Causality in Databases: Answer-Set Programs and Integrity Constraints

Leopoldo Bertossi[⋆]

Carleton University, School of Computer Science, Ottawa, Canada.

Causality in databases (DBs) was introduced in [15]; and, building on work on causality as found in artificial intelligence, appeals to the notions of counterfactuals, interventions and structural models [14]. More specifically, [15] introduces the notions of: (a) a DB tuple as an *actual cause* for a query result, (b) a *contingency set* for a cause, as a set of tuples that must accompany the cause for it to be such, and (c) the *responsibility* of a cause as a numerical measure of its strength [11].

In our research on causality in DBs we have attempted to understand causality from different angles of data and knowledge management. In [5], precise reductions between causality in DBs, DB repairs, and consistency-based diagnosis were established; and the relationships were investigated and exploited. In [6], causality in DBs was related to view-based DB updates and abductive diagnosis, establishing fruitful connections.

This work summarizes some directions and results of our recent and ongoing research in DB causality.

Causality in databases. A notion of *cause* as an explanation for a query result was introduced in [15], as follows. For a relational instance D, a tuple $\tau \in D^n$ is called a *counterfactual cause* for a Boolean conjunctive query (BCQ) $\mathcal{Q}$, if $D \models \mathcal{Q}$ and $D \setminus \{\tau\} \not\models \mathcal{Q}$. Now, $\tau \in D$ is an *actual cause* for $\mathcal{Q}$ if there exists $\Gamma \subseteq D$, called a *contingency set* for τ, such that τ is a counterfactual cause for $\mathcal{Q}$ in $D \setminus \Gamma$.

The notion of *responsibility* reflects the relative degree of causality of a tuple for a query result [15] (based on [11]). The responsibility of an actual cause τ for $\mathcal{Q}$, is $\rho(\tau) := \frac{1}{|\Gamma|+1}$, where $|\Gamma|$ is the size of a smallest contingency set for τ. If τ is not an actual cause, $\rho(\tau) := 0$. Tuples with higher responsibility are stronger explanations.

The notion of cause can be applied to any monotonic query with free variables, e.g. conjunctive queries (CQs) with built-ins, unions of CQs (UCQs) [5], Datalog queries [6], etc. Here we consider only conjunctive queries.

Example 1. Consider the relational DB $D = \{R(a_4, a_3), R(a_2, a_1), R(a_3, a_3), S(a_4), S(a_2), S(a_3)\}$, and the query $\mathcal{Q}$: $\exists x \exists y(S(x) \wedge R(x, y) \wedge S(y))$. It holds, $D \models \mathcal{Q}$.

$S(a_3)$ is a counterfactual cause for $\mathcal{Q}$: if $S(a_3)$ is removed from D, $\mathcal{Q}$ is no longer true. Its responsibility is 1. So, it is an actual cause with empty contingency set. $R(a_4, a_3)$ is an actual cause for $\mathcal{Q}$ with contingency set $\{R(a_3, a_3)\}$: if $R(a_3, a_3)$ is removed from D, $\mathcal{Q}$ is still true, but further removing $R(a_4, a_3)$ makes $\mathcal{Q}$ false. The responsibility of $R(a_4, a_3)$ is $\frac{1}{2}$. $R(a_3, a_3)$ and $S(a_4)$ are actual causes, with responsibility $\frac{1}{2}$. □

Database repairs and causes. We introduce the main ideas around DB repairs [2] by means of an example. The ICs that we consider here can be enforced only by tuple-deletions from the DB (as opposed to tuple-insertions).

Example 2. The DB $D = \{P(a), P(e), Q(a, b), R(a, c)\}$ is inconsistent with respect to the (set of) *denial constraints* (DCs) κ_1 : $\neg\exists x \exists y(P(x) \wedge Q(x, y))$, and κ_2 : $\neg\exists x \exists y(P(x) \wedge R(x, y))$. It holds $D \not\models \{\kappa_1, \kappa_2\}$.

⋆ Member of the "Millenium Institute for Foundational Research on Data", Chile. bertossi@scs.carleton.ca. Supported by NSERC Discovery Grant #06148.

A *subset-repair*, in short an *S-repair*, of D wrt. the set of DCs is a $\subseteq$-maximal subset of D that is consistent, i.e. no proper superset is consistent. The following are S-repairs: $D_1 = \{P(e), Q(a,b), R(a,b)\}$ and $D_2 = \{P(e), P(a)\}$. A *cardinality-repair*, in short a *C-repair*, of D wrt. the set of DCs is a maximum-cardinality, consistent subset of D, i.e. no subset of D with larger cardinality is consistent. D_1 is the only C-repair. □

For an instance D and a set Σ of DCs, the sets of S-repairs and C-repairs are denoted with $\mathit{Srep}(D, \Sigma)$ and $\mathit{Crep}(D, \Sigma)$, resp.

