



Query Repairs

Balder ten Cate ✉ 

University of Amsterdam, The Netherlands

Phokion G. Kolaitis ✉ 

University of California Santa Cruz, CA, USA

IBM Research – Almaden, San Jose, CA, USA

Carsten Lutz ✉ 

Leipzig University, Germany

Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Dresden/Leipzig, Germany

Abstract

We formalize and study the problem of repairing database queries based on user feedback in the form of a collection of labeled examples. We propose a framework based on the notion of a proximity pre-order, and we investigate and compare query repairs for conjunctive queries (CQs) using different such pre-orders. The proximity pre-orders we consider are based on query containment and on distance metrics for CQs.

2012 ACM Subject Classification Information systems → Query languages

Keywords and phrases Query Repairs, Databases, Conjunctive Queries, Data Examples, Fitting

Digital Object Identifier 10.4230/LIPIcs.ICDT.2025.15

Related Version *Full Version:* <https://arxiv.org/abs/2501.11162> [27]

Funding *Balder ten Cate:* Supported by EU Horizon 2020 Grant MSCA-101031081.

Phokion G. Kolaitis: Partially supported by NSF Grant IIS-1814152.

Carsten Lutz: Supported by the DFG Collaborative Research Center 1320 EASE.

1 Introduction

When querying a database, it may happen that the query result includes some undesired tuples and/or that some desired tuples are missing. In such cases, it is often necessary to adjust the query to ensure that the result aligns with expectations, i.e., it includes the desired tuples and omits the undesired ones.

► **Example 1.1.** Consider a database instance I in Figure 1. A user, wanting to retrieve movies released in both Germany and France, issues the query $q(x) :- \text{Release}(x, y, \text{FR}), \text{Release}(x, y, \text{DE})$. The query results include Babygirl but not Nosferatu. The user spots the latter as a missing answer, and wants to revise the query. A solution is to change the query to $q'(x) :- \text{Release}(x, y, \text{FR}), \text{Release}(x, z, \text{DE})$. A more radically different query such as $q''(x) :- \text{Release}(x, y, \text{FR})$ would also account for the missing answer, but clearly fails to capture the user’s intention.

Release		
Babygirl	2025	DE
Babygirl	2025	FR
Nosferatu	2025	DE
Nosferatu	2024	FR
...		

■ **Figure 1** Example instance.



© Balder ten Cate, Phokion G. Kolaitis, and Carsten Lutz;
licensed under Creative Commons License CC-BY 4.0

28th International Conference on Database Theory (ICDT 2025).

Editors: Sudeepa Roy and Ahmet Kara; Article No. 15; pp. 15:1–15:19

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

We propose a formalization of the above problem through the notion of *query repairs*, as follows. We assume that we are given a query q and a set of *labeled examples*, by which we mean pairs (I, \mathbf{a}) with I a database instance and \mathbf{a} a tuple of values from the active domain of I , labeled as positive or negative to indicate whether \mathbf{a} is desired or undesired as an answer on input I . In the above example, for instance, the input query is q and there is a positively labeled example $(I, \text{Nosferatu})$ that q fails to fit. A *query repair*, then, is a query q' that fits the given labeled examples and “differs from q in a minimal way”. Different notions of query repair arise by using different means to formalize what it means for two queries to differ in a minimal way. Besides requiring that q' fits the labeled examples and differs minimally from q , depending on the context, it may be natural to additionally require that $q \subseteq q'$ or that $q' \subseteq q$. This leads to further refinements of the notion of a query repair, namely *query generalization* and *query specialization*, respectively, which we also investigate.

We propose a broad framework for defining what it means for two queries to differ in a minimal way, based on a *proximity pre-order* \preceq , i.e., a family of pre-orders \preceq_q (one for each query q), where $q' \preceq_q q''$ asserts that query q' is at least as close to q as q'' . A query q' is then a \preceq -*repair* of a query q if q' fits the given labeled examples and there is no query q'' with the same property such that $q'' \prec_q q'$. We instantiate this framework for conjunctive queries (CQs), focussing mainly on two kinds of proximity pre-orders: the *containment-of-difference* proximity pre-order \preceq^{cod} , based on query containment, and the *edit-distance* proximity pre-order $\preceq^{\text{edit-dist}}$, based on a distance metric between queries defined in terms of a suitably adapted version of edit distance. To be more precise, $q_1 \preceq_q^{\text{cod}} q_2$ if for every instance I , the symmetric difference of $q_1(I)$ and $q(I)$ is contained in the symmetric difference of $q_2(I)$ and $q(I)$. Moreover, $q_1 \preceq_q^{\text{edit-dist}} q_2$ if the edit distance between the homomorphism core of q_1 and the homomorphism core of q is no larger than the edit distance between the homomorphism core of q_2 and the homomorphism core of q , modulo variable renaming.

► **Example 1.2 (Generalization).** Consider the CQ $q(x) :- R(x, y), R(y, z), R(z, u), R(u, x)$ which returns all values that lie on a directed R -cycle of length 4, and the instance I that consists of the facts $R(a, b), R(b, c), R(c, a)$, i.e., I is the directed R -cycle of length 3. Clearly, $a \notin q(I)$. Let E be the singleton set of examples consisting of (I, a) labeled as a positive example. Which CQs qualify as repairs for (q, E) or as generalizations for (q, E) ? Note that since we only given positive examples, specializations do not seem to be a natural choice here.

It will turn out that there are two \preceq^{cod} -repairs for (q, E) : the CQ which expresses that x lies on a directed R -cycle of length 12 and the CQ which expresses that x lies on a directed R -cycle of length 3. Both of these are reasonable options. If we ask for \preceq^{cod} -generalizations for (q, E) , then only the first repair remains.

In contrast, there are precisely three $\preceq^{\text{edit-dist}}$ -repairs of (q, E) , each obtained from q by dropping a different atom from the body. Also these are reasonable options. The same CQs are also the $\preceq^{\text{edit-dist}}$ -generalizations for (q, E) .

► **Example 1.3 (Specialization).** Consider the CQ $q(x) :- R(x, y), R(y, z)$, which returns all values that have an outgoing R -path of length 2. Let E be the set consisting of

- a negative example (I, a) with $I = \{R(a, b), R(b, c)\}$, and
- a positive example (J, a) with $J = \{R(a, b), R(b, c), R(c, d)\}$.

Which CQs qualify as specializations for (q, E) ? In the same way in which generalizations are linked closely to the positive examples, specializations are linked closely to the negative examples. Note, however, that by itself the negative example in E does not provide much guidance as to what would be a “good repair” as there are many possible options. The positive example gives (in this case, quite specific) additional guidance regarding the “direction” towards which we should look to find the repair.

It will turn out that there is precisely one $\preceq^{\text{edit-dist}}$ -specialization, namely the very natural CQ $q'(x) :- R(x, y), R(y, z), R(z, u)$. However, q' does not qualify as a \preceq^{cod} -specialization, since the CQ $q''(x) :- R(x, y), R(y, z), R(u, z), R(u, v), R(v, w)$ also fits and is “closer” to q in terms of query containment. In fact, as we will see, no \preceq^{cod} -specialization for (q, E) exists.

The problem of constructing a query that fits a given set of labeled data examples has been studied extensively and is known under different names such as *reverse engineering*, *query learning*, or *fitting*; see for instance [25] which offers a comparison of several fitting algorithms for CQs. Recently, in [9], *extremal* variants of the fitting problem for CQs were studied, including (weakly/strongly) *most-general fitting* and *most-specific fitting*. There, the input consists of a set of positive and negative examples and the task is to find a most specific CQ, or a most general CQ, that fits them. We can think of such extremal fitting problems as constrained versions of the fitting problem for CQs where an additional requirement is put on the output query. In the same spirit, the *query repair* problem can also be viewed as a constrained version of fitting where the input now includes, in addition, a CQ q , and the output is required to be a fitting CQ that differs minimally from q .¹ As a part of our contributions, we will establish close relationships between query repair problems and extremal fitting problems.

Overview of contributions. In Sect. 3, we formally define \preceq -query repairs, as well as \preceq -generalizations and \preceq -specializations, based on a given proximity pre-order \preceq . We also propose, for each of these, three algorithmic problems: *verification*, *existence* and *construction*. The remaining sections focus specifically on CQs.

