

# Unifying Causality, Diagnosis, Repairs and View-Updates in Databases

Leopoldo Bertossi  
Carleton University  
School of Computer Science  
Ottawa, Canada  
bertossi@scs.carleton.ca

Babak Salimi  
Carleton University  
School of Computer Science  
Ottawa, Canada  
bsalimi@scs.carleton.ca

## ABSTRACT

In this work we establish and point out connections between the notion of query-answer causality in databases and database repairs, model-based diagnosis in its consistency-based and abductive versions, and database updates through views. The mutual relationships among these areas of data management and knowledge representation shed light on each of them and help to share notions and results they have in common. In one way or another, these are all approaches to uncertainty management, which becomes even more relevant in the context of big data that have to be made sense of.

## 1. INTRODUCTION

Causality is not only a deep subject that appears at the foundations of many scientific disciplines, but also something we want to represent and compute in order to deal with the *uncertainty* of data, information and theories. In data management, the need to understand and compute why certain (query) results are obtained or not, or why certain natural semantic conditions are not satisfied can only grow and become more complex when confronted with big data. *It is difficult to make sense of the uncertainty associated to big data*, and causality is a fundamental and systematic way to confront the problem.

Our current research is motivated by trying to understand *causality in data management* from different perspectives. As described below, there are fruitful connections among four forms of reasoning: inferring causes from databases, model-based diagnosis, consistent query answering (and repairs), and view updates. They all reflect some sort of *uncertainty* about the information at hand.

When querying a database, a user may not obtain the expected results, and the system could provide some explanations. They could be useful to further understand the

data or reconsider the query. A notion of causality-based explanation for a query result was introduced in [21].

Intuitively, a tuple  $t$  is an *actual cause* for an answer  $\bar{a}$  to a conjunctive query  $\mathcal{Q}$  from a relational database instance  $D$  if there is a “contingent” set of tuples  $\Gamma$ , such that, after removing  $\Gamma$  from  $D$ , removing/inserting  $t$  from/into  $D$  causes  $\bar{a}$  to switch from being an answer to being a non-answer. Actual causes and contingent tuples are restricted to be among a pre-specified set of *endogenous tuples*, which are admissible, possible candidates for causes, as opposed to *exogenous tuples*. (For non-causality-based explanations for query answers in DL ontologies, see [4].)

Since some causes may be stronger than others, [21] also introduces and investigates *responsibility*, that reflects the relative degree of actual causality. In applications involving large data sets, it is crucial to rank potential causes according to their responsibilities [20, 21].

Actual causation and responsibility, as used in [21], can be traced back to [15, 6]. For connections between causality and provenance, see [21, 20].

Model-based diagnosis, and *consistency-based diagnosis* in particular [23], is an area of knowledge representation. A *system specification* in a logical formalism and an *observation* are given. Typically, the specification tells us how the system works under normal conditions, the observation is unexpected under those conditions, and an *explanation* for the *failure* is required.

In a different direction, a database instance  $D$  may not satisfy certain intended integrity constraints (ICs). A *repair* of  $D$  is a database  $D'$  that does satisfy the ICs and *minimally departs* from  $D$ . Different forms of minimality have been investigated. A *consistent answer* to a query  $\mathcal{Q}$  from  $D$  wrt. the ICs is an answer to  $\mathcal{Q}$  that is obtained from all possible repairs, i.e. is *invariant* or *certain* under the class of repairs (see [2] for a recent survey). (Not in the framework of repairs, consistency-based diagnosis techniques have been applied to restoring a DB from IC violations [12].)

Interestingly, deeper and useful connections between these areas (and others, see below) are starting to emerge. Actually, we have reported in [24], where more results and details can be found, on new results about precise connections between causality for query answers in databases, database

repairs wrt. denial constraints, and consistency-based diagnosis.

More precisely, it is possible to obtain database repairs from causes, and the other way around. Then, the vast body of research on database repairs can be applied to the newer problem of determining actual causes for query answers. We unveil a strong connection between computing causes and their responsibilities for conjunctive queries, on the one hand, and computing both *subset-based* and *cardinality-based repairs* in databases [19] wrt. denial constraints, on the other hand. These computational problems can be reduced to each other. Some results about this connection are briefly presented in Section 2.