Now, consider a BCQ $\mathcal{Q}\colon \exists \bar{x}(P_1(\bar{x}_1) \wedge \cdots \wedge P_m(\bar{x}_m))$ with $D \models \mathcal{Q}$. Notice that $\neg\mathcal{Q}$ is logically equivalent to the DC: $\kappa(\mathcal{Q})\colon\ \neg\exists \bar{x}(P_1(\bar{x}_1) \wedge \cdots \wedge P_m(\bar{x}_m))$. So, if $\mathcal{Q}$ is true in D, D is inconsistent wrt. $\kappa(\mathcal{Q})$, giving rise to repairs of D wrt. $\kappa(\mathcal{Q})$. In [5] it was shown that S-repairs can be used to obtain actual causes with their contingency sets; and C-repairs for causes' responsibilities, which we show with an example. First we need to build differences, containing a tuple τ, between D and S- or C-repairs:

$$\text{(a)}\ \ \mathit{Diff}^s(D, \kappa(\mathcal{Q}), \tau) \ = \ \{D \smallsetminus D' \mid D' \in \mathit{Srep}(D, \kappa(\mathcal{Q})),\ \tau \in (D \smallsetminus D')\}, \quad (1)$$

$$\text{(b)}\ \ \mathit{Diff}^c(D, \kappa(\mathcal{Q}), \tau) \ = \ \{D \smallsetminus D' \mid D' \in \mathit{Crep}(D, \kappa(\mathcal{Q})),\ \tau \in (D \smallsetminus D')\}. \quad (2)$$

Example 3. (ex. 1 cont.) With the same instance D and query $\mathcal{Q}$, we consider the DC $\kappa(\mathcal{Q})$: $\neg\exists x \exists y(S(x) \wedge R(x,y) \wedge S(y))$, which is not satisfied by D. Here, $\mathit{Srep}(D, \kappa(\mathcal{Q})) = \{D_1, D_2, D_3\}$ and $\mathit{Crep}(D, \kappa(\mathcal{Q})) = \{D_1\}$, with $D_1 = \{R(a_4, a_3), R(a_2, a_1), R(a_3, a_3), S(a_4), S(a_2)\}$, $D_2 = \{R(a_2, a_1), S(a_4), S(a_2), S(a_3)\}$, $D_3 = \{R(a_4, a_3), R(a_2, a_1), S(a_2), S(a_3)\}$.

For tuple $R(a_4, a_3)$, $\mathit{Diff}^s(D, \kappa(\mathcal{Q}), R(a_4, a_3)) = \{D \smallsetminus D_2\} = \{\{R(a_4, a_3), R(a_3, a_3)\}\}$. So, $R(a_4, a_3)$ is an actual cause, with responsibility $\frac{1}{2}$. Similarly, $R(a_3, a_3)$ is an actual cause, with responsibility $\frac{1}{2}$. For tuple $S(a_3)$, $\mathit{Diff}^c(D, \kappa(\mathcal{Q}), S(a_3)) = \{D \smallsetminus D_1\} = \{S(a_3)\}$. So, $S(a_3)$ is an actual cause, with responsibility 1. □

Specification of causes with ASP. A DB may have several repairs wrt. a class of ICs. Since we are mainly interested in reasoning with whole class of them, e.g. for consistent query answering [2], it is natural to try to specify this class in logical terms. This can be achieved by means of *answer-set programs* (*disjunctive logic programs with stable model semantics*) [1], the so-called *repair-programs* [9, 2]. The consistent answers to a query are certainly and logically entailed by the program. As shown in [7, 8], the reduction of DB causality to DB repairs illustrated above can be used to take advantage of repair programs for providing specifications in answer-set programming (ASP) to reason about causes and compute causes, their contingency sets, and responsibility degrees. The resulting *causality-programs* have the necessary and sufficient expressive power to capture and compute not only causes, which can be done with less expressive programs [15], but also minimal contingency sets and responsibilities (which can not). Actually, *weak program constraints* [1] are used to specify C-repairs.

Repairing the DB by changing attribute values is also possible [3, 4]. In [8, 5], a particular notion of attribute-based repair was used to define *attribute values* (in tuples) as causes for query answers. In [8] it is shown how to specify attribute-based causes via ASP via the attribute-based repairs connection.

Causality under ICs. The *interventions* at the basis of [14], i.e. actions on the structural model that determine counterfactual scenarios, take in DBs the form of tuple deletions. If a DB D is expected to satisfy a given set of ICs, they should also be considered

as a part of the model. Accordingly, the instances obtained from D by tuple deletions, as used to determine causes, should also satisfy the ICs.

