Why is it hard to obtain a dichotomy for consistent query answering?

GÄELLE FONTAINE, University of Chile

A database may for various reasons become inconsistent with respect to a given set of integrity constraints. In the late nineties, the formal approach of consistent query answering has been proposed in order to query such databases. Since then, a lot of efforts have been spent to classify the complexity of consistent query answering under various classes of constraints. It is known that for the most common constraints and queries, the problem is in \( \text{coNP} \) and might be \( \text{coNP} \)-hard, yet several relevant tractable classes have been identified. Additionally, the results that emerged suggested that given a set of key constraints and a conjunctive query, the problem of consistent query answering is either in \( \text{PTIME} \) or is \( \text{coNP} \)-complete. However, despite all the work, as of today this dichotomy remains a conjecture.

The main contribution of this paper is to explain why it appears so difficult to obtain a dichotomy result in the setting of consistent query answering. Namely, we prove that such a dichotomy with respect to common classes of constraints and queries, is harder to achieve than a dichotomy for the constraint satisfaction problem, which is a famous open problem since the 1990s.

Categories and Subject Descriptors: H.2.3 [Database Management]: Languages—Query languages; D.3.2 [Programming Languages]: Language Classifications—Constraint and logic languages

General Terms: Theory; Languages

Additional Key Words and Phrases: inconsistent databases, consistent query answering, dichotomy, constraint satisfaction problem

ACM Reference Format:


doi: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

1.1. Querying inconsistent databases

One way to control databases is to impose \textit{integrity constraints} upon them, that is, semantic properties that the database must obey. However, in many situations control can be lost (e.g. in the context of data integration or exchange [Lenzerini 2002; Arenas et al. 2014]). This gives rise to \textit{inconsistent} databases, which no longer satisfy the constraints.

To overcome the problem, one option is to restore consistency using \textit{data cleaning}. The approach consists in arbitrarily transforming the database into a well-behaved one. Another approach, introduced by Arenas et al [Arenas et al. 1999], is to directly query the original database, as inconsistent as it is. The \textit{consistent answer} of a query \( q \) on an inconsistent database \( D \) is then defined as the intersection of the answers of \( q \) on all the consistent databases that differs from \( D \) in a “minimal way”.

Authors’ addresses: G. Fontaine, Department of Computer Science, University of Chile, Avda Blanco Encalada 2120, 4to piso, Santiago, Chile.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2014 ACM 0362-5915/2014/09-ART1 $15.00

doi: http://dx.doi.org/10.1145/0000000.0000000

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
The approach is elegant and principled. However, the abstraction of the method is counterbalanced by a high computational complexity. Since the seminal work of Arenas et al [Arenas et al. 1999], the computational complexity of consistent query answering has been studied for various classes of constraints. Initially, the focus was on functional constraints, inclusion dependencies and denial constraints (see the overviews [Bertossi 2006; Chomicki 2007]). More recently, other classes of constraints such as LAV constraints, GAV constraints, tuple-generating-dependencies (tgds) and equality-generating-dependencies (egds) [Staworko and Chomicki 2010; Arenas and Bertossi 2010; ten Cate et al. 2012] have also been considered. Those constraints play a central role in data integration [Lenzerini 2002] and data exchange [Fagin et al. 2003; Arenas et al. 2014].

As an attempt to classify the complexity of consistent query answering, the question of the existence of a dichotomy result for the problem of consistent query answering under a set of key constraints, has been raised. The conjecture is that given a conjunctive query \( q \) and a set of key constraints \( \Sigma \), the problem of consistent query answering of \( q \) under \( \Sigma \) should either be in PTIME or \( \text{coNP} \)-complete. Recall that if PTIME \( \neq \) \( \text{NP} \), there are infinitely many intermediate problems in \( \text{coNP} \) that are neither \( \text{coNP} \)-complete nor belong to PTIME [Ladner 1975]. A dichotomy conjecture states that the considered class of problems does not contain any intermediate problem.

The question has been actively explored recently; yet only few results, and in very restricted settings, have been obtained. The first of these results is a necessary and sufficient condition for first-order rewriting of acyclic conjunctive queries without self-joins [Wijsen 2010] (note that first-order rewritability implies tractability for consistent query answering). Given that condition, Kolaitis and Pema [Kolaitis and Pema 2012] proved a dichotomy theorem for queries containing only two atoms and no self-joins. Even with such strong restrictions, the proof turned out to be involved.

