Open and Closed World Assumptions in Data Exchange

Leonid Libkin[1] and **Cristina Sirangelo**[2]

[1] School of Informatics, University of Edinburgh
[2] LSV, ENS–Cachan, CNRS and INRIA

Data exchange is the problem of finding an instance of a target schema, given an instance of a source schema and a specification of a mapping between the source and the target schemas, and answering queries over target instances. A schema mapping is a condition, often expressed in a fragment of first-order logic, that relates instances of source and target schemas. Most commonly it is a conjunction of sentences of the form

$$\forall \bar{x} \; \varphi(\bar{x}) \to \exists \bar{z} \; \psi(\bar{x}, \bar{z}), \tag{1}$$

where φ and ψ are both conjunctions of atomic formulae. For example, if we have a source database $S(\cdot, \cdot)$ that lists employees and their salaries, and a target database $T(\cdot, \cdot)$ that lists people and their children, then a natural mapping is $\forall x \forall y \; (S(x, y) \to \exists z \; T(x, z))$.

The study of both data exchange and schema mappings has been actively pursued recently (see, e.g., recent surveys [3, 11, 4]). Existing implementations have been incorporated into major database products.

Theoretical foundations of data exchange were developed in [5, 6]. For a source instance S and a schema mapping M, a target instance T is a *solution* for S if S and T together satisfy the conditions of M. For instance, if we have a mapping given by a conjunction of sentences of the form (1), it means that for each tuple $\bar{a}$ making $\varphi(\bar{a})$ true in the source, one can find a tuple $\bar{b}$ of values in the target database so that the conjunction $\psi(\bar{a}, \bar{b})$ is true in the target.

This semantics implies that for the same source instance one can have many different target solutions. In fact this semantics of data exchange follows the so called *open world assumption (OWA)*, as target solutions may contain arbitrary facts, some of which are not directly implied by the source data and the schema mapping. An alternative approach, developed in [12] adopts the so called *closed world assumption (CWA)* [14], for defining the semantics of data exchange. Here intuitively only facts which are a direct "translation" of source data are allowed in the target.

If a query Q is posed against target instances, then it must be answered in a way that is consistent with the source. A common approach is to look for *certain answers*:

$$\mathsf{certain}_M(Q, S) \;=\; \bigcap \{Q(T) \mid T \text{ is a solution}\}. \tag{2}$$

A desirable feature in data exchange is to find a particular solution to be materialized in the target, which allows to answer queries over the target (with certain

answer semantics), with no access to source data. In other words, we want a specific target instance T_0 so that $\mathsf{certain}_M(Q, S) = Q'(T_0)$, for some query Q'.

Data exchange literature identified two particularly nice solutions, called the *canonical solution* [5] and the *core* [6]. It was shown in [5, 6] that, under the OWA, certain answers (2) can be found in polynomial time if Q is a union of conjunctive queries (i.e., in the $(\exists, \wedge, \vee)$-fragment of FO). In fact, in that case it is easy to construct a query Q' so that $\mathsf{certain}_M(Q, S) = Q'(T_0)$ if T_0 is either the canonical solution, or the core.

While this looks reasonable, the definition of query answering is actually underspecified. We know well how to answer queries on complete databases; in contrast, databases arising in data exchange are often incomplete as they contain *null values* (for example, it is common for target schemas to have attributes that do not correspond to any attributes in the source schema; these are populated with nulls). Thus, the notions of query answering over incomplete databases are essential in data exchange, and have to be incorporated into the algorithms for answering queries in data exchange.

Let us briefly recall the standard approach to query answering over databases with incomplete information [10]. Such a database is formally a relational structure over the disjoint union of two domains: D of data values, and V of variables (or nulls, usually denoted by the symbol $\perp$ with subscripts). For such a database R, we let $Rep(R)$ denote the set of all complete databases (over D) that it represents. This can be interpreted again under two assumptions: under the closed world assumption, $Rep(R)$ consists of all databases obtained by applying mappings $V \to D$ to R; under the open world assumption, $Rep(R)$ is the set of all databases that contain $v(R)$, for some mapping $v : V \to D$.

The answers to a query Q over R are defined as

$$\square Q(R) \;=\; \bigcap \{Q(R') \mid R' \in Rep(R)\}.$$

That is, $\square Q(R)$ contains answers independent of the interpretation of nulls. One easy case of finding such answers is known: if Q is a union of conjunctive queries, then $\square Q(R)$, under both closed and open world assumptions, is obtained by a straightforward evaluation of Q on R and then discarding tuples that contain nulls (elements of V). This is referred to as *naive* evaluation. Clearly, naive evaluation is done in PTIME; adding negation to queries (e.g., going to the full FO) makes the problem coNP-complete under the closed world assumption [1] and undecidable under the OWA.