Example 4. For the DB instance D and the open, i.e. non-Boolean, CQ, $\mathcal{Q}$ in (3).

	Dep	*DName*	*TStaff*
t_1		Computing	John
t_2		Philosophy	Patrick
t_3		Math	Kevin

	Course	*CName*	*TStaff*	*DName*
t_4		COM08	John	Computing
t_5		Math01	Kevin	Math
t_6		HIST02	Patrick	Philosophy
t_7		Math08	Eli	Math
t_8		COM01	John	Computing

$$Ans_{\mathcal{Q}}(\mathit{TStaff}) \leftarrow Dep(DName, \mathit{TStaff}), Course(\mathit{CName}, \mathit{TStaff}, DName). \quad (3)$$

$\mathcal{Q}(D) = \{\mathsf{John}, \mathsf{Patrick}, \mathsf{Kevin}\}$, and $\langle \mathsf{John} \rangle$ has the actual causes: t_1, t_4 and t_8. t_1 is a counterfactual cause, t_4 has a single minimal contingency set $\Gamma_1 = \{t_8\}$; and t_8 has a single minimal contingency set $\Gamma_2 = \{t_4\}$. Now, for the query

$$Ans_{\mathcal{Q}'}(\mathit{TStaff}) \leftarrow Dep(DName, \mathit{TStaff})), \quad (4)$$

$\langle \mathsf{John} \rangle$ is still an answer from D, but it has a single cause, t_1, which is also a counterfactual cause. Now, if the following inclusion dependency is satisfied by D,

$$\psi\colon \ \forall x \forall y \, (Dep(x, y) \rightarrow \exists u \, Course(u, y, x)), \quad (5)$$

$\mathcal{Q}$ is equivalent to $\mathcal{Q}'$. The question is whether t_4 and t_8 should still be considered as causes for answer $\langle \mathsf{John} \rangle$ in the presence of ψ. Finally, consider the query $\mathcal{Q}_1$:

$$Ans_{\mathcal{Q}_1}(\mathit{TStaff}) \leftarrow Course(\mathit{CName}, \mathit{TStaff}, DName), \quad (6)$$

which, among others, has $\langle \mathsf{John} \rangle$ as an answer, with t_4 and t_8 as the only actual causes, with contingency sets $\Gamma_1 = \{t_8\}$ and $\Gamma_2 = \{t_4\}$, resp. In the presence of ψ, one should wonder if also t_1 would be a cause (it contains the referring value John in table $Dept$), or, if not, whether its presence would make the previous causes less responsible. □

A definition of query-answer cause was introduced and investigated in [6, sec. 7], as follows. For an instance D that satisfies a set Σ of ICs, i.e. $D \models \Sigma$, and a monotone query $\mathcal{Q}$ with $D \models \mathcal{Q}(\bar{a})$, a tuple $\tau \in D$ is an *actual cause for $\bar{a}$ under Σ* if there is $\Gamma \subseteq D$, such that: (a) $D \smallsetminus \Gamma \models \mathcal{Q}(\bar{a})$; (b) $D \smallsetminus \Gamma \models \Sigma$; (c) $D \smallsetminus (\Gamma \cup \{t\}) \not\models \mathcal{Q}(\bar{a})$; and (d) $D \smallsetminus (\Gamma \cup \{t\}) \models \Sigma$. The *responsibility* of τ as a cause for an answer $\bar{a}$ to query $\mathcal{Q}$ under a set Σ of ICs, denoted by $\rho^{D,\Sigma}_{\mathcal{Q}(\bar{a})}(\tau)$, is defined exactly as above.

Example 5. (ex. 4 cont.) For query $\mathcal{Q}$ in (3) and its answer $\langle \mathsf{John} \rangle$, without ψ in (5), t_4 was a cause with minimal contingency set $\Gamma_1 = \{t_8\}$. Now, it holds $D \smallsetminus \Gamma_1 \models \psi$, but $D \smallsetminus (\Gamma_1 \cup \{t_4\}) \not\models \psi$. So, in presence of ψ, t_4 is not an actual cause for $\langle \mathsf{John} \rangle$. Notice that $\mathcal{Q}$ and $\mathcal{Q}'$ in (4) have the same actual cause t_1 for $\langle \mathsf{John} \rangle$ under ψ.