In Sect. 4, we study the containment-based proximity pre-order \preceq^{cod} . Besides examples of the resulting notions of generalization, specialization, and repair, our results, here, include:

- (a) structural characterizations that relate \preceq^{cod} -generalizations and \preceq^{cod} -specializations to most-specific fittings and most-general fittings, respectively (Theorems 4.4, 4.11, 4.12). These characterizations imply that there is always a unique \preceq^{cod} -generalization (unless no suitable fitting CQ exists) while \preceq^{cod} -specializations do not always exist.
- (b) based on this, results that identify the computational complexity of the verification, existence, and construction of \preceq^{cod} -generalizations and \preceq^{cod} -specializations.
- (c) results that relate \preceq^{cod} -repairs to \preceq^{cod} -specializations and \preceq^{cod} -generalizations, allowing us to apply some of the above algorithmic results to the more general case of \preceq^{cod} -repairs. However, we also illustrate that the behaviour of \preceq^{cod} -repairs is often counterintuitive. For instance, \preceq^{cod} -repairs need not exist and also there can be infinitely many \preceq^{cod} -repairs. In contrast to \preceq^{cod} -generalizations and \preceq^{cod} -specializations, \preceq^{cod} -repairs thus do not seem to be very natural.

In Sect. 5, we study proximity pre-orders based on distance metrics. In particular, we propose a distance metric for CQs based on edit distance that gives rise to a proximity pre-order $\preceq^{\text{edit-dist}}$. We show that there is always a non-empty and finite set of $\preceq^{\text{edit-dist}}$ -repairs (respectively, $\preceq^{\text{edit-dist}}$ -generalizations, and $\preceq^{\text{edit-dist}}$ -specializations), unless the given examples do not admit a fitting CQ. Moreover, we shed light on the complexity of the construction and verification problems (Thm. 5.15). We also show that other, seemingly natural distance metric lead to repair notions that behave worse.

¹ For the trivial proximity pre-order \preceq_q relating every CQ to every CQ, the query repair problem coincides with the fitting problem.

Outline. Sect. 2 contains technical preliminaries. In Sect. 3, we define query repairs. In Sect. 4, we explore containment-based query repairs. In Sect. 5, we explore query repairs based on distance metrics. We conclude in Sect. 6 with a discussion of future directions.

Due to lack of space, most proofs are omitted. They can be found in the full version [27].

Related work. Our notion of *query repairs* is in part inspired by the literature on *database repairs* introduced in [2]. There, one is given a database D that is inconsistent in the sense that it violates one or more integrity constraints and the aim is to answer a given query over all possible repairs of D , that is, all databases consistent with the integrity constraints that “differ from D in a minimal way”. Different notions of repairs, including set-based repairs and cardinality repairs, arise by formalizing in different ways what it means for two databases to “differ in a minimal way”. Research in this area has been rather active and fruitful [5].

There is extensive literature on approximating a query q by some other query q' such that q is contained in q' or q' is contained in q . In the former case q' is often called an *upper approximation* or an *upper envelope* of q , while in the latter case it is called a *lower approximation*, a *lower envelope*, or a *relaxation* of q . For instance, [23] proposes an algorithm for relaxing the where clause of an over-constrained database query that returns an empty result. Naturally, one is interested in optimal (with respect to containment) such approximations, which are known as *tight* upper or lower envelopes. Lipski [17] studied upper and lower approximations in the context of databases with incomplete information, while Libkin [22] carried out a study of formal models of approximation in databases. A related body of work focused on the problem of using approximation to achieve more efficient query evaluation. In particular, approximations of Datalog queries by CQs or unions of CQs were investigated in [11, 12]. More recently, approximations of CQs by CQs of tractable combined complexity (such as acyclic CQs or CQs of bounded treewidth) were studied in [3, 4]. In a different, yet related direction, tight lower envelopes were used in the area of answering queries using views [14, 18], where such envelopes approximate a perfect rewriting. Upper and lower envelopes were also used as tractable approximations of the answers to ontology-mediated queries, both over consistent databases [16] and over inconsistent ones [6].

The literature on approximations summarized above is based on the notion of containment of one query to another. Notions of “closeness” or “similarity” of queries that are not based on containment have also been investigated. For example, a notion of closeness based on suitable combinations of precision and recall was used to study the problem of translating a query over some schema to a semantically similar query over a different schema [10]. Furthermore, a notion of semantic similarity of queries based on available query logs was explored in [7].

In the area of *belief revision*, a number of proposals have been made for *model-based* revision and update operators, in which a knowledge base is viewed semantically as a set of *possible worlds* (where a world is a propositional truth assignment), and update/revision is performed on sets of possible worlds. Various concrete update and revision operators have been proposed based on different notions of relative proximity for possible worlds, including using Hamming distance [13, 15] and containment-of-difference [24, 29].

In software engineering, *automated program repair* techniques seek to aid developers by suggesting likely correct patches for software bugs. They take as input a program and a specification of correctness criteria that the fixed program should meet. Most techniques assume that the correctness criteria are given by means of a test suite: one or more failing tests indicate a bug to be fixed, while passing tests indicate behavior that should not change. The desired output is a set of program changes that leads all tests to pass. See [21] for an overview.

2 Preliminaries

As usual, a schema \mathcal{S} is a set of relation symbols, each with associated arity. A *database instance* over \mathcal{S} is a finite set I of *facts* of the form $R(a_1, \dots, a_n)$ where $R \in \mathcal{S}$ is a relation symbol of arity n and a_1, \dots, a_n are *values*. We use $\text{adom}(I)$ to denote the set of all values used in I . We can then view a *query* over a schema \mathcal{S} , semantically, as a function q that maps each database instance I over \mathcal{S} to a set of k -tuples $q(I) \subseteq \text{adom}(I)^k$, where $k \geq 0$ is the *arity* of the query. A query of arity zero is called a *Boolean* query. We write $q_1 \subseteq q_2$ and say that q_1 is *contained* in q_2 if $q_1(I) \subseteq q_2(I)$ for all database instances I . Two queries q_1 and q_2 are *equivalent*, written $q_1 \equiv q_2$, if $q_1 \subseteq q_2$ and $q_2 \subseteq q_1$.

A *data example* for a k -ary query q consists of a database instance I together with a k -tuple of values. We denote by $\llbracket q \rrbracket$ the set of all data examples (I, \mathbf{a}) for which it holds that $\mathbf{a} \in q(I)$. A *labeled example* is a data example that is labeled as positive or as negative. By a *collection of labeled examples* we mean a pair $E = (E^+, E^-)$, where E^+ and E^- are sets of examples. Here, the data examples in E^+ are considered as positive examples, and the data examples in E^- are considered as negative examples. A query q *fits* $E = (E^+, E^-)$ if $\mathbf{a} \in q(I)$ for each $(I, \mathbf{a}) \in E^+$ and $\mathbf{a} \notin q(I)$ for each $(I, \mathbf{a}) \in E^-$. In other words, q fits $E = (E^+, E^-)$ if $E^+ \subseteq \llbracket q \rrbracket$ and $E^- \cap \llbracket q \rrbracket = \emptyset$. Here, we assume that q has the same arity as the data examples in E^+ and E^- . We will often abuse notation and write that q fits E^+ (or that q fits E^-), meaning that q fits (E^+, \emptyset) (respectively, q fits (\emptyset, E^-)).

We will be focusing specifically on conjunctive queries. By a *k -ary conjunctive query (CQ)* over a schema \mathcal{S} , we mean an expression of the form $q(\mathbf{x}) :- \alpha_1, \dots, \alpha_n$ where $\mathbf{x} = x_1, \dots, x_k$ is a sequence of variables and each α_i is a relational atom that uses a relation symbol from \mathcal{S} and no constants. Note: the restriction to queries without constants is not essential for our results (cf. [25, Remark 2.3]) but simplifies the presentation.

The variables in \mathbf{x} are called *answer variables* and the other variables used in the atoms α_i are the *existential variables*. Each answer variable is required to occur in at least one atom α_i , a requirement known as the *safety condition*. For CQs q, q' of the same arity, we denote their *conjunction* by $q \wedge q'$ (where, for instance, the conjunction of $q_1(x, y) :- R(x, y, z)$ and $q_2(x, x) :- S(x, z)$ is $q(x, x) :- R(x, x, z), S(x, z')$). With the *size* of a CQ, denoted $|q|$, we mean the number of atoms in it. The query output $q(I)$ is defined as usual, cf. any standard database textbook.

Every CQ $q(x_1, \dots, x_k)$ has a *canonical example* e_q , namely the data example $(I_q, \langle x_1, \dots, x_k \rangle)$, where I_q is the database instance (over the same schema as q) whose active domain consists of the variables in q and whose facts are the atomic formulas in q .

Given data examples $e = (I, \mathbf{a})$ and $e' = (J, \mathbf{b})$ over the same schema and with the same number of distinguished elements, a *homomorphism* $h : e \rightarrow e'$ is a map from $\text{adom}(I)$ to $\text{adom}(J)$ that preserves all facts and such that $h(\mathbf{a}) = \mathbf{b}$. When such a homomorphism exists, we say that e “homomorphically maps to” e' and write $e \rightarrow e'$. We say that e and e' are *homomorphically equivalent* if $e \rightarrow e'$ and $e' \rightarrow e$. It then holds that $e \in \llbracket q \rrbracket$ iff $e_q \rightarrow e$. Furthermore, the well-known Chandra-Merlin theorem states that $q \subseteq q'$ holds iff $e_{q'} \rightarrow e_q$.