We show that there is actually a very good reason for the difficulties encountered. We prove that a dichotomy result for consistent query answering would imply a solution for a famous long-standing open problem, namely the dichotomy conjecture for the constraint satisfaction problem.

1.2. Constraint satisfaction problem

The constraint satisfaction problem (CSP) [Meseguer 1989; Tsang 1993; Vardi 2000] is a fundamental topic in computer science, the main reason being that CSP provides a common framework for a wide range of problems arising in theoretical computer science and artificial intelligence. An instance of CSP is determined by a set of variables, a set of values and a set of constraints. The goal is to assign a value to each variable in such a way that the constraints are satisfied.

In general, CSP is in NP and there are families of instances (e.g. boolean satisfiability) that are known to be NP-complete. An impressive amount of effort has been devoted to isolate tractable cases and develop heuristics. The classes of instances that received the most attention are the non-uniform constraint satisfaction problems. Each of those classes is characterized by a fixed set of allowed constraint relations; examples include boolean satisfiability, graph coloring and systems of equations.

The first major result [Schaefer 1978] concerning non-uniform CSP establishes that every boolean non-uniform CSP is either polynomial or is NP-complete, where an instance of CSP is said to be boolean if its set of values contains exactly two elements. Feder and Vardi [Feder and Vardi 1998] postulated that the result holds for arbitrary non-uniform CSP, that is, each non-uniform CSP is either solvable in polynomial time or is NP-complete. This conjecture is known as the dichotomy conjecture for CSP and is the most important open problem in the field.
Initially, and despite the considerable attention received by the problem, progress was slow. However, after the adoption of an algebraic approach, some significant results have been obtained. The most recent developments include a dichotomy theorem for non-uniform CSP over sets of values with three elements [Bulatov 2006] and a dichotomy theorem for non-uniform conservative CSP [Bulatov 2003; Barto 2011], that is, non-uniform CSP over a constraint language containing all unary relations. The proofs of those results are highly complex.

1.3. Linking two conjectures about separation
Our goal is to explain why it appears so difficult to obtain a dichotomy result in the setting of consistent query answering. We do so by proving that if such a dichotomy result holds, then so does the dichotomy conjecture for CSP. We were not able to prove such a result in the setting described by Afrati and Kolaitis (i.e. key constraints and conjunctive queries). The solution is to turn our attention to GAV constraints and unions of conjunctive queries (UCQ), which are common well-studied classes of constraints and queries.

The main result establishes that if the dichotomy conjecture holds for consistent query answering of UCQs w.r.t a set of GAV constraints, then so does the dichotomy conjecture for CSP. Given the fact that the dichotomy conjecture for CSP is still open and that a proof would be the most fundamental breakthrough in the study of CSP, our result means that there is very little hope in pursuing a dichotomy result for consistent query answering in its most general form.

Concerning key constraints, even though we do not have a result similar to our main theorem, we prove that a dichotomy result for consistent query answering of UCQs with constants with respect to key constraints, would yield to an alternative proof of the dichotomy theorem for conservative CSP. Considering the time and the effort spent to obtain a dichotomy for conservative CSP, this shows that a dichotomy for consistent query answering in the setting described above is a highly difficult task.

Our third result establishes that a dichotomy result for consistent query answering of UCQs with respect to egds, would yield to an alternative proof of the dichotomy theorem for conservative CSP. Compared to our second result, this shows that if we are willing to consider egds constraints instead of key constraints, then we do not need constants in the queries.

The three results presented provide a formal explanation of the difficulty of proving a dichotomy for consistent query answering; they also emphasize the close connection between consistent query answering and CSP. It does not mean though that no further investigation of a dichotomy for consistent query answering in restricted settings should be pursued and that no meaningful understanding will be gained.

1.4. Related work
Links between the dichotomy conjecture for (non-uniform) CSP and a possible dichotomy result for problems arising in database theory have been previously explored. Feder and Vardi [Feder and Vardi 1998] proved that the logic MMSNP and non-uniform CSP are polynomially equivalent. Hence, the dichotomy conjecture holds for CSP iff it holds for MMSNP. Finally, let us mention the results of Calvanese et al [Calvanese et al. 2000] establishing a connection between the tractable instances of CSP and the instances of query rewriting that admit a perfect rewriting in polynomial time. Those results do not prove though that a dichotomy theorem in one setting implies a dichotomy result in the other setting.