The theory of incomplete information explains why the OWA works well for answering unions of conjunctive queries in data exchange. In fact the canonical solution in data exchange, viewed as an open-world incomplete database, turns out to represent the set of all data exchange solutions under the OWA (property already referred to as *universality* in [5]). It follows that, if T_c is the canonical solution, $\square Q(T_c)$ computes data exchange certain answers $\mathsf{certain}_M(Q, S)$ under the OWA. If Q is a union of conjunctive query, all one needs to do is to compute the canonical solution, and naively evaluate the query over it – then results of [10] mentioned above guarantee that the process computes certain answers.

However, going beyond unions of conjunctive queries poses problems. They appear even in the simplest mappings of the form $\forall \bar{x}\ R(\bar{x}) \rightarrow R'(\bar{x})$ that copy contents of each relation R into a new relation R'. Under the OWA, even FO query answering in data exchange is undecidable, and for every Boolean query, either its certain answers are false for *all* instances, or the certain answers for its negation are false for all instances [2]. There are other examples of anomalous behavior of query answering in data exchange.

The reason behind such anomalies is that our intuition tells us that the mapping $\forall \bar{x}\ R(\bar{x}) \rightarrow R'(\bar{x})$ forces R' to be a copy of R; however, under the OWA semantics, R' is allowed to be an arbitrary extension of R. It seems natural then to adopt the CWA that would not permit adding arbitrary facts to R' and would only put there what the constraints dictate – that is, a copy of R.

This approach was developed in [12]. It introduced the notion of CWA-solutions based on three assumptions. First, every fact put in a target instance must be justified. Second, every justification for putting a tuple into the target should not generate multiple nulls. And third, every fact true in the target instance must be derivable from the source instance and constraints in the mappings. When these are properly formalized, the CWA-solutions can be characterized as homomorphic images of the canonical solution that have a homomorphism back into the canonical solution. It follows that every CWA-solution contains a copy of the core [12].

When the definition of $\mathsf{certain}_M(Q, S)$ is applied to CWA-solutions, a proper notion of query answers $Q(T)$ has to be adopted for CWA-solutions T. The most natural semantics is $\Box Q(T)$, where T is viewed as a closed-world incomplete database. This approach naturally leads to a new semantics of query answering: a tuple t is in $\mathsf{certain}_M(Q, S)$ under the CWA if, for every CWA-solution T and every $R \in Rep(T)$, this tuple is in $Q(R)$. That is,

$$\mathsf{certain}_M^{\mathrm{CWA}}(Q, S) \;=\; \bigcap\{Q(R) \mid R \in Rep(T),\ T \text{ is a CWA-solution.}\}$$

It was shown in [12] that the canonical solution is also a CWA-solution and, viewed as a closed-world incomplete database, represents all databases that are in $Rep(T)$ for some CWA-solution T. It follows that, as in the OWA case, the certain answers can be computed over the canonical solution T_c: namely, $\mathsf{certain}_M^{\mathrm{CWA}}(Q, S) = \Box Q(T_c)$. Hence, the problem of answering queries in data exchange is reduced to the well-studied problems of answering queries in closed-world databases with nulls. In particular, from [1, 10] one derives that computing $\mathsf{certain}_M^{\mathrm{CWA}}(Q, S)$ is coNP-complete for arbitrary FO queries and tractable for unions of conjunctive queries (for which the results coincide under both the OWA and the CWA).

Extensions of this approach to mappings that involve constraints over target schemas were developed in [9].

But fully open or fully closed mappings, being two extreme cases, are bound to have their shortcomings. Consider again the example of a source database $S(\cdot, \cdot)$ that lists employees and their salaries, and a target database $T(\cdot, \cdot)$ that lists people and their children. The mapping is given by $\forall x \forall y\ (S(x, y) \rightarrow$

$\exists z\ T(x, z))$. Under the CWA, if we assume that each employee has one salary, in the target each employee will have exactly one value for the 'child' attribute. This example indicates that it is natural to declare some target attributes as *open*, and some as *closed*. In the above example, we would formulate the mapping as $\forall x \forall y\ (S(x, y) \rightarrow \exists z\ T(x^{cl}, z^{op}))$, saying that only employees from the source are moved to the target, but the number of values of the 'child' attribute are arbitrary.

Such mixed mappings were studied in [13]. Data exchange solutions under mixed mappings are incomplete databases where domain elements are annotated as either open (*op*) or closed (*cl*). Such an approach for databases with nulls was used back in the 80s, see [8]. If T is such an annotated instance, $Rep_A(T)$ is defined so as to apply the OWA on values annotated as open, and the CWA on values annotated as closed. In particular, to get an instance in $Rep_A(T)$, after applying a valuation v to T, any tuple in $v(T)$ can be replicated arbitrarily many times by changing the values of attributes annotated as open, but keeping the same value of attributes annotated as closed. For example, $Rep_A(\{(a^{cl}, \perp^{op})\})$ contains all relations R whose projection on the first attribute is $\{a\}$. Similarly $Rep_A(\{(a^{cl}, \perp_1^{op}, \perp_2^{cl})\})$ contains all relations whose first projection is $\{a\}$ and whose third projection is a singleton. On the contrary, $Rep_A(\{(a^{cl}, \perp^{cl})\})$ contains all one-tuple relations $\{(a, b)\}$ for some constant b. If all annotations in T are open (closed, resp), clearly $Rep_A(T)$ coincides with $Rep(T)$ under the OWA (CWA, resp.).