A data example e is said to be a *core* if every homomorphism $h : e \rightarrow e$ is surjective. It is well known that for every data example $e = (I, \mathbf{a})$ there is a subinstance $I' \subseteq I$ such that (I', \mathbf{a}) is a core and such that (I, \mathbf{a}) and (I', \mathbf{a}) are homomorphically equivalent. Moreover, such (I', \mathbf{a}) is unique up to isomorphism, and may be referred to as *the core of e* , denoted $\text{core}(e)$. We say that a CQ q is a *core* if its canonical example e_q is a core.

The *direct product* of two database instances, denoted $I \times J$, is the database instance containing all facts $R(\langle a_1, b_1 \rangle, \dots, \langle a_n, b_n \rangle)$ over the domain $\text{adom}(I) \times \text{adom}(J)$ such that $R(a_1, \dots, a_n)$ is a fact of I and $R(b_1, \dots, b_n)$ is a fact of J . This naturally extends to data examples: $(I, \langle a_1, \dots, a_k \rangle) \times (J, \langle b_1, \dots, b_k \rangle) = (I \times J, \langle \langle a_1, b_1 \rangle, \dots, \langle a_k, b_k \rangle \rangle)$.

3 Query Repairs

Fix a query language \mathcal{L} . A *proximity pre-order* \preceq for \mathcal{L} is a family of pre-orders \preceq_q , one for every $q \in \mathcal{L}$, satisfying the following conditions:

Conservativeness For all $q, q' \in \mathcal{L}$, $q \preceq_q q'$.

Syntax independence Whenever $q'_1 \equiv q_1$, $q'_2 \equiv q_2$ and $q'_3 \equiv q_3$, then $q_1 \preceq_{q_2} q_3$ iff $q'_1 \preceq_{q'_2} q'_3$.

Let $q \in \mathcal{L}$, and let E a collection of labeled examples (of the same arity as q). We call the pair (q, E) an *annotated \mathcal{L} -query*. The following are the 3 main notions studied in this paper.

- A \preceq -*repair* for (q, E) is a query $q' \in \mathcal{L}$ such that (i) q' fits E , and (ii) there is no $q'' \in \mathcal{L}$ with $q'' \prec_q q'$ that satisfies (i).
- A \preceq -*generalization* for (q, E) is a query $q' \in \mathcal{L}$ such that (i) q' fits E and $q \subseteq q'$ and (ii) there is no $q'' \in \mathcal{L}$ with $q'' \prec_q q'$ that satisfies (i).
- A \preceq -*specialization* for (q, E) is a query $q' \in \mathcal{L}$ such that (i) q' fits E and $q' \subseteq q$, and (ii) there is no $q'' \in \mathcal{L}$ with $q'' \prec_q q'$ that satisfies (i).

(where $q_1 \prec_q q_2$ is short for $q_1 \preceq_q q_2$ and $q_2 \not\preceq_q q_1$).

Note: under this definition, \preceq -specializations and \preceq -generalizations need not be \preceq -repairs.

Conservativeness and syntax-independence are minimal conditions on \preceq needed to yield intuitive behavior for query repairs: conservativeness ensures that if the input query q already fits the given examples, then q is its own \preceq -repair, while syntax independence ensures that equivalent queries have equivalent \preceq -repairs.

In the rest of this paper, we will restrict attention to CQs. That is, \mathcal{L} is the class of CQs.

► **Remark 3.1.** We will restrict our attention to repairs, generalizations, and specializations that use only the relation symbols which occur in (q, E) . Note that when E consists only of negative examples, a query repair or specialization could in principle contain relation symbols that do not occur in (q, E) . We will disregard such repairs. It is not difficult, however, to adapt our results to the case where such additional symbols would be admitted.

Several algorithmic problems arise, all parameterized with a proximity pre-order \preceq .

\preceq -REPAIR-VERIFICATION

Input: An annotated CQ (q, E) and a CQ q'

Output: Yes if q' is a \preceq -repair for (q, E) , No otherwise

\preceq -REPAIR-EXISTENCE

Input: an annotated CQ (q, E)

Output: Yes if there is a \preceq -repair for (q, E) , No otherwise

\preceq -REPAIR-CONSTRUCTION

Input: an annotated CQ (q, E) for which a \preceq -repair exists

Output: a \preceq -repair for (q, E)

We will also study the analogous algorithmic problems for \preceq -generalization and \preceq -specialization, defined in the expected way.

In all of the above problems, the input queries and examples are assumed to be compatible in terms of their arity. Moreover, in our complexity analyses, we will assume a fixed (constant) query arity $k \geq 0$. This is in fact only necessary for some of the upper bounds in Sect. 4.2.

4 Containment-Based Approach

In this section, we study notions of query generalization, query specialization and query repair defined based on query containment. For generalization and specialization, it seems intuitively clear what the definition should be. Let (q, E) be an annotated CQ. Then

- a *containment-based generalization* for (q, E) is a CQ q' that fits E and such that $q \subseteq q'$ and there is no CQ q'' that fits E with $q \subseteq q'' \subsetneq q'$.
- a *containment-based specialization* for (q, E) is a CQ q' that fits E and such that $q' \subseteq q$ and there is no CQ q'' that fits E with $q' \subsetneq q'' \subseteq q$.

It is less immediately clear what the right query containment-based definition of *query repairs* should be. As it turns out, the above notions of query generalization and query specialization can be viewed as query generalizations and query specializations with respect to the following natural proximity pre-order (cf. also [24, 29, 4]).

► **Definition 4.1** (Containment of Difference). *For queries q, q_1, q_2 , we write $q_1 \preceq_q^{\text{cod}} q_2$ if $\llbracket q \rrbracket \oplus \llbracket q_1 \rrbracket \subseteq \llbracket q \rrbracket \oplus \llbracket q_2 \rrbracket$, where \oplus denotes symmetric difference.*

Recall that $\llbracket q \rrbracket$ denotes the set of all positive examples of a query q . Therefore, $\llbracket q \rrbracket \oplus \llbracket q_i \rrbracket$ denotes the set of all examples on which q_i disagrees with q . Thus, $q_1 \preceq_q^{\text{cod}} q_2$ means that the set of examples on which q and q_1 disagree is a subset of the set of examples on which q and q_2 disagree (cod stands for “containment of difference”). It is easy to see that \preceq^{cod} is indeed a proximity pre-order. Moreover, it gives rise to the intended containment-based notions of query generalization and specialization:

► **Proposition 4.2.** *For all annotated CQs (q, E) and CQs q' ,*

1. q' is a \preceq^{cod} -generalization for (q, E) iff q' is a containment-based generalization for (q, E)
2. q' is a \preceq^{cod} -specialization for (q, E) iff q' is a containment-based specialization for (q, E) .

It also follows that \preceq^{cod} -generalizations and \preceq^{cod} -specializations are \preceq^{cod} -repairs. This furthermore suggests \preceq^{cod} -repairs as a (seemingly) natural containment-based notion of query repair. Next, we will study \preceq^{cod} -generalizations, \preceq^{cod} -specializations, and \preceq^{cod} -repairs. Our main findings can be summarized as follows: \preceq^{cod} -generalizations and \preceq^{cod} -specializations are well-behaved notions, although the latter do not always exist (Example 1.3) and can be too plentiful (Example 4.14). The associated existence, verification, and construction problems admit effective algorithms, although often of super-polynomial complexity. The more general \preceq^{cod} -repairs exhibits counter-intuitive behavior.

4.1 Containment-Based Query Generalizations

The following example illustrates \preceq^{cod} -generalizations.

► **Example 4.3.** Consider the schema consisting of unary relations P, Q . Let $q(x) :- P(x), Q(x)$ and let I be the instance that consists of the facts $P(a), Q(b)$. There is exactly one \preceq^{cod} -generalization for (q, E^+) , where E^+ is the set of positive examples $\{(I, a)\}$, namely $q'(x) :- P(x), Q(y)$. This is, in fact, also the only \preceq^{cod} -repair.

The next result show that there is a precise, two-way correspondence between \preceq^{cod} -generalizations and most-specific fitting CQs. A *most-specific fitting CQ* for a collection of labeled examples E is a fitting CQ q such that for every fitting CQ q' , $q \subseteq q'$ [9].

- **Theorem 4.4.** For all CQs q, q' and collections of labeled examples $E = (E^+, E^-)$,
1. q' is a \preceq^{cod} -generalization for (q, E) iff q' is a most-specific fitting CQ for $(E^+ \cup \{e_q\}, E^-)$.
 2. q is a most-specific fitting CQ for E iff q is a \preceq^{cod} -generalization for (q_\perp, E) ,
- where q_\perp denotes the maximally-constrained CQ over the relevant schema $\mathcal{S} = \{R_1, \dots, R_n\}$ and of the relevant arity, i.e., the CQ $q_\perp(x, \dots, x) :- R_1(x, \dots, x), \dots, R_n(x, \dots, x)$.

