Plausible Query-answering Inference in Data Integration

Zoran Majkić
Dept. of Computer Science,UMIACS, University of Maryland, College Park, MD 20742
zoran@cs.umd.edu
http://www.cs.umd.edu/ $\sim$ majkic/

Abstract

One of the main issue in formalizing the Data Integration Systems (DIS) is the semantic characterization of its global schema and the mappings with its source databases. Each DIS must be robust enough in order to take in account the incomplete and inconsistent information of its source databases, typical in Web applications: the extension of source databases change in an unpredictable way so that in different time instances we pass from a consistent to inconsistent DIS and viceversa. Thus, DIS will generally have possibly infinite number of consistent repairs and their models and, consequently, query answering in such DISs is very complex and time consuming. The current systems adopt the two extreme solutions for a query-answering: *certain* answers (true in all models of a given DIS) or *all possible* answers (true at any model). The first solution is to much strong requirement and practically non applicable in real situations, the second one is less meaningful (not all possible Skolem-based completions of an incomplete logical theory are plausible for users) and time/space consuming (they may be infinite also). In this paper we propose the middle solution between these two extremes based on the plausible nonmonotonic query-answering inference.

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 schema*.
The global schema is therefore a reconciled view of the information, which can be queried by the user. It can be thought of as a set of virtual relations, in the sense that their extensions are not actually stored anywhere. A data integration system frees the user from having to locate the sources relevant to a query, interact with each source in isolation, and manually combine the data from different sources.
Unlike a traditional query execution engine that communicates with a local storage manager to fetch the data, the query execution plan in the Web based data integration must obtain data from remote sources. A wrapper is a program (method) which is specific to a data source, whose task is to translate data from the source to a form that is usable by the query processor (agent) of the system. Thus, the extension of source databases is unpredictable, may be partially incomplete and with possible conflicting, mutually inconsistent information w.r.t. a subset of integrity constraints defined in a data integration systems.
An increasing amount of data is becoming available in the World-Wide Web, and the data is managed under an increasing diversity of data model and access mechanisms. Much of this data is semistructured. By semistructured data we mean data that has no absolute schema fixed in advance, and whose structure may be irregular or incomplete. A new set of requirements for query processing has emerged, as Internet and web-based query systems have become more prevalent. In this emerging data management domain, queries are posed over multiple *semistructured* information sources *distributed* across a wide-area network. Each source may be autonomous and may potentially have data of a different format and new sources are frequently added.
In certain contexts, the query processing system will handle a small number of concurrent queries; in others, there can be hundreds or even thousands of simultaneous requests. These different Internet query applications have many common requirements, but also require certain context-specific behaviors.
In modern query processors the query is first parsed and then passed to the query optimizer. The role of the optimizer is to produce an efficient query execution plan; the optimizer selects a query execution plan by searching a space of possible plans, and comparing their estimated cost. To evaluate the cost of a query execution plan the optimizer relies on extensive statistics about the underlying data, such as sizes of relations, sizes of domains and the selectivity of predicates.
However, data management systems for the Internet have demonstrated a pressing need for new techniques. Since data sources in this domain may be distributed, autonomous, and heterogeneous, the query optimizer will often not have histograms or any other quality statistics. Moreover, since the data is only accessible via a wide-area network, the cost of I/O operations is high, unpredictable and variable.
All these considerations, and from the fact that the DISs will generally have possibly infinite number of consistent repairs and their models and, consequently, query answering in such DISs is very complex and time consuming. The current systems adopt the two extreme solutions for a query-answering: *certain* answers (true in all models of a given DIS) or *all possible* answers (true at any model). The first solution is

too much strong requirement and practically non applicable in real situations, the second one is less meaningful (not all possible Skolem-based completions of an incomplete logical theory are plausible for users) and time/space consuming (they may be infinite also). In this paper we propose the middle solution between these two extremes based on the *plausible* nonmonotonic query-answering inference.
The plan of this paper is the following: After short introduction to deductive logics and Data Integration Systems, In Section 2 we introduce the model-theoretic structure for a plausible query-answering in (inconsistent) data integration, based on the abstract choice function over the set of possible repairing of a DIS, and we show that such set is a closed set in the class of all DISs. Finally, in Section 3 we present the inference for query-answering in Data Integration Systems with the proof that it is cumulative nonmonotonic inference relation.