This article is an extended version of [Fontaine 2013]. It contains the proofs of Theorem 4.1 and Theorem 4.2. The techniques used in those proofs are non trivial and
might offer some insight on how to extend the main result in the case of key constraints. That is, how to prove that a dichotomy result for consistent query answering of UCQs with respect to key constraints, would yield to a dichotomy theorem for CSP. The case of key constraints is of particular interest, as this is the setting of the original dichotomy conjecture.

Organisation of the paper In Section 2, we introduce the basics of consistent query answering and CSP. In Section 3, we present our main result, namely that a dichotomy result for consistent query answering of UCQs with respect to GAV constraints, implies a dichotomy theorem for CSP. Finally, in Section 4, we mention two other results establishing a connection between conservative CSP and consistent query answering of UCQs with respect to key constraints and egds. Concluding remarks can be found in Section 5.

2. PRELIMINARIES
2.1. Consistent query answering
A schema $\sigma$ is a set of relation symbols with associated arities. A database $D$ over the schema $\sigma$ assigns to each relation symbol $R_i$ with arity $n_i$ a finite $n_i$-ary relation $R^D_i$. The active domain is the set of all elements that occur in any of the relations $R^D_i$. Databases can be seen as first-order structures by taking the domain to be the active domain.

If $(a_1, \ldots, a_n)$ belongs to $R^D_i$, we say that $R_i(a_1, \ldots, a_n)$ is a fact of $D$. Each database can be identified with the set of its facts.

A set of constraints $\Sigma$ is a set of first-order formulas over $\sigma$. A database is consistent with respect to $\Sigma$ if it satisfies the formulas in $\Sigma$. Otherwise, the database is inconsistent. In this paper, we focus on the following constraints.

Definition 2.1. [Beeri and Vardi 1984; Lenzerini 2002] A tuple-generating dependency (tgdl) is a first-order formula of the form
\[
\forall x \exists y (\phi(x) \rightarrow \psi(x, y))
\]
where $\phi$ and $\psi$ are conjunctions of atomic formulas and $x$ and $y$ are tuples of variables. Such a tgd is a local-as-view dependency (LAV) if $\phi$ consists of a single atomic formula.

A global-as-view dependency (GAV) is a tgd of the form
\[
\forall x (\phi(x) \rightarrow R(x'))
\]
where $x$ and $x'$ are tuples of variables and the variables in $x'$ occur in $x$.

An equality-generating dependency (egd) is a first-order formula of the form
\[
\forall x (\phi(x) \rightarrow y = z)
\]
where $\phi$ is a conjunction of atomic formulas, $x$ is a tuple of variables and $y$ and $z$ are variables occurring in $x$.

A key constraint is a first-order formula of the form
\[
\forall x, y, z (R(x, y) \land R(x, z) \rightarrow y = z)
\]
where $x$ and $y$ and $z$ are tuples of variables\(^1\). Here, two tuples $y = (y_1, \ldots, y_n)$ and $z = (z_1, \ldots, z_n)$ are equal if $y_i = z_i$ for all $1 \leq i \leq n$.

For the sake of readability, we will drop the universal quantifiers when writing constraints.

\(^1\)Note that in order to simplify notations, we assumed that the variables in $x$ occur in the first positions of $R$. In general, this does not need to be the case.
Why is it hard to obtain a dichotomy for consistent query answering?

Tuple-generating-dependencies (tgds) and egds play a fundamental role in data exchange [Fagin et al. 2003; Arenas et al. 2014] and data integration [Lenzerini 2002]; they are used to express the relationship between a local source database and a global target database. Typically the relation symbols occurring on the left side of the implication of a tgd belong to the schema of the source database, while the symbols occurring on the right side belong to the schema of the target database. Hence, a tgd specifies how conditions verified by the source imply conditions on the target.

Among the class of tgds, two important subclasses have been extensively studied: the LAV (local-as-view) dependencies and the GAV dependencies. In the case of GAV, since only one relation symbol occurs on the right side of the implication, each relation of the target database is defined in terms of the relations in the source database. In the case of LAV, relations of the source are described in terms of the relations of the target.

Our main result is concerned with the problem of querying databases that do not satisfy a given set of GAV constraints. The approach of querying inconsistent databases introduced by Arenas et al [Arenas et al. 1999] has been developed around the notion of repair. Intuitively, a database is a repair of an inconsistent database if it satisfies the constraints and differs from the original database in a “minimal way”. Several notions of minimality have been introduced, giving rise to different definitions of repairs. Here, we opt for a standard notion of minimality, based on the set inclusion order. If \( D \) and \( E \) are databases, we denote by \( D \oplus E \) the symmetric difference of \( D \) and \( E \), i.e. the set \( D \setminus E \cup E \setminus D \).