Furthermore, by formulating the causality problem as a diagnosis problem, it is possible to characterize causes in terms of the system's diagnoses. Actually, inferring and computing actual causes and responsibility in a database setting become, in different forms, consistency-based diagnosis reasoning problems and tasks. More specifically, a causal explanation for a conjunctive query answer can be viewed as a diagnosis, where in essence the relational database provides the system description, and the observation is the query answer. Some results are summarized in Section 3.

Abduction, as another approach to model-based diagnosis, can also be used to characterize causes for answers to queries, in particular Datalog queries. We present some results on this connection in Section 4.

We conclude this overview of mutual relationships between database causality and other areas of data management and knowledge representation by making some very general remarks, in Section 5, on database updates through views. We point out several connections with the above mentioned areas and problems.

## 2. CAUSES AND REPAIRS

We assume that a relational instance is split in two subinstances, say  $D = D^n \cup D^x$ , where  $D^n$  and  $D^x$  are formed by *endogenous* and *exogenous* tuples, respectively. The former tuples are possible candidate for causes. Now, given a boolean conjunctive  $\mathcal{Q}$ , a tuple  $t \in D^n$  is a *counterfactual cause* for  $\mathcal{Q}$  if  $D \models \mathcal{Q}$  and  $D \setminus \{t\} \not\models \mathcal{Q}$ . A tuple  $t \in D^n$  is an *actual cause* for  $\mathcal{Q}$ , if there exists  $\Gamma \subseteq D^n$ , called a *contingency set*, such that  $t$  is a counterfactual cause for  $\mathcal{Q}$  in  $D \setminus \Gamma$  [20].

The numerical function, *responsibility*, reflects the relative degree of causality of a tuple for a query result. More precisely, the *responsibility* of an actual cause  $t$  for  $\mathcal{Q}$ , denoted by  $\rho(t)$ , is the numerical value  $\frac{1}{(|\Gamma|+1)}$ , where  $|\Gamma|$  is the size of the smallest contingency set for  $t$  (with minimum cardinality). Tuples with higher responsibility are considered to provide more interesting explanations for query results [20, 21].

We will denote with  $\mathcal{CS}(D^n, D^x, \mathcal{Q})$  the set of actual causes for  $\mathcal{Q}$  (being true) in instance  $D = D^n \cup D^x$ . The following (anti)monotonicity results immediately hold.

**Proposition 1.** Let  $(D^n)', (D^x)'$  denote updates of  $D^n, D^x$

by insertion of tuple  $t$ , resp. It holds: (a)  $\mathcal{CS}(D^n, D^x, \mathcal{Q}) \subseteq \mathcal{CS}((D^n)', D^x, \mathcal{Q})$ . (b)  $\mathcal{CS}(D^n, (D^x)', \mathcal{Q}) \subseteq \mathcal{CS}(D^n, D^x, \mathcal{Q})$ .  $\square$

Actually, the set of actual causes may shrink by adding an exogenous tuple [24]. The possible loss of actual causes, which shows a non-monotonic behavior, is in line with the connections of causality with database repairs and model-based diagnosis, both of which have associated reasoning tasks that are also non-monotonic.

Now assume that the query is of the form  $\mathcal{Q}: \exists \bar{x}(P_1(\bar{x}_1) \wedge \dots \wedge P_m(\bar{x}_m))$ , with  $\bar{x} = \cup \bar{x}_i$ , and  $\mathcal{Q}$  is unexpectedly true in  $D$ , i.e we expected  $D \models \neg \mathcal{Q}$ . Then, we may want to trace back the causes for  $\mathcal{Q}$  to be true.

We first notice that  $\neg \mathcal{Q}$  is logically equivalent to the *denial constraint* (DC)  $\kappa(\mathcal{Q}): \forall \bar{x} \neg(P_1(\bar{x}_1) \wedge \dots \wedge P_m(\bar{x}_m))$  (that is also sometimes written as the Datalog constraint,  $\leftarrow P_1(\bar{x}_1), \dots, P_m(\bar{x}_m)$ ). If  $\mathcal{Q}$  is not expected to hold, we may consider  $D$  to be inconsistent wrt.  $\kappa(\mathcal{Q})$ . Repairs of  $D$  wrt.  $\kappa(\mathcal{Q})$  may be considered.