Introduction to Deductive logic

The concepts introduced here, based on the work of Tarski, are basic to the development of nonmonotonic logic in the rest of the paper.
We assume that a fixed 2-valued (**f** -false, **t**-truee) object language $\mathcal{L}$ is given. The details of $\mathcal{L}$ are left open, except that it contains the standard connectives, $\Rightarrow, \wedge, \vee$ (material implication, conjunction and disjunction respectively). Hence, the set Φ of *sentences* of $\mathcal{L}$ is closed under the rules: $\mathbf{f} \in \Phi$; if $\alpha, \beta \in \Phi$ then $\alpha \Rightarrow \beta, \alpha \wedge \beta, \alpha \vee \beta \in \Phi$. $\neg\alpha$ is taken as a abbreviation of $\alpha \Rightarrow \mathbf{f}$. We use $\forall$ symbol for "for all" quantifier, and $\exists$ for existential quantifier.
By a *consequence relation* is a binary relation $\vdash$ which takes a set of sentences $\Gamma \subseteq \Phi$ as its first argument and a single sentences $\alpha \in \Phi$ as its second, denoted by $\Gamma \vdash \alpha$. Equivalently, we can define a *consequence operation* (infinitary) mapping $C_n : 2^\Phi \rightarrow 2^\Phi$ (2^Φ denotes the set of *all* subsets of Φ), such that $C_n(\Gamma) = \{\alpha \mid \Gamma \vdash \alpha\}$, and viceversa, $\Gamma \vdash \alpha$ iff $\alpha \in C_n(\Gamma)$. The finitary version of this operator is a mapping $C_{fin} : \mathcal{P}_{fin}(\Phi) \rightarrow 2^\Phi$, where $\mathcal{P}_{fin}$ is a finitary powerset operator. We write $C_n(\Gamma, \Delta)$ instead of $C_n(\Gamma \bigcup \Delta)$, and $C_n(\alpha)$ instead of $C_n(\{\alpha\})$. The following table define properties of the *deductive* monotonic consequence relation (Reflexivity, Cut, Monotonicity and Compactness):
The finitistic or "Gentzen-style" (Tab I):

$\alpha \in \Gamma$ implies $\Gamma \vdash \alpha$
$\Gamma \bigcup \Delta \vdash \alpha, \forall\beta \in \Delta.(\Gamma \vdash \beta)$ implies $\Gamma \vdash \alpha$
$\Gamma \vdash \alpha, \Gamma \subseteq \Delta$ implies $\Delta \vdash \alpha$
if $\Gamma \vdash \alpha$ then for some finite $\Delta \subseteq \Gamma$, $\Delta \vdash \alpha$

Tab I

The infinitistic or "Tarski-style" (Tab II):

$\Gamma \subseteq C_n(\Gamma)$
$\Delta \subseteq C_n(\Gamma)$ implies $C_n(\Gamma \bigcup \Delta) \subseteq C_n(\Gamma)$
$\Gamma \subseteq \Delta$ implies $C_n(\Gamma) \subseteq C_n(\Delta)$
$C_n(\Gamma) \subseteq \bigcup\{C_n(\Delta) \mid \Delta \subseteq \Gamma$ and Δ is finite $\}$

Tab II

Example 1: the classical propositional logic is a deductive logic.
□
It is easy to verify that C_n is a closure idempotent operator: for any Γ, we obtain the closet set, called *theory* $T = C_n(\Gamma) = C_n(C_n(\Gamma))$.
For $\Gamma, \Delta \subseteq \Phi$ we shall say that Γ is *consistent* iff $C_n(\Gamma) \neq \Phi$ and that Γ is *consistent with* Δ iff $C_n(\Gamma, \Delta) \neq \Phi$. A set is *inconsistent* iff it is not consistent.
A set Γ is $\mathcal{L}$-maximal iff is consistent and for every Δ, if $\Gamma \subseteq \Delta$ and Δ consistent, then $\Gamma = \Delta$.
We denote by $\mathcal{M}_L$ the set of all $\mathcal{L}$-maximal sets, and by $\mathcal{T}_L$ the set of all theories in 2^Φ, and by $|\Gamma| = \{m \in \mathcal{M}_L \mid \Gamma \subseteq m\}$ the set of all maximal extensions of Γ.
For any deductive logic the following properties hold:
1. Every $\mathcal{L}$-maximal set Γ is a theory (i.e., $\Gamma = C_n(\Gamma)$).
2. (Lindenbaum) Every consistent set is included in some $\mathcal{L}$-maximal theory.
3. if $\alpha \notin C_n(\Gamma)$, then there exists a $\mathcal{L}$-maximal theory m such that $C_n(\Gamma) \subseteq m$ and $\alpha \notin m$.
4. $C_n(\Gamma) = \bigcap |\Gamma|$, that is, α is a consequence of Γ iff α belongs to every maximal extension of Γ.
So far, our discussion has been purely syntactical and proof-theoretic. We shall also suppose that, with the language $\mathcal{L}$ and the consequence operator C_n comes a suitable *semantics* in the form of a set $\mathcal{U}$ (the universe), the elements of which we shall call worlds, and the relation $\models \subseteq \mathcal{U} \times 2^\Phi$ of *satisfaction* between worlds and formulae: $\alpha \in C_n(\Gamma)$ iff $\forall u \in \mathcal{U}.(u \models \Gamma$ implies $u \models \alpha)$.
We will use a deductive logics as underlying logic in order to define the inference operator for (generally non-monotonic) query-answering in general Data integration framework.