**Definition 2.2 (Repair).** Let \( \Sigma \) be a set of constraints. A database \( E \) is a repair of a database \( D \) with respect to \( \Sigma \) if \( E \models \Sigma \) and there is no database \( E' \) such that \( E' \models \Sigma \) and \( E' \oplus D \subsetneq E \oplus D \).

The queries that we consider in this work are unions of conjunctive queries (UCQ). Recall that a conjunctive query (CQ) is a formula of the form

\[
q(x) = \exists y \phi(x,y)
\]

where \( \phi \) is a conjunction of atomic formulas. If a variable \( x \) occurs in \( x \) and not in \( y \), \( x \) is a free variable. A conjunctive query with constants is a CQ for which we allow the use of constants in the atomic formulas. We stick to the usual convention that the interpretation of a constant on a database is the constant itself. UCQs also correspond to the select-project-join-union fragment of relational algebra.

 Conjunctive queries are the most fundamental class of queries in database theory and form the core of all practical query languages. UCQs are disjunctions of conjunctive queries; they are easily seen to be equivalent to the existential and positive fragment of first-order logic.

A UCQ is boolean if it does not contain any free variable. If \( D \) is a database and \( q \) an UCQ, we denote by \( q(D) \) the set of tuples that belong to the evaluation of \( q \) over \( D \). The answers of a query on an inconsistent database \( D \) are obtained by evaluating the query over all the repairs of \( D \) and taking the intersection.

**Definition 2.3 (Consistent query answering).** Let \( \Sigma \) be a set of constraints, \( D \) a database and \( q \) a query. The consistent answers of \( q \) on \( D \) with respect to \( \Sigma \), denoted by \( CQA(q, D, \Sigma) \), is defined as the set

\[
\bigcap \{q(E) : E \text{ is a repair of } D \text{ with respect to } \Sigma\}.
\]

If \( q \) is a boolean query, we write \( CQA(q, D, \Sigma) = \top \) if \( q \) is true in all the repairs of \( D \) with respect to \( \Sigma \). Otherwise, \( CQA(q, D, \Sigma) = \bot. \)
The consistent query answering problem of \( q \) with respect to \( \Sigma \), denoted by \( CQA(q, \Sigma) \), is the following problem: given a database \( D \) and a tuple, determine whether the tuple is a consistent answer of \( q \) on \( D \) with respect to \( \Sigma \). We write \( CQA(q, \Sigma) \) for the following problem: given a database \( D \) and a tuple, determine whether the tuple is not a consistent answer of \( q \) on \( D \) with respect to \( \Sigma \).

As mentioned in the introduction, the complexity of consistent query answering under various classes of constraints has been deeply investigated since the late nineties. Since here we only consider constraints that are GAV, egds or keys, we simply recall that in each of those cases, the consistent query answering problem is known to be in \( \text{coNP} \) [Chomicki andMarcinkowski 2005; Staworko 2007].

The study of the complexity of consistent query answering was pushed further by investigating the problem of deciding the complexity of \( CQA(q, \Sigma) \). Although the original conjecture was stated for key constraints and conjunctive queries, we give here a more general formulation.

Definition 2.4 (Dichotomy conjecture). Let \( C \) be a class of constraints and let \( Q \) be a class of queries such that for all subsets \( \Sigma \) of \( C \) and for all queries \( q \in Q \), \( CQA(q, \Sigma) \) is in \( \text{coNP} \). The dichotomy conjecture with respect to \( C \) and \( Q \) states that for all subsets \( \Sigma \) of \( C \) and for all queries \( q \in Q \), \( CQA(q, \Sigma) \) is either in \( \text{PTIME} \) or is \( \text{coNP} \)-complete.

Conjecture 2.5. The dichotomy conjecture with respect to key constraints and conjunctive queries holds.

As mentioned earlier, the most recent contribution to the above conjecture is a dichotomy result for the case of CQs with two atoms and no self-joins [Kolaitis and Pema 2012].

2.2. Constraint satisfaction problem

An instance of the constraint satisfaction problem is defined by a set of values, a set of variables and a set of constraints, and asks whether there is a way to assign a value to each variable such that the constraints are satisfied. For our purpose, we adopt an equivalent formulation of the constraint satisfaction problem in terms of homomorphisms [Feder and Vardi 1998].