More precisely, we first consider the *S-repairs* (aka. *subset-repairs*), which are those consistent instances obtained from  $D$  via tuple insertion/deletion and make the symmetric difference with  $D$  minimal wrt. set inclusion. In the case of DCs, S-repairs are subsets of  $D$  that do not have any proper subset that is a repair [2]. Next, we consider the class containing the set differences between  $D$  and those *S-repairs* that do not contain tuple  $t \in D^n$ , and are obtained by removing a subset of  $D^n$ :

$$\mathcal{DF}(D, D^n, \kappa(\mathcal{Q}), t) = \{D \setminus D' \mid D' \in \text{Srep}(D, \kappa(\mathcal{Q})), t \in (D \setminus D') \subseteq D^n\}. \quad (1)$$

Here,  $\text{Srep}(D, \kappa(\mathcal{Q}))$  denotes the class of S-repairs of instance  $D$  wrt.  $\kappa(\mathcal{Q})$ .

**Proposition 2.** Given  $D = D^n \cup D^x$ , a BCQ  $\mathcal{Q}$ , and  $t \in D^n$ :  
(a)  $t$  is an actual cause for  $\mathcal{Q}$  iff  $\mathcal{DF}(D, D^n, \kappa(\mathcal{Q}), t) \neq \emptyset$ .  
(b) If  $\mathcal{DF}(D, D^n, \kappa(\mathcal{Q}), t) = \emptyset$ , then  $\rho(t) = 0$ .  
(c) If  $\rho(t) \neq 0$ ,  $\rho(t) = \frac{1}{|s|}$ , where  $s \in \mathcal{DF}(D, D^n, \kappa(\mathcal{Q}), t)$  and there is no  $s' \in \mathcal{DF}(D, D^n, \kappa(\mathcal{Q}), t)$  such that,  $|s'| < |s|$ .  $\square$

In the other direction, it is also possible to obtain repairs from actual causes. In fact, consider the database instance  $D$  and the denial constraint  $\kappa: \leftarrow A_1(\bar{x}_1), \dots, A_n(\bar{x}_n)$ . As usual, a boolean conjunctive *violation view*,  $V^\kappa: \exists \bar{x}(A_1(\bar{x}_1) \wedge \dots \wedge A_n(\bar{x}_n))$ , can be associated to  $\kappa$ .

Given an inconsistent instance  $D$  wrt.  $\kappa$ , we collect all S-minimal contingency sets associated with the actual cause  $t$  for  $V^\kappa$ , as follows:

$$\mathcal{CT}(D, D^n, V^\kappa, t) = \{s \subseteq D^n \mid D \setminus s \models V^\kappa, D \setminus (s \cup \{t\}) \not\models V^\kappa, \text{ and } \forall s'' \subsetneq s, D \setminus (s'' \cup \{t\}) \models V^\kappa\}.$$

**Proposition 3.** (a)  $D$  is consistent wrt.  $\kappa$  iff  $\mathcal{CS}(D, \emptyset, V^\kappa) = \emptyset$ . (b)  $D' \subseteq D$  is an S-repair for  $D$  iff, for every  $t \in D \setminus D'$ ,  $t \in \mathcal{CS}(D, \emptyset, V^\kappa)$  and  $D \setminus (D' \cup \{t\}) \in \mathcal{CT}(D, D, V^\kappa, t)$ .  $\square$

This proposition is stated for a single DC. However, it is possible to obtain repairs from causes for larger sets of DCs.

In [24] we provide a (naive) algorithm that computes the S-repairs for an inconsistent instance  $D$  and a set  $\Sigma$  of DCs from causes for their violation views being true.

Consider an instance  $D$ , and a DC  $\kappa$ . The following proposition establishes a relationship between *consistent query answering* (CQA) wrt. the S-repair semantics [2] and actual cases for the violation view  $V^\kappa$ .