Technical preliminaries for Data Integration

In this section we illustrate the formalization of a data integration system (Lenzerini 2002) , which is based on the relational model with integrity constraints.
In the relational model, predicate symbols are used to denote the relations in the database, whereas constant symbols denote the values stored in relations. We assume to have a fixed (infinite) alphabet of constants , and, if not specified otherwise, we will consider only databases over such an alphabet. In such a setting, the UNA *unique name assumption* (that is, to assume that different constants denote different objects) is implicit.
A *relational schema* (or simply *schema*) is constituted by:
1. An *alphabet* $\mathcal{A}$ of predicate (or relation) symbols, each

one with the associated arity. i.e., the number of arguments of the predicate (or, attributes of the relation).
2. A set $\Sigma_{\mathcal{G}}$ of *integrity constraints*, i.e., assertions on the symbols of the alphabet $\mathcal{A}$ that express conditions that are intended to be satisfied in every database coherent with the schema.
A *relational database* (or simply, *database*) $\mathcal{DB}$ for a schema $\mathcal{C}$ is a set of relations with constants as atomic values, and with one relation $r^{\mathcal{DB}}$ of arity n for each predicate symbol r of arity n in the alphabet $\mathcal{A}$: the relation $r^{\mathcal{DB}}$ is the interpretation in $\mathcal{DB}$ of the predicate symbol r, in the sense that it contains the set of tuples that satisfy the predicate r in $\mathcal{DB}$.
A *relational query* is a formula that specifies a set of tuples to be retrieved from a database. The answer to a query q of arity n over a database $\mathcal{DB}$ for $\mathcal{G}$, denoted $q^{\mathcal{DB}}$, is the set of n-tuples of constants $(c_1, \ldots, c_n)$, such that, when substituting each x_i with c_i, the formula
$\exists(y_1, \ldots, y_n).q(x_1, \ldots, x_n, y_1, \ldots, y_m)$ evaluates to true in $\mathcal{DB}$.
We now turn our attention to the notion of data integration system.

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

with the following characteristics:

- The *global schema* is expressed in the relational model with constraints $\Sigma_{\mathcal{G}}$.
- The *source schema* is expressed without integrity constraints. We denote by $\mathcal{D}$ the current extension of all source databases represented by this source schema.
- The *mapping* $\mathcal{M}$ is defined following the GLAV (global/Local-as-view) approach, by the set of assertions of the form:
  $q_S \Rightarrow q_G$ or $q_G \Rightarrow q_S$
  where q_S and q_G are two queries of the same arity, respectively over the source schema $\mathcal{S}$, and over the global schema $\mathcal{G}$. Queries q_S are expressed in a query language $\mathcal{L}_{\mathcal{M},\mathcal{S}}$ over the alphabet of a source databases $\mathcal{A}_S$, and queries g_G are expressed in a query language $\mathcal{L}_{\mathcal{M},\mathcal{G}}$ over the alphabet $\mathcal{A}_G$ of a global schema.

We call *global database* for $\mathcal{I}$, or simply *database* for $\mathcal{I}$, any database for $\mathcal{G}$. A database $\mathcal{B}$ for $\mathcal{I}$ is said to be *legal* with respect to $\mathcal{D}$ if:

- $\mathcal{B}$ satisfies the integrity constraints $\Sigma_{\mathcal{G}}$ of $\mathcal{G}$;
- $\mathcal{B}$ satisfies $\mathcal{M}$ with respect to $\mathcal{D}$.