As a consequence of this, we can leverage known results about most-specific fitting CQs. For instance, it is known that, for every collection of labeled examples E , there is at most one most-specific fitting CQ up to equivalence, namely the CQ whose canonical example is the direct product of the positive examples in E . This implies:

► **Corollary 4.5.** Let (q, E) be an annotated CQ for which a fitting CQ q' with $q \subseteq q'$ exists. Then there is, up to equivalence, exactly one \preceq^{cod} -generalization for (q, E) .

► **Example 4.6** (Example 1.2 revisited). Using Thm. 4.4, we can verify the claim, in Example 1.2, that the CQ q' expressing “ x lies on a directed R -cycle of length 12” is the unique (up to equivalence) \preceq^{cod} -generalization for (q, E) . This is true because the canonical example of q' is homomorphically equivalent to the direct product of the positive example (I, a) and the canonical example e_q of the input CQ.

The complexity of various algorithmic problems pertaining to most-specific fitting CQs, as well as size bounds, were studied in [9]. From these, we obtain complexity results and size bounds for \preceq^{cod} -generalizations. We include here also an analysis for the case where the input consists of positive examples only, which is particularly natural in the case of query-generalizations: it follows from Thm. 4.4 and what is said below it that negative examples have no effect on generalizations except that they may render them non-existent.

► **Corollary 4.7.**

1. \preceq^{cod} -GENERALIZATION-EXISTENCE is *coNExpTime*-complete. For inputs consisting of a bounded number of examples, it is *coNP*-complete. For inputs consisting of positive examples only, it is in *PTime*.
2. \preceq^{cod} -GENERALIZATION-VERIFICATION is *NExpTime*-complete, even for inputs consisting of positive examples only. It is *DP*-complete for a bounded number of input examples.
3. \preceq^{cod} -GENERALIZATION-CONSTRUCTION is in *ExpTime* (and in *PTime* if the number of examples is bounded).
4. Let $q() :- R(x, x)$. There is a sequence of examples $(e_n)_{n \in \mathbb{N}}$ of size polynomial in n , such that (i) there is a \preceq^{cod} -generalization for (q, E_n^+) , where $E_n^+ = \{e_1, \dots, e_n\}$, and (ii) every \preceq^{cod} -generalization for (q, E_n^+) has size at least 2^n .

► **Remark 4.8.** Corollary 4.7(4) implies that the size of \preceq^{cod} -repairs is in general exponential in the number of positive examples, and also that the size of \preceq^{cod} -repairs for (q, E) cannot be bounded by *any* function in the size of q and the size of the smallest fitting CQ for E .

4.2 Containment-Based Query Specializations

The following example illustrate \preceq^{cod} -specializations.

- **Example 4.9.** Consider the CQ $q(x) :- P(x)$ and let E consist of
- a negative example (I, a) where I consists only of the fact $P(a)$, and
 - a positive example (J, a) where J extends I with the additional facts $Q(a)$ and $R(a, a)$.
- There are two \preceq^{cod} -repairs for (q, E) , namely $q'_1(x) :- P(x), Q(y)$ and $q'_2(x) :- P(x), R(y, z)$. It can be shown with the help of Thm. 4.11 below that these are the only two \preceq^{cod} -repairs.

An annotated CQ (q, E) may lack a \preceq^{cod} -specialization even when a fitting CQ exists:

► **Example 4.10.** Consider the Boolean CQ $q() :- R(x, y)$, and let E consist of

- a negative example I consisting of facts $R(b, c), R(c, b)$, and
- a positive example J consisting of the fact $R(a, a)$.

The positive example, here, is strictly speaking redundant: every CQ over the relevant schema fits it. It is included only for intuition. There are CQs q' with $q' \subseteq q$ that fit E (for instance, $q'() :- R(x, x)$ is such a query), but there does not exist a \preceq^{cod} -specialization for (q, E) . This can be seen as follows: for every CQ q' that fits E , by construction, the canonical example $e_{q'}$ is a non-2-colorable graph. By a well-known result in graph theory, $e_{q'}$ must then contain a cycle of odd length. By blowing up the length of this cycle (e.g. using the sparse incomparability lemma [20]), one can construct a fitting CQ q'' such that $q' \subsetneq q'' \subseteq q$.

In the previous subsection, we saw that \preceq^{cod} -generalizations are closely related to most-specific fitting CQs. Similarly, \preceq^{cod} -specializations are closely related to weakly most-general fitting CQs, where a *weakly most-general fitting CQ* for a collection of labeled examples E is a fitting q such that for every fitting CQ q' , $q \subseteq q'$ implies $q \equiv q'$ [9].

► **Theorem 4.11.** *Let (q, E) be any annotated CQ with $E = (E^+, E^-)$, such that q has no repeated answer variables. Then, for all CQs q' , the following are equivalent:*

1. q' is a \preceq^{cod} -specialization for (q, E) ,
2. q fits E^+ and q' is equivalent to $q \wedge q''$ for some q'' that is a weakly most-general fitting CQ for (E^+, E'^-) , where $E'^- = \{e \in E^- \mid e \in \llbracket q \rrbracket\}$.

Moreover, in the direction from 1 to 2, q'' can be chosen so that $|q''| \leq |q'|$.

Again, there is also a converse reduction, but it is more cumbersome to state because for $k > 0$, there does not exist a k -ary “most-general” CQ q_{\top} that is contained in all k -ary CQs (due to the safety condition in the definition of CQs). Instead, we need to consider all CQs $q(x_1, \dots, x_k)$ whose body only contains, for each $i \leq k$ one atom of the form $R(\mathbf{y}, x_i, \mathbf{z})$, where R is any relation symbol and \mathbf{y}, \mathbf{z} are tuples of distinct fresh existential variables. We call any such CQ a *minimally-constrained CQ*. Note that, for any schema and arity, there are finitely many minimally-constrained CQs (up to renaming of variables).

► **Theorem 4.12.** *For every CQ q and collection E of labeled examples, the following are equivalent:*

1. q is a weakly most-general fitting CQ for E ,
2. q is a \preceq^{cod} -specialization for (q, E) for all minimally-constrained CQs q_{\top} with $q \subseteq q_{\top}$.

For a given CQ q , the set of all minimally-constrained CQs q_{\top} with $q \subseteq q_{\top}$ can easily be constructed in polynomial time (for fixed query arity). This implies that the above proposition can be viewed as a polynomial-time (Turing) reduction.

► **Example 4.13** (Example 1.3 revisited). Let us revisit Example 1.3 from the introduction. There, we claimed that there is no \preceq^{cod} -specialization for (q, E) . In light of Thm. 4.11, it suffices to argue that there is no weakly most-general fitting CQ for E . We will not give a formal proof here, it suffices to consider CQs that describe an oriented R -paths of the form $\rightarrow \cdot (\rightarrow \cdot \leftarrow)^n \cdot \rightarrow$ for increasing values of n . The resulting infinite sequence $q_0 \subseteq q_1 \subseteq q_2 \subseteq \dots$ of CQs (each of which fits E) can be used to disprove the existence of a weakly most-general fitting CQ, and hence of a \preceq^{cod} -specialization for (q, E) .

Weakly most-general fitting CQs were studied in depth in [9]. Based on results from [9] and the above reductions, we obtain a number of results. In particular, the following example shows that an annotated CQ may have infinitely many non-equivalent \preceq^{cod} -specializations.

► **Example 4.14.** Consider the CQ $q() :- R(x, y)$ and let E consist of

- a positive example consisting of the facts $R(a, a), P_1(a), P_2(a)$, and
- a negative example consisting of the facts $R(a, a), R(b, b), R(a, b), P_1(a), P_2(b)$.

Note that the positive example is strictly speaking redundant as it belongs to $\llbracket q' \rrbracket$ for all CQs q' . It is added only for intuition. It follows from results in [9] that there are infinitely many non-equivalent weakly most-general fitting CQs for E . Indeed, for every $n \geq 1$, the CQ $q_n() :- R(x_1, x_2), \dots, R(x_{n-1}, x_n), P_2(x_1), P_1(x_n)$ is a weakly most-general fitting CQ for E . It follows by Thm. 4.11 that there are infinitely many \preceq^{cod} -specializations for (q, E) .

We also obtain a number of complexity results.

► **Corollary 4.15.**

1. \preceq^{cod} -SPECIALIZATION-EXISTENCE is *ExpTime*-complete.
2. \preceq^{cod} -SPECIALIZATION-VERIFICATION is in P_{\parallel}^{NP} and is DP-hard already for a bounded number of input examples.
3. \preceq^{cod} -SPECIALIZATION-CONSTRUCTION is in 2ExpTime .
4. There is a sequence $(q_n, E_n)_{n \in \mathbb{N}}$, where q_n and E_n are of total size polynomial in n , such that (i) there is a \preceq^{cod} -specialization for (q_n, E_n) , and (ii) every \preceq^{cod} -specialization for (q_n, E_n) is of size at least 2^n .