**Proposition 4.** A ground atomic query  $A$  is consistently true, denoted  $D \models_S A$ , iff  $A \in D \setminus \mathcal{CS}(D, \emptyset, V^\kappa)$ .  $\square$

Along a similar line,  $C$ -repairs<sup>1</sup> are related to most responsible actual causes. We can collect the most responsible actual causes for  $V^\kappa$ :

$$\mathcal{MRC}(D, V^\kappa) = \{t \in D \mid t \in \mathcal{CS}(D, \emptyset, V^\kappa), \nexists t' \in \mathcal{CS}(D, \emptyset, V^\kappa) \text{ with } \rho(t') > \rho(t)\}.$$

**Proposition 5.** For an instance  $D$  and denial constraint  $\kappa$ ,  $D'$  is a C-repair for  $D$  wrt.  $\kappa$  iff for a  $t \in D \setminus D'$ :  $t \in \mathcal{MRC}(D, V^\kappa)$  and  $D \setminus (D' \cup \{t\}) \in \mathcal{CT}(D, V^\kappa, t)$ .  $\square$

In our ongoing research we have established and exploited a close connection (to be reported somewhere else) between *consistent answers* -under the cardinality-based repair semantics- to conjunctive queries wrt. denial constraints and most-responsible causes. These problems can be reduced to each other.

The partition of a database into endogenous and exogenous tuples has been exploited in the context of causality. However, this kind of partition is also of interest in the context of repairs. Considering that we should have more control on endogenous tuples than on exogenous ones, which may come from external sources, it makes sense to consider *endogenous repairs* that are obtained by updates (of any kind) on endogenous tuples. For example, in the case of violation of denial constraints, endogenous repairs would be obtained -if possible- by deleting endogenous tuples only.

If there are no repairs based on endogenous tuples only, a preference condition could be imposed on repairs, privileging those that change exogenous the least. (Of course, it could also be the other way around, i.e. we may feel more inclined to change exogenous tuples than our endogenous ones.)

Actually, we could go even further and apply notions of *preferred repairs* [25, 26]. If *Prep* denotes a given class of preferred repairs, it would be possible to explore the use of a relationship as the one in (1), replacing *Srep* by *Prep*, and by doing so, define other notions of causes, say tuples that are *Pcauses* for query answers. Considering other, e.g. preference based, forms of causality was mentioned as an interesting open direction in [21, 20].

As a further extension, it could be possible to assume that combinations of (only) exogenous tuples never violate the ICs, something that could be checked at upload time. In

<sup>1</sup>These are *cardinality-repairs*, and are defined as the S-repairs, but they minimize the cardinality of the symmetric difference. For DCs, they become maximal subsets, in cardinality, of the original instance [2].

this sense, there would be a part of the database that is considered to be consistent, while the other is subject to possible repairs. A situation like this has been considered, for other purposes and in a different form, in [14].

Actually, going a bit further, we could even consider the relations in the database with an extra, binary attribute,  $N$ , that is used to annotate if a tuple is endogenous or exogenous (it could be both), e.g. a tuple like  $R(a, b, yes)$ . ICs could be annotated too, e.g. the “exogenous” version of DC  $\kappa$ , could be  $\kappa^E: \leftarrow P(x, y, yes), R(y, z, yes)$ , and could be assumed to be satisfied.

### 3. CAUSES AND CONSISTENCY-BASED DIAGNOSIS

As above, let  $D = D^n \cup D^x$  be a database instance for schema  $\mathcal{S}$ , and  $\mathcal{Q}: \exists \bar{x}(P_1(\bar{x}_1) \wedge \dots \wedge P_m(\bar{x}_m))$  a BCQ. Assume that  $\mathcal{Q}$  is, possibly unexpectedly, true in  $D$ . That is, for the associated DC  $\kappa(\mathcal{Q}): \forall \bar{x} \neg(P_1(\bar{x}_1) \wedge \dots \wedge P_m(\bar{x}_m))$ , it holds  $D \not\models \kappa(\mathcal{Q})$ , i.e.  $D$  violates the DC. This becomes our *observation*, and we want to find causes for it, using a diagnosis-based approach, more precisely a *consistency-based diagnosis approach* [23].