Intuitively, the source schema describes the structure of the sources, where the real data are, while the global schema provides a reconciled, integrated, and virtual view of the underlying sources. The assertions in the mapping establish the connection between the elements of the global schema and those of the source schema.
Queries to $\mathcal{I}$ are posed in terms of the global schema $\mathcal{G}$, and are expressed in a query language $\mathcal{L}_Q$ over the alphabet $\mathcal{A}_G$. A query is intended to provide the specification of which data to extract from the virtual database represented by the integration system.
The above definition of data integration system is general enough to capture virtually all approaches in the literature. Obviously, the nature of a specific approach depends on the characteristics of the mapping, and on the expressive power of the various schema and query languages. For example (Majkić 2004) , if we use the negation in query languages then we introduce a kind of general closed world assumption, and the inference query-answering operator becomes nonmonotonic.

Semantics for plausible query answering in data integration

The essential idea behind semantic modelling of nonmonotonic inference goes back to McCarthy's classical paper on *circumscription* (McCarthy 1980): the essential model-theoretic idea is to single out only a subset of "minimal" models. Shoham (Shoham 1987) generalized the concept of circumscription, or minimal entailment, to a more abstract notion: *preferential entailment*. Different other approaches are used in order to generalize the semantics for nonmonotonic reasoning, (S.Kraus, D.Lehmann, & M.Magidor 1990; S.Lindstrom ; N.Friedman & J.Y.Halpern 1996), based on preference structures and choice functions, and in (D.Lehman) is presented the comparison between these two semantic approaches. In this section is presented a reexamination of a *choice function* approach.
In the following we denote the fixed part of a DIS $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ (i.e., without the extension of its source data bases $\mathcal{D}$) by $\mathcal{I}_{fix}$, so that we denote by $\mathcal{I} = (\mathcal{I}_{fix}, \mathcal{D})$. Informally, $\mathcal{I}_{fix}$ is the set of sentences (usually universally quantified) of integrity constraints over global schema and mapping assertions (sentences). It may be the case that, for a given source data base extension $\mathcal{D}$, $\mathcal{I} = (\mathcal{I}_{fix}, \mathcal{D})$ is inconsistent. We denote with $\mathcal{D} \models_{\mathcal{I}} \mathcal{I}_{fix}$ the fact that for this extension of a source databases $\mathcal{D}$ all integrity constraints and mappings are satisfied. In this case we say that $\mathcal{D}$ is consistent w.r.t. $\mathcal{I}_{fix}$; otherwise we say that $\mathcal{D}$ is inconsistent w.r.t. $\mathcal{I}_{fix}$.
We denote by U the set of *all consistent* DISs. It is easy to verify that each model of a consistent DIS $\mathcal{I}$ is a $\mathcal{L}$-maximal set of the underlying deductive logic $\mathcal{L}$ for data integration systems (consider that a models of Φ are interpretations $f : \Phi \rightarrow 2$, i.e. subset of mappings $f \in 2^{\Phi}$ which satisfy logical connectives (that is, $f(\neg\alpha) = 1 - f(\alpha)$, $f(\alpha \wedge \beta) = min(f(\alpha), f(\beta))$, etc.). Thus each model $m = \{\alpha \in \Phi \mid f(\alpha) = 1\}$ is a closed set (theory in $\mathcal{L}$, such that $m = C_n(m)$). The Herbrand model is the restriction of such closed set (that is, model) m to only *ground atoms*.
We define the mapping $Mod : U \rightarrow 2^{\Phi}$, such that for any consistent DIS, $\mathcal{I} \in U$, the non-empty set $Mod(\mathcal{I})$ is the set of all models for this Data Integration System $\mathcal{I}$.
Now, following (M.Arenas, L.Bertossi, & J.Chomicki 1999), given an extension $\mathcal{D}$, possibly inconsistent with $\mathcal{I}_{fix}$, we say that, for a given extension (instance) $\mathcal{D}'$ of source databases, the DIS $\mathcal{I}' = (\mathcal{I}_{fix}, \mathcal{D}')$ is a *repair* for