Annotated data exchange solutions have been defined along the lines of CWA-solutions. Each tuple put in the target still needs to be justified by some source data and the schema mapping. The tuple annotation (determined by the schema mapping) will then give either and open-world or closed-world semantics to the tuple attributes. The restriction that all facts true in the target must be implied by the source and the schema mapping is enforced only limited to closed attributes.

A particular annotated solution, the *annotated canonical solution*, has the usual property that it represents the semantics of all the solutions. That is, if T_c is the annotated canonical solution, $Rep_A(T_c)$ contains precisely all the instances R that are in $Rep_A(T)$ for some annotated solution T. When annotation is removed from T_c, this is the usual canonical solution of [5] and [12], obtained by disregarding annotation in the schema mapping. This also shows, as proved in [13], that if a mapping has all open (resp, all closed) annotation, the semantics of annotated solutions coincides with the semantics of the OWA solutions of [5] (respectively CWA-solutions of [12]).

The semantics of query answering is standard: for an annotated schema mapping M_α a tuple t is $\mathsf{certain}_{M_\alpha}(Q, S)$ if, for every annotated solution T and every $R \in Rep_A(T)$, this tuple is in $Q(R)$. Since the semantics of the annotated canonical solution T_c captures the semantics of all solutions, the usual equation $\mathsf{certain}_{M_\alpha}(Q, S) = \square Q(T_c)$ holds also under annotated mappings. Thus the complexity of computing certain answers for arbitrary FO queries, must range from

coNP-complete (in the case of all 'closed' annotation) to undecidable (in the case of all 'open' annotation).

In particular it was shown that for query answering, there is a complexity trichotomy that depends on the number of target attributes declared as open per each atom in a rule in the mapping. If the number is 0 (i.e., we are under the CWA), then the complexity is in general coNP-complete. If the number is 2 or greater, then, as for the OWA, the problem is undecidable. But if the number is 1, the problem is decidable, albeit with a high complexity: coNEXPTIME-complete. Note that this means that each atom in a rule has at most one open annotation; the rule itself (or the mapping) can have arbitrarily many open annotations. We prohibit atoms such as $T(x^{op}, y^{op})$ but can have, for example, $T_1(x^{op}, y^{cl}) \wedge T_1(y^{op}, z^{cl}) \wedge T_2(x^{op}, z^{cl})$.

In the case of unions of conjunctive queries, all the cases collapse and the usual naive evaluation over the canonical solution produces the answer in polynomial time.

Finally, [13] considered the issue of *composing* schema mappings, and showed that the OWA approach of [7] extends to CWA mappings, but does not guarantee composition in general for mixed mappings.

References

1. S. Abiteboul, P. Kanellakis, G. Grahne. On the representation and querying of sets of possible worlds. *TCS* 78 (1991), 158–187.
2. M. Arenas, P. Barceló, R. Fagin, L. Libkin. Locally consistent transformations and query answering in data exchange. In *PODS 2004*, pages 229–240.
3. P. Barceló. Logical foundations of relational data exchange. *SIGMOD Record* 38 (2009).
4. P. Bernstein, S. Melnik. Model management 2.0: manipulating richer mappings. *SIGMOD'07*, pages 1–12.
5. R. Fagin, Ph. Kolaitis, R. Miller, L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.* 336(1): 89–124 (2005).
6. R. Fagin, Ph. Kolaitis, L. Popa. Data exchange: getting to the core. *ACM TODS* 30(1): 174–210 (2005).
7. R. Fagin, Ph. Kolaitis, L. Popa, W.C. Tan. Composing schema mappings: second-order dependencies to the rescue. *ACM TODS* 30(4) 994–1055 (2005).
8. G. Gottlob, R. Zicari. Closed world databases opened through null values. In *VLDB'88*, pages 50–61.
9. A. Hernich, N. Schweikardt. CWA-solutions for data exchange settings with target dependencies. In *PODS'07*, pages 113–122.
10. T. Imielinski, W. Lipski. Incomplete information in relational databases. *J. ACM* 31 (1984), 761–791.
11. Ph. Kolaitis. Schema mappings, data exchange, and metadata management. In *PODS 2005*.
12. L. Libkin. Data exchange and incomplete information. In *PODS'06*, pages 60–69.

13. L. Libkin and C. Sirangelo. Data exchange and schema mappings in open and closed worlds In *PODS'08*, pages 139–148.
14. R. Reiter. On closed world databases. In *Logic and Databases*, Plenum Press, 1978, pages 55–76.