We consider a *diagnosis problem*,  $\mathcal{M} = (SD, D^n, \mathcal{Q})$ , associated to  $\mathcal{Q}$ . Here,  $SD$  is a FO system description (or specification) containing the following elements:

- (a) Reiter’s logical reconstruction of  $D$  as a FO theory [22].
- (b) Sentence  $\kappa(\mathcal{Q})^{ext}$ , which is  $\kappa(\mathcal{Q})$  rewritten as follows:
$$\kappa(\mathcal{Q})^{ext}: \forall \bar{x} \neg(P_1(\bar{x}_1) \wedge \neg ab_{P_1}(\bar{x}_1) \wedge \dots \wedge P_m(\bar{x}_m) \wedge \neg ab_{P_m}(\bar{x}_m)).$$
- (c) The sentence  $\neg \kappa(\mathcal{Q}) \longleftrightarrow \mathcal{Q}$ , where  $\mathcal{Q}$  is the initial boolean query.
- (d) The inclusion dependencies:  $\forall \bar{x}(ab_P(\bar{x}) \rightarrow P(\bar{x}))$ .

Here, predicate  $ab$  stands, as usual, for *abnormality*. Then, the intended meaning of  $\kappa(\mathcal{Q})^{ext}$  is that *under normal conditions on the tuples, the DC is satisfied*.

Now, the last entry,  $\mathcal{Q}$ , in  $\mathcal{M}$  is the *observation* (or the fact that it is true), which together with the system description, *plus the assumption that all tuples are normal* (i.e. not abnormal), produces and inconsistent theory. Consequently, a *diagnosis* for the diagnosis problem  $\mathcal{M}$  is a  $\Delta \subseteq D^n$ , such that  $SD \cup \{ab_P(\bar{c}) \mid P(\bar{c}) \in \Delta\} \cup \{\neg ab_P(\bar{c}) \mid P(\bar{c}) \in D \setminus \Delta\} \cup \{\mathcal{Q}\}$  becomes consistent.

With  $\mathcal{D}(\mathcal{M}, t)$  we denote the set of all subset-minimal diagnoses for  $\mathcal{M}$  that contain tuple  $t \in D^n$ . Similarly,  $\mathcal{MCD}(\mathcal{M}, t)$  denotes the set of diagnoses of  $\mathcal{M}$  that contain tuple  $t \in D^n$  and have the minimum cardinality (among those diagnoses that contain  $t$ ). Clearly  $\mathcal{MCD}(\mathcal{M}, t) \subseteq \mathcal{D}(\mathcal{M}, t)$ .

**Proposition 6.** (a) Tuple  $t \in D^n$  is an actual cause for  $\mathcal{Q}$  iff  $\mathcal{D}(\mathcal{M}, t) \neq \emptyset$ .

(b) For tuple  $t \in D^n$ ,  $\rho(t) = 0$  iff  $\mathcal{MCD}(\mathcal{M}, t) = \emptyset$ . Otherwise,  $\rho(t) = \frac{1}{|s|}$ , where  $s \in \mathcal{MCD}(\mathcal{M}, t)$ .  $\square$

Taking advantage of results and techniques for database repairs and *consistency-based diagnosis* through hitting sets,

as done in [23], it is possible to extend complexity results reported in [21] for the causality and responsibility problems for conjunctive queries. This is particularly the case of the problem of deciding whether a tuple is a most responsible cause for a query answer.

## 4. CAUSES AND ABDUCTION

Causality in databases (and everywhere) can be seen as a very fundamental concept to which many other data management notions are connected. Some of them have been mentioned above, and there are others. We envision a broad, common framework in which these rich connections can be formulated and investigated, contributing to shed light on each of the areas involved, and most importantly, to take advantage of each them for theoretical and computational purposes in relation to the others.

Still from the model-based diagnosis point of view, but this time appealing to *abductive diagnosis* [7, 10], it is possible to extend and formulate the notion of query-answer causality for Datalog queries via abductive diagnosis from Datalog specifications. So, the connection between (query-answer) causality and abduction via Datalog makes it possible to go beyond conjunctive queries (the case considered in [21]),<sup>2</sup> extending causality to, e.g. recursive queries, and obtaining new results for them. Notice that consistency-based diagnosis is usually practiced with first-order (FO) specifications, but abductive reasoning is commonly performed under logic programming approaches [9, 11].