For $\mathcal{Q}_1$ in (6), and its answer $\langle \mathsf{John} \rangle$, t_4 and t_8 are still (non-counterfactual) actual causes under ψ. However, their previous contingency sets are not such anymore: $D \smallsetminus (\Gamma_1 \cup \{t_4\}) \not\models \psi$, $D \smallsetminus (\Gamma_2 \cup \{t_8\}) \not\models \psi$. Actually, the smallest contingency set for t_4 is $\Gamma_3 = \{t_8, t_1\}$; and for t_8, $\Gamma_4 = \{t_4, t_1\}$. Accordingly, the causal responsibilities of t_4, t_8 decrease under ψ: $\rho^{D}_{\mathcal{Q}(\mathsf{John})}(t_4) = \frac{1}{2}$, but $\rho^{D,\psi}_{\mathcal{Q}(\mathsf{John})}(t_4) = \frac{1}{3}$. Under ψ, tuple t_1 is still not an actual cause for answer $\langle \mathsf{John} \rangle$ to $\mathcal{Q}_1$. □

Since *denial constraints* are never violated by tuple deletions, they do not affect on the causes for a query answer. However, inclusion dependencies may make a set of causes grow, together with sizes of minimal contingency sets, and then, responsibilities decrease. Intuitively, the responsibility is spread out through inclusion dependencies.

Also, causes are preserved under logically equivalent query rewriting under ICs. (Cf. [6, prop. 20] for general properties.) For some classes of CQs, the complexity of deciding responsibility of causes (under the usual notion of causality) may decrease from NP-complete to tractability when the DB satisfies certain key constraints [12]. On the other side, without ICs, deciding causality for CQs is tractable [15], but their presence may make complexity grow: There are a CQ $\mathcal{Q}$ and an inclusion dependency ψ, for which deciding causality is NP-complete [6, prop. 22].

In [6] also an abductive/deductive methodology for computing causes for answers to Datalog queries under ICs was proposed. The involved logical rewritings are reminiscent of those used for *semantic query optimization*.

As part of our ongoing research, as an extension of [8], we are appealing to ASPs to specify causes under ICs. This can be achieved by taking advantage of repair programs for DBs that may be inconsistent wrt. soft ICs, but consistent wrt. hard ICs. (CQA for this kind of DBs is investigated in [13].)

References

[1] Baral, C. *Knowledge Representation, Reasoning and Declarative Problem Solving*. Cambridge Univ. Press, 2003.

[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 Trans. Knowledge and Data Engineering*, 2013, 25(5):987-1000.

[4] Bertossi, L. and Bravo, L. Consistency and Trust in Peer Data Exchange Systems. *Theory and Practice of Logic Programming*, 2017, 17(2):148-204.

[5] Bertossi, L. and Salimi, B. From Causes for Database Queries to Repairs and Model-Based Diagnosis and Back. *Theory of Computing Systems*, 2017, 61(1):191232.

[6] Bertossi, L. and Salimi, B. Causes for Query Answers from Databases: Datalog Abduction, View-Updates, and Integrity Constraints. *Int. J. Approximate Reasoning*, 2017, 90:226-252. Corr Arxiv Paper cs.DB/1611.01711.

[7] Bertossi, L. The Causality/Repair Connection in Databases: Causality-Programs. Proc. Scalable Uncertainty Management (SUM'17). Springer LNCS 10564, 2017

[8] Bertossi, L. Characterizing and Computing Causes for Query Answers in Databases from Database Repairs and Repair Programs. Corr Arxiv Paper cs.DB/1712.01001, 2017. To appear in Proc. FoIKS'18.

[9] Caniupan-Marileo, M. and Bertossi, L. The Consistency Extractor System: Answer Set Programs for Consistent Query Answering in Databases". *Data & Know. Eng.*, 2010, 69(6):545-572.

[10] Chakravarthy, U. S., Grant, J. and Minker, J. Logic-Based Approach to Semantic Query Optimization. *ACM TODS*, 1990, 15(2):162-207.

[11] Chockler, H. and Halpern, J. Y. Responsibility and Blame: A Structural-Model Approach. *J. Artif. Intell. Res.*, 2004, 22:93-115.

[12] Cibele, F., Gatterbauer, W., Immerman, N. and Meliou A. A Characterization of the Complexity of Resilience and Responsibility for Conjunctive Queries. *PVLDB*, 2015, 9(3):180-191.

[13] Greco, S., Pijcke, F. and Wijsen, J. Certain Query Answering in Partially Consistent Databases. *PVLDB*, 2014, 7(5):353-364.

[14] Halpern, J. and Pearl, J. Causes and Explanations: A Structural-Model Approach: Part 1. *British J. Philosophy of Science*, 2005, 56:843-887.

[15] Meliou, A., Gatterbauer, W., Moore, K. F. and Suciu, D. The Complexity of Causality and Responsibility for Query Answers and Non-Answers. Proc. VLDB, 2010, pp. 34-41.