$\mathcal{I}$ iff $\mathcal{D}' \models_{\mathcal{I}} \mathcal{I}_{fix}$.
We denote by $R(\mathcal{I})$ the set of all repairs of $\mathcal{I}$; if $\mathcal{I}$ is consistent, then $R(\mathcal{I}) = \{\mathcal{I}\}$.
We say that a sentence α holds in a state (i.e., DIS) $\mathcal{I} \in U$ (relative to the DIS-structure $\mathcal{I}_{\mathcal{M}}$ and write $\mathcal{I}_{\mathcal{M}} \Vdash_{\mathcal{I}} \alpha$ iff for every $m \in Mod(\mathcal{I}),\ \alpha \in m$. Intuitively, a sentence α is true in the state $\mathcal{I}$ (consistent data integration system) just in case α is true in all models (set $Mod(\mathcal{I})$) of this consistent DIS.
The set of consistent DISs in which α holds will be written $\widehat{\alpha} = \{\mathcal{I} \in U \mid \mathcal{I}_{\mathcal{M}} \Vdash_{\mathcal{I}} \alpha\} = \{\mathcal{I} \in U \mid \alpha \in \bigcap Mod(\mathcal{I})\}$, and for a set of sentences Γ:
$\widehat{\Gamma} = \bigcap(\{\widehat{\alpha} \mid \alpha \in \Gamma\} = \{\mathcal{I} \in U \mid \Gamma \subseteq \bigcap Mod(\mathcal{I})\}$, so by monotonicity of C_n, and the fact that $\bigcap Mod(\mathcal{I})$ is a closed set, that is $\bigcap Mod(\mathcal{I}) = C_n(\bigcap Mod(\mathcal{I}))$, we obtain
$\widehat{\Gamma} = \{\mathcal{I} \in U \mid C_n(\Gamma) \subseteq \bigcap Mod(\mathcal{I})\}$, that is, $\widehat{\Gamma}$ is the set of all DISs in which all sentences in Γ are accepted.
Given a set of consistent DISs, $X \subseteq U$, we may also define the set $th(X)$ of sentences that are accepted in all DISs in X, i.e., $th(X) = \{\alpha \mid X \subseteq \widehat{\alpha}\}$.

Proposition 1 *For any set X of consistent DISs the set of accepted sentences $th(X)$ is a theory in $\mathcal{L}$.*

Proof 1: $th(X) = \{\alpha \mid X \subseteq \widehat{\alpha}\} = \{\alpha \mid \forall \mathcal{I} \in X.\alpha \in \bigcap Mod(\mathcal{I})\} = \bigcap\{\bigcap Mod(\mathcal{I}) \mid \mathcal{I} \in X\}$, thus from the fact that the intersection of closed sets is closed set, we obtain that $th(X)$ is a theory. In the singleton case we obtain the theory $th(\{\mathcal{I}\}) = \bigcap Mod(\mathcal{I})$, while $th(\{\}) = \Phi$.
□
It is easy to verify these two mappings form a Galois connection between $\mathcal{P}(\Phi)$ and $\mathcal{P}(U)$, that is hold
1. if $\Gamma \subseteq \Delta$ then $\widehat{\Delta} \subseteq \widehat{\Gamma}$,
2. if $X \subseteq Y$ then $th(Y) \subseteq th(X)$,
3. $\Gamma \subseteq th(\widehat{\Gamma})$,
4. $X \subseteq \widehat{th(X)}$.
Thus the mapping $C_{\mathcal{I}} = \widehat{\ } \circ th : \mathcal{P}(U) \to \mathcal{P}(U)$ is a closure operation on the set of DISs, that is hold for all $X, Y \in U$:
1. $X \subseteq C_{\mathcal{I}}(X) = \{\alpha \mid \widehat{X \subseteq \widehat{\alpha}}\} = \bigcap\{\widehat{\alpha} \mid X \subseteq \widehat{\alpha}\}$,
2. if $X \subseteq Y$ then $C_{\mathcal{I}}(X) \subseteq C_{\mathcal{I}}(Y)$,
3. $C_{\mathcal{I}}(X) = C_{\mathcal{I}}(C_{\mathcal{I}}(X))$.

Proposition 2 *Each closed (sub)set $X \subseteq U$ is the set of all repairs of some data integration system $\mathcal{I}$. We denote by U_{clo} the set of all closed subsets of U.*