A *Datalog abduction problem* (DAP) [11] is of the form  $\mathcal{AP} = \langle \Pi, EDB, Hyp, Obs \rangle$ , where: (a) *EDB* is an input structure (a set of ground atoms), (b)  $\Pi$  is a set of Datalog rules, (c) *Hyp* (the hypothesis) and *Obs* (the observations) are finite sets of ground atoms with  $\Pi \cup EDB \cup Hyp \models Obs$ .<sup>3</sup> The elements of *Hyp* are the *abducible atoms* (or simply, *abducibles*), and they, in combinations, should explain the observations. An *abductive diagnosis* (or simply, *a solution*) for  $\mathcal{AP}$  is a minimal subset  $\Delta \subseteq Hyp$  (wrt. subset minimality), such that  $\Pi \cup EDB \cup \Delta \models Obs$ . We denote with  $Sol(\mathcal{AP})$  the set of abductive diagnoses for problem  $\mathcal{AP}$ .

The *relevance problem* is a decision problem that naturally arises in abduction: Given  $\mathcal{AP} = \langle \Pi, EDB, Hyp, Obs \rangle$ , and a ground fact  $h \in Hyp$ , determine whether  $h$  is *relevant* in  $\mathcal{AP}$ , i.e.  $h$  occurs in an abductive diagnosis of  $\mathcal{AP}$ , denoted  $h \in Rel(\mathcal{AP})$ .

Now, assume we are given a relational instance with  $D = D^x \cup D^n$ , and a Datalog program  $\Pi$  that represents a boolean, possibly recursive query. Then,  $\Pi$  has a highest level zero-ary predicate *ans* that returns the result (or not). If  $\Pi \cup D \models ans$ , we want to find actual causes and their responsibility degrees for *ans*. It holds that actual causes for *ans*<sup>4</sup> can be obtained from abductive diagnosis of the associated DAP

<sup>2</sup>In [20] the authors apply the definition of cause as above to monotone queries.

<sup>3</sup>We will assume that no predicate in a rule head of  $\Pi$  appears in  $EDB \cup Hyp$ .

<sup>4</sup>That can be defined by means of an extension of the definition of actual causes for conjunctive queries as given in Section 2.

$\mathcal{AP}^c := \langle \Pi, D^x, D^n, \{ans\} \rangle$ , where *ans* is the observation,  $\Pi \cup D^x$  is the *background theory*, and  $D^n$  is the set of *hypothesis*. More precisely, it holds: tuple  $t \in D^n$  is an actual cause for *ans* iff  $t \in Rel(\mathcal{AP}^c)$ .

Now, in order to obtain *responsibilities*, consider again  $\mathcal{AP}^c = \langle \Pi, D^x, D^n, \{ans\} \rangle$ , with  $Sol(\mathcal{AP}^c) \neq \emptyset$ .  $N \subseteq D^n$  is a *set of necessary hypothesis* if  $N$  is minimal (wrt. set inclusion), such that  $Sol(\mathcal{AP}_N^c) = \emptyset$ , where  $\mathcal{AP}_N^c = \langle \Pi, D^x, D^n \setminus N, \{ans\} \rangle$ . It holds: The responsibility of a tuple  $t$  for *ans* is  $\frac{1}{|N|}$ , where  $N$  is a necessary hypothesis set with minimum cardinality, such that  $t \in N$ .

The notion of necessary hypothesis set we just introduced extends the notion of (single) necessary hypothesis that has been studied in abduction [10, 11]. The extended notion not only captures responsibility as defined in [21] for Datalog queries, but it is also interesting in the context of abduction *per se*. In particular, it provides a sort of quantitative metric to rank relevant hypothesis according to their *degree of necessity*, which can be captured in terms of the sizes of necessary hypothesis sets to which they belong. In particular, the smaller the size of a necessary hypothesis set, the the more necessary are its elements to be included among the relevant hypothesis that explain the observation. We believe that the degree of necessity in abduction captures the notion of responsibility as introduced in the context of causality [6].

**Example 1.** Consider an instance  $D$  with predicates  $R$  and  $S$  as below, and the query  $\Pi: ans \leftarrow R(x, y), S(y)$ .

$R$	$X$	$Y$	$S$	$X$
	$a_1$	$a_4$		$a_1$
	$a_2$	$a_1$		$a_2$
	$a_3$	$a_3$		$a_3$

The DAP  $\mathcal{AP}^c = \langle \Pi, \emptyset, D, \{ans\} \rangle$  (all the tuples are endogenous) has two abductive diagnosis:  $\Delta_1 = \{S(a_1), R(a_2, a_1)\}$  and  $\Delta_2 = \{S(a_3), R(a_3, a_3)\}$ . Then,  $Rel(\mathcal{AP}^c) = \{S(a_3), R(a_3, a_3), S(a_1), R(a_3, a_3)\}$ . It is easy to verify that the relevant hypothesis are the actual causes for *ans*.