We do not know what happens in case 1 and 3 if the number of input examples is bounded.

4.3 Containment-Based Query Repairs

We now move to the general setting of \preceq^{cod} -repairs q' for an annotated CQ (q, E) where we no longer require that $q' \subseteq q$ or $q \subseteq q'$. Our first result on \preceq^{cod} -repairs provides a reduction to the case with only positive examples or only negative examples.

► **Proposition 4.16.** Let (q, E) be an annotated CQ with $E = (E^+, E^-)$. Then a CQ q' is a \preceq^{cod} -repair for (q, E) if and only if one of following holds:

- (a) q' is a \preceq^{cod} -repair for (q, E^+) and q' fits E^- , or
 - (b) q' is a \preceq^{cod} -repair for (q, \widehat{E}^-) and q' fits E^+ .
- where $\widehat{E}^- = \{e \times \prod_{e' \in E^+}(e') \mid e \in E^-\}$ if $E^+ \neq \emptyset$ and $\widehat{E}^- = E^-$ otherwise.

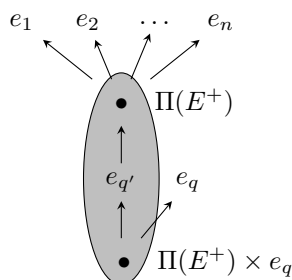
This is promising, as one might hope that case (a) above could be reduced to a statement about \preceq^{cod} -generalizations, since we are repairing w.r.t. a set of positive examples, and likewise for case (b) and \preceq^{cod} -specializations. For case (b) this approach indeed works:

► **Proposition 4.17.** For all annotated CQs (q, E) , if E consists of negative examples only, then every \preceq^{cod} -repair for (q, E) is a \preceq^{cod} -specialization for (q, E) .

The same, however, does *not* hold for case (a), as \preceq^{cod} -repairs w.r.t. positive examples are not necessarily \preceq^{cod} -generalizations:

► **Example 4.18.** Consider a schema consisting of unary relations P, Q, R . Let $q(x) :- P(x), Q(y)$, and let I be the instance consisting of the facts $P(a), R(b)$. There are, up to equivalence, two \preceq^{cod} -repairs for (q, E^+) where $E^+ = \{(I, a)\}$, namely the queries $q'_1(x) :- P(x)$ and $q'_2(x) :- P(x), R(y)$. Of these, only the first is a \preceq^{cod} -generalization. It seems counter-intuitive that q'_2 is a \preceq^{cod} -repair, since q'_2 is, intuitively, closer to q . However, there are instances on which q'_2 agrees with q but q'_1 does not. An example is the instance consisting only of the fact $P(a)$.

The following result characterizes the \preceq^{cod} -repairs for a CQ and set of positive examples.



■ **Figure 2** Picture of the condition in Thm. 4.19.

► **Theorem 4.19.** *Let (q, E^+) be an annotated CQ where E^+ consists only of positive examples. Then a CQ q' is a \preceq^{cod} -repair for (q, E^+) if and only if one of the following holds:*

1. q fits E^+ and q' is equivalent to q , or
 2. q does not fit E^+ , q' fits E^+ , and $(\Pi(E^+) \times e_{q \wedge q'}) \rightarrow e_{q'}$.
- In case of Boolean CQs, item 2 can be replaced by*
- 2'. q does not fit E^+ , and $(\Pi(E^+) \times e_q) \rightarrow e_{q'} \rightarrow \Pi(E^+)$ (cf. Figure 2).

► **Example 4.20** (Example 1.2 revisited). Using Thm. 4.19, one can easily verify the claim that the CQ expressing “ x lies on a directed R -cycle of length 3” is a \preceq^{cod} -repair for (q, E) . The same holds (it can be shown, with some more work) for the CQ expressing “ x lies on a directed R -cycle of length 6”, whose canonical example lies homomorphically in-between (I, a) (the cycle of length 3) and $(I, a) \times e_q$ (the cycle of length 12).

Using Thm. 4.19, we can show that there can be infinitely many \preceq^{cod} -repairs for a CQ and set of positive examples.

► **Example 4.21.** Over a schema consisting of a unary relation P and a binary relation R , consider the Boolean CQ $q() :- P(x)$ and the set of positive examples $E^+ = \{I\}$, where I consists of the fact $R(b, b)$. It follows from Thm. 4.19 that every CQ q' that fits E^+ is a \preceq^{cod} -repair. There are infinitely many pairwise non-equivalent such CQs.

The next result shows another connection between \preceq^{cod} -repairs and \preceq^{cod} -generalizations:

► **Proposition 4.22.** *The \preceq^{cod} -generalizations for an annotated CQ (q, E) are precisely the \preceq^{cod} -repairs for (q, E') , where E' extends E with the positive example e_q .*

Example 4.18 and Example 4.21 show that \prec^{cod} -repairs can behave counterintuitively. Various algorithmic results regarding the verification, existence and construction of \preceq^{cod} -repairs can be derived from the above characterizations and reductions, but we refrain from stating them here as they seem of little value given the problematic behaviour of \preceq^{cod} -repairs.

5 Query Repairs Based on Distance Metrics

In this section, we study proximity pre-orders based on distance metrics. In particular, we propose and study a variant of *edit distance* for CQs. We also study proximity pre-orders based on several other natural distance metrics. Our main findings can be summarized as follows: edit distance, suitably defined, yields a proximity pre-order that avoids some of the problems of \preceq^{cod} . We also show that other natural distance metrics induce proximity pre-orders that are less well behaved.

► **Definition 5.1** (Semantic distance metric). A semantic distance metric for CQs is a function $dist(\cdot, \cdot)$ from pairs of CQs (of the same arity) to non-negative real numbers, satisfying:

1. $dist(q_1, q_2) = dist(q_2, q_1)$,
2. $dist(q_1, q_2) = 0$ iff q_1 and q_2 are equivalent,
3. $dist(q_1, q_2) \leq dist(q_1, q_3) + dist(q_3, q_2)$ (the triangle inequality).

If all the conditions are met except for the only-if direction of 2, we say that $dist$ is a weak semantic distance metric for CQs.

One can think of a semantic distance metric for CQs as a distance metric (in the standard sense) on the equivalence classes of CQs. Every semantic distance metric, and in fact every weak semantic distance metric, induces a pre-order.

► **Definition 5.2** (Pre-order induced by a semantic distance metric). Let $dist$ be a weak semantic distance metric for CQs. We define \preceq^{dist} as follows: $q' \preceq_q^{dist} q''$ iff $dist(q, q') \leq dist(q, q'')$.

► **Proposition 5.3.** Let $dist$ be a weak semantic distance metric for CQs. Then \preceq^{dist} is a proximity pre-order.

We study \preceq^{dist} -repairs for several distance metrics. Besides the algorithmic problems of *repair existence, verification and construction*, we also consider the following natural fitting problem that is closely related to query repairs based on distance metrics:

dist-BOUNDED FITTING EXISTENCE

Input: an annotated CQ (q, E) and a distance bound $d \geq 0$.

Output: Yes if there is a CQ that fits E such that $dist(q, q') \leq d$, No otherwise

5.1 Edit Distance

A naive definition of the *edit distance* of two CQs q, q' would be the number of atoms that appear in one of the two CQs but not in the other, that is, $|I_q \oplus I_{q'}|$. It is easy to see that this is not a semantic distance metric: it is not invariant under logical equivalence, because simple syntactic changes such as renaming a quantified variable, which do not affect the semantics of the query, affect the edit distance. This can be fixed, however, by (i) taking homomorphism cores (i.e., minimizing the CQs), and (ii) working modulo bijective variable renamings.

This leads to the following definition. For simplicity, in this section we restrict attention to CQs whose sequence of answer variables is repetition-free (a restriction that could be lifted at the expense of a more intricate definition of edit distance, cf. Remark 5.16).

► **Definition 5.4** (Edit distance for CQs). Given CQs $q_1(x_1, \dots, x_k)$ and $q_2(y_1, \dots, y_k)$,

$$\text{edit-dist}(q_1, q_2) = \min_{\substack{\rho \text{ a bijective variable renaming} \\ \text{with } \rho(y_i) = x_i \text{ for } i = 1 \dots k}} |\text{core}(e_{q_1}) \oplus \text{core}(e_{\rho(q_2)})|$$

where $e_1 \oplus e_2$ denotes the set of facts occurring in example e_1 and not in e_2 or vice versa.