Proof 2: Let X be the set of all repairs of some DIS $\mathcal{I}_1$. Let prove that $X = C_{\mathcal{I}}(X)$. From the property of closure operator, holds that $X \subseteq C_{\mathcal{I}}(X)$. Let $\mathcal{I} \in C_{\mathcal{I}}(X)$ and prove that $\mathcal{I} \in X$.
From $\mathcal{I} \in C_{\mathcal{I}}(X)$ we have that $\mathcal{I} \in \bigcap\{\widehat{\alpha} \mid X \subseteq \widehat{\alpha}\} = \bigcap\{\widehat{\alpha} \mid \mathcal{I}' \in \widehat{\alpha} \text{ for all } \mathcal{I}' \in X\}$. Suppose that $\mathcal{I} \notin X$, i.e. that $\mathcal{I}$ is not a repair of $\mathcal{I}_1$ so that does not satisfy the constraints in $\mathcal{I}_1$, but it is not possible because all members in the set $\bigcap\{\widehat{\alpha} \mid \mathcal{I}' \in \widehat{\alpha} \text{ for all } \mathcal{I}' \in X\}$ satisfy such constraints.
□
Example 2: For a consistent DIS $\mathcal{I} \in U$, we have that $X = \{\mathcal{I}\} = R(\mathcal{I})$ is closed subset of U, that is $\{\mathcal{I}\} = C_{\mathcal{I}}(\{\mathcal{I}\})$.
□
Let us describe the semantics for plausible query answering in data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$ by the following structure:

Definition 2 *A DIS-structure based on the deductive logic $\mathcal{L}$ is a tuple $\mathcal{I}_{\mathcal{M}} =< U, Mod, F >$, where $F : U_{clo} \to \mathcal{P}(U)$ is a choice function such that for any inconsistent Data Integration System $\mathcal{I}$ takes its preferred repairs, that is for any $\mathcal{I} \in U$ holds:*
$F(R(\mathcal{I})) \subseteq R(\mathcal{I})$,
where $R(\mathcal{I})$ is a closed subset of U composed by alll repairs of $\mathcal{I}$.

Example 3: For a consistent DIS $\mathcal{I} \in U$, we have that $R(\mathcal{I}) = \{\mathcal{I}\}$, thus $F(R(\mathcal{I})) = F(\{\mathcal{I}\}) \subseteq R(\{\mathcal{I}\}) = \{\mathcal{I}\}$. So, we obtain that $F(\{\mathcal{I}\}) = \{\mathcal{I}\}$.

Cumulative inference for query-answering in data integration

Instead of monotonic deduction, presented in the introduction, we will consider nonmonotonic inference, considering that, generally, the query-answering in data integration is nonmonotonic (for example, when we use negation operator in query languages, in the case of incomplete information in sources, or in the case when we consider only a subset of preferred repairs for mutually inconsistent information coming from different source databases).
The natural way to relax the monotonicity property can be obtained by introducing the Cautious monotonicity: it was introduced by Gabbay (D.Gabbay) for finite set of premises (Gentzen-style), and by Makinson (D.Makinson) (Tarski-style) for a more general infinite set of premises. The last one is a necessary condition in the data integration framework: the incomplete information introduce Skolem functions and infinite Herbrand bases, so that, together with recursive logic specification of a global schema constraints, the models for databases are possibly infinite.
The following table presents the definition for a *cumulative* nonmonotonic inference relation $\vdash_C$ and operation C_C (Reflexivity, Cut and Cautious monotonicity):

Finitistic or "Gentzen-style" (Tab III):

$\alpha \in \Gamma$ implies $\Gamma \vdash_C \alpha$
$\Gamma \bigcup \Delta \vdash_C \alpha, \forall \beta \in \Delta.(\Gamma \vdash_C \beta)$ implies $\Gamma \vdash_C \alpha$
$\forall \beta \in \Delta.(\Gamma \vdash_C \beta), \Gamma \vdash_C \alpha$ implies $\Gamma \bigcup \Delta \vdash_C \alpha$

Tab III

Infinitistic or "Tarski-style" (Tab IV):

$\Gamma \subseteq C_C(\Gamma)$
$\Delta \subseteq C_C(\Gamma)$ implies $C_C(\Gamma \bigcup \Delta) \subseteq C_C(\Gamma)$
$\Delta \subseteq C_C(\Gamma)$ implies $C_n(\Gamma) \subseteq C_C(\Gamma \bigcup \Delta)$

Tab IV

It is easy to verify that the Cut and Cautious monotonicity can be combined into a simple principle of *cumulation*:

$\Gamma \subseteq \Delta \subseteq C_C(\Gamma)$ implies $C_C(\Gamma) = C_C(\Delta)$

or to the following two conditions (S.Kraus, D.Lehmann, & M.Magidor 1990):

- Right Weakening: if $\forall \alpha \in \Delta.\ \Gamma \vdash_C \alpha$, and $\Delta \vdash \beta$, then $\Gamma \vdash_C \beta$
  that is, $C_n(C_C(\Gamma)) \subseteq C_C(\Gamma)$
- Left Logic Equivalence: if $\forall \alpha.(\Gamma \vdash \alpha$ iff $\Delta \vdash \alpha)$, then $\Gamma \vdash_C \beta$ iff $\Delta \vdash_C \beta$
  that is, $C_n(\Gamma)) = C_n(\Delta)$ implies $C_C(\Gamma)) = C_C(\Delta)$

