and M. Lenzerini. Description logics for information integration. In A. Kakas and F. Sadri, editors, *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski*, pages 41–60. LNCS 2408, Springer, 2002.
14. D. Calvanese, G. De Giacomo, and M. Lenzerini. Ontology of integration and integration of ontologies. In *Working Notes of the 2001 International Description Logics Workshop (DL-2001), 2001*, CEUR Workshop Proc. 49, 2001.
15. M. Denecker. Extending classical logic with inductive definitions. In *Computational Logic - CL 2000*, pages 703–717. LNCS 1861, Springer, 2000.
16. M. Denecker, M. Bruynooghe, and V. Marek. Logic programming revisited: logic programs as inductive definitions. *ACM Transactions on Computational Logic*, 2(4):623–654, 2001.
17. M. Denecker, and E. Ternovska. Inductive Situation Calculus. In *Proc. of 9th Internation Conference on Principles of Knowledge Representation and Reasoning*, 2004, accepted.

18. M. Denecker, and E. Ternovska. A Logic of Non-Monotone Inductive Definitions and its Modularity Properties. In *Proc. of 7th International Conference on Logic Programming and Nonmonotonic Reasoning*, pages 47-60, LNCS 2923, Springer, 2004.
19. M. Denecker and A.C. Kakas. Abduction in logic programming. In *Computational Logic: Logic Programming and Beyond, Essays in Honour of Robert A. Kowalski*, pages 402–436. LNCS 2407, Springer, 2002.
20. O. Duschka, M. Genesereth, and A. Levy. Recursive query plans for data integration. In *Journal of Logic Programming*, 1:49–73, 2000.
21. S. Friedman, A. Levy and T Millstein. Navigational plans for data integration. In *Proc. 16th National Conference on AI*, pages 67-73. AAAI Press,1999.
22. G. Grahne and A. Mendelzon. Tableau techniques for querying information sources through global schemas. In *Proceedings of 7th International Conference on Database Theory, ICDT'99*, pages 332–347. LNCS 1540, Springer, 1999.
23. A.C Kakas, B. Van Nuffelen, and M. Denecker. A-system : Problem solving through abduction. In *Proc. of the Seventeenth Int. Joint Conf. on Artificial Intelligence, IJCAI 2001*, pages 591–596. 2001.
24. M. Lenzerini. Data integration: A theoretical perspective. In *Proceedings of the Twenty-first ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, PODS 2002*, pages 233–246. 2002.
25. A. Y. Levy, A. Rajaraman, and J.J. Ordille. Querying heterogeneous information sources using source descriptions. In *Int. Conf. on Very Large Data Bases, VLDB'96*, pages 251–262. Morgan Kaufmann, 1996.
26. A.Y. Levy. Obtaining complete answers from incomplete databases. In *Int. Conf. on Very Large Data Bases, VLDB'96*, pages 402–412, 1996.
27. A.Y. Levy. Logic-based techniques in data integration. In Minker, J, ed., *Logic-Based Artificial Intelligence*, Kluwer, 2000.
28. P. McBrien and A. Poulovassilis. Data integration by bi-directional schema transformation rules. In *Int. Conf. on Data Engineering, ICDE 2003*, pages 227-238. IEEE Computer Society, 2003.
29. R. Pottinger and A.Y. Levy. A scalable algorithm for answering queries using views. In *Int. Conf. on Very Large Data Bases, VLDB 2000*, pages 484–495, 2000.
30. F. Sadri, F. Toni, and I. Xanthakos. A logic-agent based system for semantic integration. In *Proceedings of 17th International Data and Information for the Coming Knowledge Millennium Conference ,CODATA 2000*, 2000.
31. J.D. Ullman. Information integration using logical views. *Theoretical Computer Science*, 239(2):189–210, 2000.
32. K. Van Belleghem, M. Denecker, and D. De Schreye. A strong correspondence between description logics and open logic programming. In *Int. Conf. on Logic Programming, ICLP'97*, pages 346–360, 1997.
33. A. Van Gelder, K.A. Ross, and J.S. Schlipf. The Well-Founded Semantics for General Logic Programs. *Journal of the ACM*, 38(3):620–650, 1991.
34. S. Verbaeten and A. Bossi. Composing complete and partial knowledge. *Journal of Functional and Logic Programming*, 2000(6):1–25, 2000.
35. I. Xanthakos. *Semantic integration of information by abduction*. Phd thesis, University of London, United Kingdom, 2003.