The necessary hypothesis sets of  $\mathcal{AP}^c$  all have cardinality 2. Thus, the responsibility of each actual cause is  $\frac{1}{2}$ .  $\square$

In [21], complexity and algorithmic results for query-answer causality apply to conjunctive queries. Now they can be extended by applying results and techniques for abductive reasoning, as those obtained in [11, 13]. These extensions are a matter of ongoing research.

## 5. ABDUCTION, VIEW UPDATES AND REPAIRS

Another direction to explore for fruitful connections with all the above turns around the *view update problem*, which is about updating a database through views. The problem is about minimally changing the underlying database (i.e. the base relations), in such a way that the changed base instance produces the intended changes in the view contents. Put in other terms, it is an update propagation problem, from views to base relations. This old and important problem in databases.

Since intended changes on the views may create alternative, admissible candidates to be the updated underlying database. As a consequence, user knowledge imposed through view updates creates or reflects *uncertainty* about the base data.

The view update problem is related to all the problems mentioned above. We mention some connections without going into details. First of all, the view update problem has been treated from the point of view of abductive reasoning [16, 8]. The idea is to “abduce” changes on base tables that explain the intended changes on the views.

The view update problem, specially in its particular form of of *deletion propagation*<sup>5</sup>, has been recently related in [17, 18] to causality as introduced in [21].

We also should mention that database repairs are related to the view update problem. Actually, *answer set programs* (ASPs) for database repairs (c.f. [5] for a comprehensive account and references) implicitly repair the database by updating conjunctive combinations of intentional, annotated predicates. Those logical combinations -views after all- capture violations of integrity constraints in the original database or along the (implicitly iterative) repair process (hence the need for annotations).

Even more, in [3], in order to protect sensitive information, databases are explicitly and virtually “repaired” through secrecy views that specify the information that has to be kept secret. In order to protect information, a user is allowed to interact only with the virtually repaired versions of the original database that result from making those views empty or contain only null values. Repairs are specified and computed using ASP, and an explicit connection to prioritized attribute-based repairs [2] is made [3].

Finally, we should note that abduction has also been explicitly applied to database repairs [1]. The idea, again, is to “abduce” possible repair updates that bring the database to a consistent state.

The areas of causality for query answers, database repairs and consistent query answering, model-based diagnosis (in its consistency-based and abductive versions),<sup>6</sup> and database updates through views will all benefit from a deeper investigations of their mutual relationships, for better understanding them, and taking advantage of known results for some of them to obtain new results for the others.

**Acknowledgments:** Research funded by NSERC Discovery, and the NSERC Strategic Network on Business Intelligence (BIN). Conversations on causality in databases with Alexandra Meliou during Leo Bertossi’s visit to U. of Washington in 2011 are much appreciated. He is also grateful to Dan Suciu and Wolfgang Gatterbauer for their hospitality.

<sup>5</sup>This is the relevant case when the views are conjunctive and capture violations of denial constraints: the violations are solved by tuple deletion, from views and base tables.

<sup>6</sup>Consistency-based diagnosis and abduction-based diagnosis are not independent approaches. Connections between them have been established [7].

Leo Bertossi also appreciates stimulating conversations with Benny Kimelfeld during his two-month stay at LogicBlox, and the hospitality of the LogicBlox team.