The common idea in the literature on nonmonotonic reasoning is the following: α is a nonmonotonic consequence of Γ, that is $\Gamma \vdash_C \alpha$, just in the case α holds in all those Γ-states (in our case Γ-DISs repairs) that are maximally plausible. Formally we represented this idea by introducing a *choice function* F which, given a set $R(\mathcal{I})$ of all repairs of DID $\mathcal{I}$, picks out the set $F(R(\mathcal{I}))$ of all the "best" repairs in $R(\mathcal{I})$.
For the DIS-structure $\mathcal{I}_{\mathcal{M}}$, based on the deductive logic $\mathcal{L}$, we define the following relation $|\!|\!\vdash$ between sets of sentences and single sentences:

- $\Gamma |\!|\!\vdash \alpha$ iff $F(\widehat{\Gamma}) \subseteq \widehat{\alpha}$

We have to demonstrate that this relation is nonmonotonic consequence (or plausible inference). First, this definition will in general lead to $|\!|\!\vdash$ being nonmonotonic, since there is no guarantee that $F(\widehat{\Gamma}) \subseteq \widehat{\alpha}$ will imply that $F(\widehat{\Gamma \bigcup \{\alpha\}}) \subseteq \widehat{\alpha}$.
Clearly, one the best preferred $\Gamma \bigcup \{\alpha\}$-states may fail to be a best preferred member of the more inclusive class of Γ-states. Therefore, it need not be the case that $F(\widehat{\Gamma \bigcup \{\alpha\}}) \subseteq \widehat{\Gamma}$. Neither does it follow that $F(\widehat{\Gamma \bigcup \{\alpha\}}) \subseteq \widehat{\alpha}$.
Different choices on the selection function F will give rise to different non monotonic logics:

Example 4: Let consider the choice function, in the case of the so called minimal repairs w.r.t. *set inclusion preference criterion* (M.Arenas, L.Bertossi, & J.Chomicki 1999): the distance between two database instances $\mathcal{D}_1$ and $\mathcal{D}_2$ is their symmetric difference $\delta(\mathcal{D}_1, \mathcal{D}_2) = (\mathcal{D}_1 - \mathcal{D}_2) \bigcup (\mathcal{D}_2 - \mathcal{D}_1)$. It is *minimal* under set inclusion in the class of instances that satisfy $\mathcal{I}_{fix}$.
Other example for the choice function is the *minimal cardinality preference criterion* (M.Dalal 1988): used in order to minimize deletions and insertions of tuples during a repairing.
□
Let now show the fundamental property of the introduced relation $|\!|\!\vdash$ form Data Integration Systems:

Proposition 3 *If $\mathcal{I}_{\mathcal{M}}$ is a DIS-structure based on the deductive logic $\mathcal{L}$ then $|\!|\!\vdash$ is a cumulative inference relation based on $\mathcal{L}$. We define the cumulative operation C, as follows: for any $\Gamma \subseteq \Phi$, $C(\Gamma) = \{\alpha \mid \Gamma |\!|\!\vdash \alpha\}$.*
Proof is analog to the proof of the Lemma 4.4. in (S.Lindstrom).
□
Let now consider the problem of a plausible query-answering in Data Integration Systems. Let $q(x) \in \mathcal{L}_Q$ be an user query over a global schema $\mathcal{G}$ of the (possibly inconsistent) Data integration system $\mathcal{I} = \langle \mathcal{G}, \mathcal{S}, \mathcal{M} \rangle$. We can consider the set of sentences $th(R(\mathcal{I})) = \{\alpha \mid R(\mathcal{I}) \subseteq \widehat{\alpha}\}$, where $R(\mathcal{I})$ is a set all repairs of $\mathcal{I}$, and define the relation $\models_I$ as follows:

- $\mathcal{I} |\!|\!\vdash_I q(c)$ iff $th(R(\mathcal{I})) |\!|\!\vdash q(c)$

where $q(c)$ is a (ground) sentence obtained by substitution of a variables in x by database constants in a query formula. Thus, a plausible answer to the query $q(x)$ can be defined by this cumulative nonmonotonic inference as follows:

$q^{\mathcal{DB}} = \{q(c) \mid \mathcal{I} |\!|\!\vdash_I q(c)$, c is a tuple of constants of a fixed DIS alphabet$\}$.