► **Example 5.5.** Consider the Boolean CQs

$$\begin{aligned} q_1() & :- R(x_1, x_2), R(x_1, x_3), R(x_2, x_4), R(x_3, x_4) \\ q_2() & :- R(x_1, x_2), R(x_1, x_3), R(x_2, x_4), R(x_3, x_4), A(x_2), B(x_3) \end{aligned}$$

Note that q_1 is not a core, but q_2 is. The core of q_1 is obtained by dropping the second and fourth atom. Thus $\text{edit-dist}(q_1, q_2) = 4$. Thus, perhaps surprisingly, $\text{edit-dist}(q_1, q_2)$ can be larger than the naive edit distance of q_1 and q_2 (which, in this case, is 2).

► **Proposition 5.6.** *edit-dist is a semantic distance metric.*

We next take a look at the complexity of computing edit distance.

► **Proposition 5.7.** *Testing whether $\text{edit-dist}(q_1, q_2) \leq n$ (on input q_1, q_2, n) is DP-hard and in Σ_2^P . When restricted to input queries that are cores, it is NP-complete.*

We now move on to studying the proximity pre-order $\preceq^{\text{edit-dist}}$. This pre-order has some useful structural properties. Let us say that a proximity pre-order \preceq is *well-founded* if for each CQ q , every non-empty set of CQs has a \preceq_q -minimal element (i.e., there are no infinite descending chains $\cdots \prec_q q_2 \prec_q q_1 \prec_q q_0$); and \preceq has the *finite-basis property* if for each CQ q , every set of CQs has only finitely many \preceq_q -minimal elements, up to equivalence.

► **Proposition 5.8.** *$\preceq^{\text{edit-dist}}$ is well-founded and has the finite-basis property.*

As an immediate consequence, we obtain:

► **Theorem 5.9.** *Let (q, E) be an annotated CQ.*

1. *If there is any CQ that fits E , then there is a $\preceq^{\text{edit-dist}}$ -repair for (q, E) .*
 2. *If there is any CQ q' that fits E with $q \subseteq q'$, there is a $\preceq^{\text{edit-dist}}$ -generalization for (q, E) .*
 3. *If there is any CQ q' that fits E with $q' \subseteq q$, there is a $\preceq^{\text{edit-dist}}$ -specialization for (q, E) .*
- Moreover, there are at most finitely many $\preceq^{\text{edit-dist}}$ -repairs, $\preceq^{\text{edit-dist}}$ -generalizations, and $\preceq^{\text{edit-dist}}$ -specializations for (q, E) , up to equivalence.*

We now compare $\preceq^{\text{edit-dist}}$ -repairs with \preceq^{cod} -repairs.

► **Example 5.10.** This example serves to compare $\preceq^{\text{edit-dist}}$ -generalizations with \preceq^{cod} -generalizations. Consider the following Boolean CQ and positive example:

$$\begin{aligned} q() & :- R(x, y), R(x, z), P_1(y), P_2(y), Q_1(z), Q_2(z) \\ e & = \{R(a, b), R(a, c), P_1(b), Q_1(b), P_2(c), Q_2(c)\} \end{aligned}$$

Let $E^+ = \{e\}$. There are four $\preceq^{\text{edit-dist}}$ -generalizations for (q, E^+) , namely

$$q'' :- R(x, y), R(x, z), P_i(y), Q_j(y) \text{ with } i, j \in \{1, 2\}.$$

In contrast, there is a unique (up to equivalence) \preceq^{cod} -generalization for (q, E) , namely

$$q'() :- R(x, y), R(x, z), R(x, u), R(x, v), P_1(y), P_2(z), Q_1(u), Q_2(v).$$

A variation of this example shows that (i) there can be exponentially more $\preceq^{\text{edit-dist}}$ -repairs than \preceq^{cod} -repairs, and (ii) \preceq^{cod} -repairs can be exponentially longer than $\preceq^{\text{edit-dist}}$ -repairs.

► **Example 5.11.** Consider again Example 4.10 which shows that \preceq^{cod} -specializations are not guaranteed to exist. There is a unique $\preceq^{\text{edit-dist}}$ -specialization, namely $q() :- R(x, x)$.

► **Example 5.12.** In Example 4.21, where \preceq^{cod} -repairs showed degenerative behavior, there exists a unique $\preceq^{\text{edit-dist}}$ -repair, namely the (intuitively expected) query $q() :- P(x)$.

A $\preceq^{\text{edit-dist}}$ -repair w.r.t. positive examples is not necessarily a $\preceq^{\text{edit-dist}}$ -generalization:

► **Example 5.13.** Consider the following Boolean CQs and example:

$$\begin{aligned} q() & :- R(x, y), R(x, z), R(y, u), R(z, u), P(y), Q(z) \\ q'_1() & :- R(x, y), R(x, z), R(y, u), R(z, u), P(y), W(z) \\ q'_2() & :- R(x, y), R(x, z), R(y, u), R(z, u), P(y) \\ e & = \{R(a, b), R(a, c), R(b, d), R(c, d), P(b), W(c)\} \end{aligned}$$

Let $E = (E^+, E^-)$ with $E^+ = \{e\}$ and $E^- = \emptyset$. Then q'_1 is the unique $\preceq^{\text{edit-dist}}$ -repair for (q, E) (having edit distance 2), but it is not a $\preceq^{\text{edit-dist}}$ -generalization. On the other hand, q'_2 is a $\preceq^{\text{edit-dist}}$ -generalization for (q, E) but not a $\preceq^{\text{edit-dist}}$ -repair as it has edit distance 3 (due to the fact that it is not a core). Similarly, it can be shown that a $\preceq^{\text{edit-dist}}$ -repair with respect to *negative* examples is not necessarily a $\preceq^{\text{edit-dist}}$ -specialization (cf. the full version of this paper).

The following upper bound on the size of $\preceq^{\text{edit-dist}}$ -repairs is implied by the definitions.

► **Proposition 5.14.** *Let (q, E) be an annotated CQ and q' a core CQ. If q' is an $\preceq^{\text{edit-dist}}$ -repair for (q, E) , then $|q'| \leq |q| + n$, where n is the size of the smallest fitting CQ for E . The same holds for $\preceq^{\text{edit-dist}}$ -generalizations and for $\preceq^{\text{edit-dist}}$ -specializations, where n is then the size of the smallest fitting CQ q'' that satisfies $q \subseteq q''$, respectively $q'' \subseteq q$.*

Prop. 5.14 stands in stark contrast with Remark 4.8 for \preceq^{cod} -repairs. We remark that, while the smallest fitting CQ may be exponential in the size of the input examples [28], one may expect it to be typically much smaller in practice.

We now consider algorithmic problems for $\preceq^{\text{edit-dist}}$ -repairs. By Thm. 5.9, the $\preceq^{\text{edit-dist}}$ -REPAIR-EXISTENCE problem coincides with the fitting existence problem. In particular, the existence of a $\preceq^{\text{edit-dist}}$ -repair for (q, E) does not depend on q . The verification and construction problems for $\preceq^{\text{edit-dist}}$ -repairs are more interesting and *do* depend on q . Of course, the existence of $\preceq^{\text{edit-dist}}$ -generalizations and $\preceq^{\text{edit-dist}}$ -specializations depends on q as well.

► **Theorem 5.15.**

1. $\text{edit-dist-BOUNDED-FITTING-EXISTENCE}$ is Σ_2^P -complete (provided the distance bound is given in unary). It is in NP if the input CQ is core and only positive examples are given.
 2. $\preceq^{\text{edit-dist}}$ -REPAIR-EXISTENCE is coNExpTime -complete. It is coNP -complete for inputs that consist of a bounded number of examples. If the input contains only positive examples, or only negative examples, it is in PTime.
 3. $\preceq^{\text{edit-dist}}$ -REPAIR-VERIFICATION is Π_2^P -hard and in Σ_3^P .
- Items 2 and 3 also hold if “repair” are replaced by “generalization” or “specialization”, except for the case of $\preceq^{\text{edit-dist}}$ -GENERALIZATION-EXISTENCE with a bounded number of examples, where we only have a DP upper bound.

Finally, let us discuss $\preceq^{\text{edit-dist}}$ -REPAIR-CONSTRUCTION. It follows from known results [8] that whenever a fitting CQ exists, there is one of size at most $n_{\max} = \prod_{e \in E^+} |e|$. A brute-force algorithm for $\preceq^{\text{edit-dist}}$ -REPAIR-CONSTRUCTION simply enumerates CQs in the order of increasing size and, for each, checks if it is a $\preceq^{\text{edit-dist}}$ -repair (cf. Thm. 5.15(2)). Since we are promised that an $\preceq^{\text{edit-dist}}$ -repair exists, this process terminates and, by Prop. 5.14, yields a CQ of size at most $|q| + n_{\max}$. We do not know of an algorithm for constructing $\preceq^{\text{edit-dist}}$ -repairs with asymptotically better running time, see Sect. 6 for further discussion.