## 6. REFERENCES

- [1] Arieli, O., Denecker, M., Van Nuffelen, B. and Bruynooghe, M. Coherent Integration of Databases by Abductive Logic Programming. *J. Artif. Intell. Res.*, 2004, 21:245-286.
- [2] Bertossi, L. *Database Repairing and Consistent Query Answering*. Morgan & Claypool, Synthesis Lectures on Data Management, 2011.
- [3] Bertossi, L. and Li, L. Achieving Data Privacy through Secrecy Views and Null-Based Virtual Updates. *IEEE TKDE*, 2013, 25(5):987-1000.
- [4] Borgida, A., Calvanese, D. and Rodriguez-Muro, M. Explanation in DL-Lite. *Proc. DL Workshop*, CEUR-WS 353, 2008.
- [5] Caniupan, M. and Bertossi, L. The Consistency Extractor System: Answer Set Programs for Consistent Query Answering in Databases. *Data & Knowledge Engineering*, 2010, 69(6):545-572.
- [6] Chockler, H. and Halpern, J. Y. Responsibility and Blame: A Structural-Model Approach. *J. Artif. Intell. Res.*, 2004, 22:93-115.
- [7] Console, L., Theseider-Dupre, D. and Torasso, P. On the Relationship between Abduction and Deduction. *J. Log. Comput.*, 1991, 1(5):661-690.
- [8] Console, L., Sapino M. L., Theseider-Dupre, D. The Role of Abduction in Database View Updating. *J. Intell. Inf. Syst.*, 1995, 4(3): 261-280.
- [9] Denecker, M., and Kakas A. C. Abduction in Logic Programming. In *Computational Logic: Logic Programming and Beyond*, 2002, LNCS 2407.
- [10] Eiter, T. and Gottlob, G. The Complexity of Logic-Based Abduction. *J. ACM*, 1995, 42(1): 3-42.
- [11] Eiter, T., Gottlob, G. and Leone, N. Abduction from logic programs: Semantics and complexity. *Theor. Comput. Sci.*, 1997, 189(1-2):129-177.
- [12] Gertz, M. Diagnosis and Repair of Constraint Violations in Database Systems. PhD Thesis, Universität Hannover, 1996.
- [13] Gottlob, G., Pichler, R. and Wei, F. Efficient Datalog Abduction through Bounded Treewidth. *Proc. AAAI*, 2007.
- [14] Greco, S., Pijcke, F. and Wijsen, J. Certain Query Answering in Partially Consistent Databases. *PVLDB*, 2014, 7(5):353-364.
- [15] Halpern, Y. J., Pearl, J. Causes and Explanations: A Structural-Model Approach: Part 1. *British J. Philosophy of Science*, 2005, 56:843-887.
- [16] Kakas A. C. and Mancarella, P. Database Updates through Abduction. *Proc. VLDB*, 1990, pp. 650-661.
- [17] Kimelfeld, B. A Dichotomy in the Complexity of Deletion Propagation with Functional Dependencies. *Proc. PODS*, 2012.
- [18] Kimelfeld, B., Vondrak, J. and Williams, R. Maximizing Conjunctive Views in Deletion Propagation. *ACM TODS*, 2012, 7(4):24.
- [19] Lopatenko, A. and Bertossi, L. Complexity of Consistent Query Answering in Databases under Cardinality-Based and Incremental Repair Semantics. *Proc. ICDT*, 2007, LNCS 4353.
- [20] Meliou, A., Gatterbauer, W., Halpern, J. Y., Koch, C., Moore K. F. and Suciu, D. Causality in Databases. *IEEE Data Eng. Bull.*, 2010, 33(3):59-67.
- [21] Meliou, A., Gatterbauer, W., Moore, K. F. and Suciu, D. The Complexity of Causality and Responsibility for Query Answers and Non-Answers. *PVLDB*, 2010, 4(1):34-45.
- [22] Reiter, R. Towards a Logical Reconstruction of Relational Database Theory. In *On Conceptual Modelling*, M.L.

- Brodie, J. Mylopoulos and J.W. Schmidt (eds.), Springer, 1984, pp. 191-233.
- [23] Reiter, R. A Theory of Diagnosis from First Principles. *Artificial Intelligence*, 1987, 32(1):57-95.
  - [24] Salimi, B. and Bertossi, L. Causality in Databases: The Diagnosis and Repair Connections. To appear in Proc. 15th International Workshop on Non-Monotonic Reasoning (NMR 2014). Corr Arkiv Paper cs.DB/1404.6857, 2014.
  - [25] Staworko, S., Chomicki, J. and Marcinkowski, J. Prioritized Repairing and Consistent Query Answering in Relational Databases. *Ann. Math. Artif. Intell.*, 2012, 64(2-3):209-246.
  - [26] Yakout, M., Elmagarmid, A., Neville, J., Ouzzani, M. and Ilyas, I. Guided Data Repair. *PVLDB*, 2011, 4(5):279-289.