This plausible query-answering in DISs is a general one: it holds for any particular query language $\mathcal{L}_Q$, any kind of mappings between source databases and a global schema, and in presence of incomplete and inconsistent information.

Example 5: Let us see, that in the case of a consistent DIS, this plausible answering correspond to the certain (or known) ansering to queries.
When DIS $\mathcal{I}$ is consistent, then $R(\mathcal{I}) = \{\mathcal{I}\}$, and $th(R(\mathcal{I})) = th(\{\mathcal{I}\}) = \bigcap Mod(\mathcal{I})$, thus, for any given query $q(x)$ we have that:

- $\mathcal{I} |\!|\!\vdash_I q(c)$ iff $th(\{\mathcal{I}\}) |\!|\!\vdash q(c)$ iff $F(\widehat{th(\{\mathcal{I}\})} \subseteq \widehat{q(c)}$

So, from the fact that $F(\widehat{th(\{\mathcal{I}\})}) = F(C_{\mathcal{I}}(\mathcal{I})) = F(\{\mathcal{I}\}) = \{\mathcal{I}\}$, and the fact that $\widehat{q(c)} = \{\mathcal{I}' \in U \mid q(c) \in \bigcap Mod(\mathcal{I}')\}$, we obtain that must hold
$\{\mathcal{I}\} \subseteq \{\mathcal{I}' \in U \mid q(c) \in \bigcap Mod(\mathcal{I}')\}$, that is must hold $q(c) \in \bigcap Mod(\mathcal{I})\}$. That is, $q(c)$ must hold in all models of this consistent $\mathcal{I}$, i.e., $q(c)$ is a certain (or known) answer of this Data Integration System.
□

Conclusion

The problem of answering queries in Data Integration systems raises a multitude of challenges, ranging from theoretical foundations to considerations of a more practical nature. The algorithms for answering queries using views are already incorporated into a number of data integration systems with integrity constraints to obtain certain answers. The difficulties basically arise because of the need of dealing with incomplete information and, moreover, with mutually inconsistent information which comes from different source databases: in such case we need some kind of plausible query-answering. A number of different partial solutions, based on the model-theoretic approach are adopted in practice, without an unifying general framework. We presented

such general framework for plausible query-answering inference, based on choice functions. As result we obtained a cumulative nonmonotonic inference for query-answering in data integration.

References

D.Gabbay. Theoretical foundations for non-monotonic reasoning in expert systems. *Logics and Models of Concurrent Systems, Springer Verlag, NATO ASI Series, Vol.F13*.

D.Lehman. Non monotonic logics and semantics. *Tech. Rep. TR-98-6, Institute of Comp. Science, Hebrew University, Jerusalem*.

D.Makinson. General theory of cumulative inference. *Non-Monotonic Reasoning, Spinger Verlag, Lecture Notes on Artificial Intelligence, n.346*.

Lenzerini, M. 2002. Data integration: A theoretical perspective. In *Proc. of the 21st ACM SIGACT SIGMOD SIGART Symp. on Principles of Database Systems (PODS 2002)*, 233–246.

Majkić, Z. 2004. Closed world assumption for gav data integration systems with key integrity constraints. *Notes in http://www.dis.uniroma1.it/ ∼ majkic/*.

M.Arenas; L.Bertossi; and J.Chomicki. 1999. Consistent query answers in inconsistent databases. *Proc. ACM Symp. on Principles of Database Systems* 68–79.

McCarthy, J. 1980. Circumscription — a form of non-monotonic reasoning. *Artificial Intelligence* 13:27–39,171–172.

M.Dalal. 1988. Investigations into a theory of knowledge base revision. *Proc. National Conference on Artificial Intelligence* 475–479.

N.Friedman, and J.Y.Halpern. 1996. Plausibility measures and default reasoning. *Proc. AAAI' 96* 1297–1304.

Shoham, Y. 1987. A semantical approach to nonmonotonic logics. In *Proc. of the 2nd IEEE Symp. on Logic in Computer Science (LICS'87)*, 275–279.

S.Kraus; D.Lehmann; and M.Magidor. 1990. Nonmonotonic reasoning, preferential models and cumulative logics. *Artificial Intelligence 44* 167–207.

S.Lindstrom. A semantical approach to nonmonotonic reasoning: Inference operations and choice. *Tech. Rep. Upsala Prints and Preprints in Philosophy, 1994:10, University of Upsala*.