► **Remark 5.16.** One can further refine edit distance by requiring that all equalities between variables in the body of the query are represented explicitly by means of equality atoms, and by counting these when computing the symmetric difference. For example, consider the CQs

$$q(x) :- P(x) \quad q'_1(x) :- P(x), R(y, z) \quad q'_2(x) :- P(x), R(y, y)$$

Under Def. 5.4, $\text{edit-dist}(q, q'_1) = \text{edit-dist}(q, q'_2)$, while under the more refined definition of edit distance (where we treat q'_2 as shorthand for $q'_2(x) :- P(x), R(y, z), y = z$), $\text{edit-dist}(q, q'_1) < \text{edit-dist}(q, q'_2)$. We omit the details, but we believe that the above results continue to hold under such a more refined notion of edit distance.

5.2 Other Distance Metrics

Distance as size of the smallest distinguishing instance. The next distance metric we consider is based on the smallest instance on which the two queries produce different answers.

► **Definition 5.17.** For CQs q_1 and q_2 , $\text{sdi-dist}(q_1, q_2) = 1/n$, where n is the size of the smallest instance I (measured by the number of facts) such that $q_1(I) \neq q_2(I)$, or 0 if no such instance I exists.

► **Proposition 5.18.** sdi-dist is a semantic distance metric, and in fact an ultrametric (i.e., it satisfies $\text{dist}(q_1, q_3) \leq \max(\text{dist}(q_1, q_2), \text{dist}(q_2, q_3))$).

► **Example 5.19.** Consider the CQs $q_1() :- R(x, x)$ and $q_2() :- \bigwedge_{i,j \in \{1, \dots, N\}, i \neq j} R(x_i, x_j)$. An instance that satisfies q_2 either isomorphically embeds a clique of size N or else contains a “reflexive” fact of the form $R(a, a)$. Therefore, every example distinguishing q_1 from q_2 must contain at least $N(N - 1)$ facts. It follows that $\text{sdi-dist}(q_1, q_2) = 1/(N(N - 1))$.

Two further relevant basic facts about sdi-dist are the following:

► **Proposition 5.20.** For all CQs q, q' , $\text{sdi-dist}(q, q') = 0$ or $\text{sdi-dist}(q, q') \geq 1/\max(|q|, |q'|)$.

► **Proposition 5.21.** Computing sdi-dist is NP-hard. More precisely, testing $\text{sdi-dist}(q, q') \leq 1/k$ (on input CQs q, q' and natural number $k \geq 0$ in unary) is NP-hard and is in Π_2^P .

We will now move on to study the pre-order $\preceq^{\text{sdi-dist}}$. The next example shows that $\preceq^{\text{sdi-dist}}$ is not a good pre-order for query repairs, since it is not sufficiently discriminative.

► **Example 5.22.** Let $q_1(x) = R(x, y)$, and let E_1 consist of

- the negative example (I, a) where $I = \{R(a, b)\}$, and
- the positive example (J, a) where $J = \{R(a, a)\}$.

The positive example is strictly speaking redundant: every CQ over the relevant schema fits it. It is added only for intuition. There are infinitely many pairwise non-equivalent CQs q' that fit E with $q' \subseteq q_1$. Furthermore, by construction, every fitting CQ q' disagrees with q_1 on I , and hence has $\text{sdi-dist}(q', q_1) = 1$. It follows that all infinitely-many fitting CQs are $\preceq^{\text{sdi-dist}}$ -repairs for (q_1, E) , and there are infinitely many $\preceq^{\text{sdi-dist}}$ -specializations for (q_1, E) as well.

A similar situation arises for $\preceq^{\text{sdi-dist}}$ -generalizations: let $q_2(x) = R(x, x, x)$ and let E_2 consist of the positive example (I', a) where $I' = \{R(a, a, b)\}$. There are infinitely many CQs (up to equivalence) that fit E_2 and they all disagree with q_2 on the single-fact instance I' .

This shows that there are annotated CQs with infinitely many $\preceq^{\text{sdi-dist}}$ -repairs (as well as $\preceq^{\text{sdi-dist}}$ -specializations and $\preceq^{\text{sdi-dist}}$ -generalizations). On the flip side, we have:

► **Proposition 5.23.** Let (q, E) be an annotated CQ.

1. If there is any CQ that fits E , then there is a $\preceq^{\text{sdi-dist}}$ -repair for (q, E) .
2. If there is any CQ q' that fits E with $q \subseteq q'$, there is a $\preceq^{\text{sdi-dist}}$ -generalization for (q, E) .
3. If there is any CQ q' that fits E with $q' \subseteq q$, there is a $\preceq^{\text{sdi-dist}}$ -specialization for (q, E) .

We omit a complexity-theoretic analysis of $\preceq^{\text{sdi-dist}}$ -repairs in light of Example 5.22.

► **Remark 5.24.** Dually to sdi-dist one can also consider the distance metric sdq-dist defined as the size of the smallest distinguishing query. More precisely, $\text{sdq-dist}(q, q')$ is the size of the smallest CQ (as measured by the number of atoms) that maps homomorphically to precisely one of q, q' . Unfortunately, sdq-dist fares no better than sdi-dist . For instance, for the annotated CQ (q_2, E_2) from Example 5.22, all CQs q' that fit E_2 again have $\text{sdq-dist}(q, q_2) = 1$.

Distance as probability of disagreement. Let μ be a discrete probability distribution over the space of all examples (an *example distribution*, for short). For instance, μ may be a uniform distribution over values in some pre-existing (unlabeled) database instance. We define $dist_\mu(q, q') = \mu(\llbracket q \rrbracket \oplus \llbracket q' \rrbracket)$. That is, $dist_\mu(q, q')$ is the probability of drawing an example on which q and q' disagree. The same works for non-discrete probability distributions, as long as $\llbracket q \rrbracket$ is measurable for each CQ q . We restrict to discrete distributions for simplicity. It is easy to see that, for all example distributions μ , $dist_\mu$ is a weak semantic distance metric.

► **Proposition 5.25.** $dist_\mu(q, q')$ can be computed in $P_{||}^{NP}$ (for example distributions μ with finite support, specified as part of the input). Testing $dist_\mu(q, q') \leq r$ is $P_{||}^{NP}$ -complete.

For the sake of readability, we will denote the proximity pre-order \preceq^{dist_μ} by \preceq^μ .

► **Proposition 5.26.** Let μ be any example distribution. If μ has finite support, \preceq^μ is well-founded. The same does not necessarily hold when μ has infinite support.

As a consequence, we obtain:

► **Proposition 5.27.** Let (q, E) be an annotated CQ and μ an example distribution with finite support.

1. If there is any CQ that fits E , then there is a \preceq^μ -repair for (q, E) .
2. If there is any CQ q' that fits E with $q \subseteq q'$, there is a \preceq^μ -generalization for (q, E) .
3. If there is any CQ q' that fits E with $q' \subseteq q$, there is a \preceq^μ -specialization for (q, E) .

On the other hand, there can be infinitely many \preceq^μ -repairs. Indeed, it is not difficult to show the following using a pigeon-hole argument:

► **Proposition 5.28.** Let μ be any example distribution with finite support. There are annotated CQs (q, E) for which there are infinitely many \preceq^μ -repairs, up to equivalence.

In light of this, we omit a complexity-theoretic analysis for \preceq^μ -repairs.

6 Discussion

We proposed and studied notions of \preceq -generalizations, \preceq -specializations, and \preceq -repairs, parameterized by a proximity pre-order \preceq , providing a principled framework for example-driven query debugging and refinement (an idea partially inspired by the interactive schema-mapping design tool EIRINI [1]). We explored two ways to obtain a proximity pre-order for CQs: containment-based and edit distance-based. In each case, we assessed the behavior of the obtained repair notions through examples, and we studied the existence, verification and construction problems, as well as the size of repairs. Other algorithmic problems may be considered in followup work, such as repair enumeration (cf. [19]) and computing “possible” or “certain” answers across all repairs of a given CQ (as in [4] for query approximations).

Based on our findings, \preceq^{cod} -generalizations and \preceq^{cod} -refinements are reasonably well-behaved (although the latter do not always exist) while unconstrained \preceq^{cod} -repairs are not; $\preceq^{\text{edit-dist}}$ -repairs behave favorably: they exist whenever a fitting CQ exists; there are always only finitely many repairs up to equivalence; and the main algorithmic tasks are decidable with typically lower complexity than for \preceq^{cod} -repairs; in addition, $\preceq^{\text{edit-dist}}$ -repairs tend to be of smaller size (Prop. 5.14). Although $\preceq^{\text{edit-dist}}$ -repairs, too, in some cases display surprising behavior (Example 5.13), this may be an unavoidable consequence of our syntax-independence requirement taken together with the inherently syntactic nature of edit distance.

An important outstanding issue with $\preceq^{\text{edit-dist}}$ -repairs is to design practical algorithms for constructing them. In [26], a SAT-based approach for computing minimal-size fitting \mathcal{ELT} -concepts (i.e., Berge-acyclic connected unary CQs) was proposed and evaluated, showing promising performance. While a SAT-solver may not be applicable here due to the higher complexity of the problem, we believe that inspiration can be taken from this approach. It may also be worthwhile to study approximation algorithms for distance-based query repairs, i.e., algorithms that produce fitting CQs that have near-minimal distance to the input CQ.

Naturally, these result are specific to the particular class of queries we considered: CQs. One may also study query repairs for other query classes (e.g., self-join free CQs, unions of CQs, or nested queries). Note, in particular, that self-join free CQs are cores and our examples of counterintuitive behavior of $\preceq^{\text{edit-dist}}$ -repairs are based on CQs that are not cores. Moreover, the computational problems associated with $\preceq^{\text{edit-dist}}$ -repairs are often of lower complexity for CQs that are cores.

It is also natural to let the space of candidate repairs depend on the input query in a stronger way. For instance, we may require that the join structure of the query remains fixed, and that repairs differ only in their WHERE-clause (cf. [23]).

Among different avenues for further research, let us mention one: query repair operations could be studied from a more structural perspective. For instance, how does repairing a query w.r.t. a collection of labeled examples $E \cup E'$ compare to repairing it w.r.t. E followed by repairing it w.r.t. E' ? And, assuming a “true” CQ q^* , if we repair a given CQ q repeatedly based on labeled examples for q^* , does this process converge towards q^* in a formal sense?

References

- 1 Bogdan Alexe, Balder ten Cate, Phokion G. Kolaitis, and Wang-Chiew Tan. Designing and refining schema mappings via data examples. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data, SIGMOD '11*, pages 133–144, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1989323.1989338.
- 2 Marcelo Arenas, Leopoldo E. Bertossi, and Jan Chomicki. Consistent query answers in inconsistent databases. In Victor Vianu and Christos H. Papadimitriou, editors, *Proceedings of the Eighteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 31 - June 2, 1999, Philadelphia, Pennsylvania, USA*, pages 68–79. ACM Press, 1999. doi:10.1145/303976.303983.
- 3 Pablo Barceló, Leonid Libkin, and Miguel Romero. Efficient approximations of conjunctive queries. *SIAM J. Comput.*, 43(3):1085–1130, 2014. doi:10.1137/130911731.
- 4 Pablo Barceló, Mikael Romero, and Thomas Zeume. A more general theory of static approximations for conjunctive queries. *Theory of Computing Systems*, 64(4):916–964, 2020. doi:10.1007/s00224-019-09924-0.
- 5 Leopoldo E. Bertossi. *Database Repairing and Consistent Query Answering*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2011. doi:10.2200/S00379ED1V01Y201108DTM020.
- 6 Meghyn Bienvenu and Riccardo Rosati. Tractable approximations of consistent query answering for robust ontology-based data access. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 775–781. IJCAI/AAAI, 2013. URL: <http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6904>.
- 7 Ilaria Bordino, Carlos Castillo, Debora Donato, and Aristides Gionis. Query similarity by projecting the query-flow graph. In Fabio Crestani, Stéphane Marchand-Maillet, Hsin-Hsi Chen, Efthimis N. Efthimiadis, and Jacques Savoy, editors, *Proceeding of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2010, Geneva, Switzerland, July 19-23, 2010*, pages 515–522. ACM, 2010. doi:10.1145/1835449.1835536.

- 8 Balder ten Cate and Víctor Dalmau. The product homomorphism problem and applications. In Marcelo Arenas and Martín Ugarte, editors, *18th International Conference on Database Theory, ICDT 2015, March 23-27, 2015, Brussels, Belgium*, volume 31 of *LIPICs*, pages 161–176. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICs.ICDT.2015.161.
- 9 Balder ten Cate, Victor Dalmau, Maurice Funk, and Carsten Lutz. Extremal fitting problems for conjunctive queries. In *Proceedings of PODS 2023*, 2023.
- 10 Kevin Chen-Chuan Chang and Hector Garcia-Molina. Approximate query mapping: Accounting for translation closeness. *VLDB J.*, 10(2-3):155–181, 2001. doi:10.1007/S007780100042.
- 11 Surajit Chaudhuri. Finding nonrecursive envelopes for datalog predicates. In Catriel Beeri, editor, *Proceedings of the Twelfth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 25-28, 1993, Washington, DC, USA*, pages 135–146. ACM Press, 1993. doi:10.1145/153850.153862.
- 12 Surajit Chaudhuri and Phokion G. Kolaitis. Can datalog be approximated? *J. Comput. Syst. Sci.*, 55(2):355–369, 1997. doi:10.1006/JCSS.1997.1528.
- 13 Mukesh Dalal. Investigations into a theory of knowledge base revision: preliminary report. In *Proceedings of the Seventh AAAI National Conference on Artificial Intelligence, AAAI’88*, pages 475–479. AAAI Press, 1988.
- 14 Oliver M. Duschka and Michael R. Genesereth. Answering recursive queries using views. In Alberto O. Mendelzon and Z. Meral Özsoyoglu, editors, *Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, May 12-14, 1997, Tucson, Arizona, USA*, pages 109–116. ACM Press, 1997. doi:10.1145/263661.263674.
- 15 Kenneth D. Forbus. Introducing actions into qualitative simulation. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, pages 1273–1278, Detroit, MI, August 1989. Morgan Kaufmann. URL: <http://ijcai.org/Proceedings/89-2/Papers/068.pdf>.
- 16 Anneke Haga, Carsten Lutz, Leif Sabellek, and Frank Wolter. How to approximate ontology-mediated queries. In *Proc. of KR*, pages 323–333, 2021. doi:10.24963/KR.2021/31.
- 17 Witold Lipski Jr. On semantic issues connected with incomplete information databases. *ACM Trans. Database Syst.*, 4(3):262–296, 1979. doi:10.1145/320083.320088.
- 18 Verena Kantere, George Orfanoudakis, Anastasios Kementsietsidis, and Timos K. Sellis. Query relaxation across heterogeneous data sources. In James Bailey, Alistair Moffat, Charu C. Aggarwal, Maarten de Rijke, Ravi Kumar, Vanessa Murdock, Timos K. Sellis, and Jeffrey Xu Yu, editors, *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015, Melbourne, VIC, Australia, October 19 - 23, 2015*, pages 473–482. ACM, 2015. doi:10.1145/2806416.2806529.
- 19 Benny Kimelfeld, Ester Livshits, and Liat Peterfreund. Counting and enumerating preferred database repairs. *Theoretical Computer Science*, 837:115–157, 2020. doi:10.1016/j.tcs.2020.05.016.
- 20 Gábor Kun. Constraints, MMSNP and expander relational structures. *Combinatorica*, 33(3):335–347, 2013. doi:10.1007/S00493-013-2405-4.
- 21 Claire Le Goues, Michael Pradel, and Abhik Roychoudhury. Automated program repair. *Commun. ACM*, 62(12):56–65, November 2019. doi:10.1145/3318162.
- 22 Leonid Libkin. Models of approximation in databases. *Theor. Comput. Sci.*, 190(2):167–210, 1998. doi:10.1016/S0304-3975(97)00090-X.
- 23 Ion Muslea and Thomas J. Lee. Online query relaxation via bayesian causal structures discovery. In Manuela M. Veloso and Subbarao Kambhampati, editors, *Proceedings, The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference, July 9-13, 2005, Pittsburgh, Pennsylvania, USA*, pages 831–836. AAAI Press / The MIT Press, 2005. URL: <http://www.aaai.org/Library/AAAI/2005/aaai05-131.php>.

- 24 K. Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems (FGCS'88)*, pages 455–462, 1988.
- 25 Balder ten Cate, Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Fitting algorithms for conjunctive queries. *SIGMOD Record*, 52, 2023. doi:10.1145/3641832.3641834.
- 26 Balder ten Cate, Maurice Funk, Jean Christoph Jung, and Carsten Lutz. Sat-based pac learning of description logic concepts. In Edith Elkind, editor, *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3347–3355. International Joint Conferences on Artificial Intelligence Organization, August 2023. Main Track. doi:10.24963/ijcai.2023/373.
- 27 Balder ten Cate, Phokion Kolaitis, and Carsten Lutz. Query repairs, 2025. arXiv:2501.11162.
- 28 Ross Willard. Testing expressibility is hard. In David Cohen, editor, *Principles and Practice of Constraint Programming - CP 2010 - 16th International Conference, CP 2010, St. Andrews, Scotland, UK, September 6-10, 2010. Proceedings*, volume 6308 of *Lecture Notes in Computer Science*, pages 9–23. Springer, 2010. doi:10.1007/978-3-642-15396-9_4.
- 29 Marianne Winslett. *Updating Logical Databases*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.