Recall that a map \( h : A \rightarrow B \) between two structures is an homomorphism if for all relation symbols \( R \) and for all \( (a_1, \ldots, a_n) \in R^A \), \( (h(a_1), \ldots, h(a_n)) \) belongs to \( R^B \).

Given a map \( h : A \rightarrow B \) and a tuple \( a = (a_1, \ldots, a_n) \) of elements in \( A \), we denote by \( h(a) \) the tuple \( (h(a_1), \ldots, h(a_n)) \). Moreover, we denote by \( A \) the domain of the structure \( A \) and by \( B \) the domain of \( B \).

Definition 2.6. Let \( B \) be a structure. The (non-uniform) constraint satisfaction problem \( \text{CSP}(B) \) is the following problem: given a structure \( A \), determine whether there is a homomorphism \( h : A \rightarrow B \).

The dichotomy conjecture for CSP states that for every structure \( B \), \( \text{CSP}(B) \) is either in \( \text{PTIME} \) or is \( \text{NP} \)-complete. It follows from various results [Jeavons et al. 1997; Bulatov et al. 2000] that this dichotomy is equivalent to the dichotomy for the pointed homomorphism problem.

Definition 2.7 (Pointed homomorphism problem). Let \( B \) be a structure. We define the pointed homomorphism problem \( \text{pHom}(B) \) as the following problem: given a structure \( A \) and a partial homomorphism \( f : A \rightarrow B \), determine whether there is a homomorphism \( g : A \rightarrow B \) extending \( f \). Recall that a partial homomorphism from \( A \) to \( B \) is an homomorphism from a substructure of \( A \) to \( B \).
Why is it hard to obtain a dichotomy for consistent query answering?

The dichotomy conjecture for the pointed homomorphism problems states that for every structure \( B \), \( pHom(B) \) is either in PTIME or is NP-complete.

It was shown [Jeavons et al. 1997] that if \( B' \) is the core of \( B \), then \( CSP(B) \) and \( CSP(B') \) are polynomially equivalent. The core of a structure \( B \) is the minimal substructure (with respect to inclusion) that is an homomorphic image of \( B \).

Moreover, Bulatov et al. [Bulatov et al. 2000] established that if \( B' \) is a core, \( CSP(B') \) is tractable (resp. NP-complete) iff \( pHom(B') \) is tractable (resp. NP-complete), where \( pHom(B') \) is the pointed homomorphism problem as defined below. Hence, in order to prove the dichotomy conjecture, we may restrict ourselves to the study the pointed homomorphism problem.

PROPOSITION 2.8. [Jeavons et al. 1997; Bulatov et al. 2000] The dichotomy conjecture for the pointed homomorphism problems holds iff the dichotomy conjecture for CSP holds.

Finally, we recall the dichotomy result proved by Bulatov [Bulatov 2003] for CSP over a schema containing all unary relations. In terms of homomorphisms, the result is formulated as follows.

Definition 2.9. Let \( B \) be a structure. The conservative homomorphism satisfaction problem \( cHom(B) \) is the following problem: given a structure \( A \) and given for each \( a \in A \), a set \( L_a \subseteq B \), determine whether there is a homomorphism \( h : A \rightarrow B \) such that \( h(a) \) belongs to \( L_a \) for all \( a \in A \).

THEOREM 2.10. [Bulatov 2003; Barto 2011] The dichotomy conjecture for the conservative homomorphism satisfaction problems holds.

Note that if \( L_a = A \) for all \( a \in A \), then there is a homomorphism \( h : A \rightarrow B \) such that \( h(a) \) belongs to \( L_a \) (for all \( a \in A \)) iff there is a homomorphism \( h : A \rightarrow B \). Hence, if the complexity of the problem \( cHom(B) \) is polynomial, so is the complexity of the problem \( CSP(B) \). However, the converse is not true. This is why the dichotomy result for the conservative homomorphism satisfaction problems, does not imply a dichotomy for CSP.

3. MAIN RESULT

Our main result establishes a connection between the dichotomy conjecture for CSP and the dichotomy conjecture for consistent query answering of UCQs with respect to GAV constraints.

THEOREM 3.1. If the dichotomy conjecture for consistent query answering of UCQs with respect to GAV constraints holds, then so does the dichotomy conjecture for the constraint satisfaction problems.

By Proposition 2.8, in order to prove Theorem 3.1, it is sufficient to show that if the dichotomy conjecture for consistent query answering of UCQs with respect to GAV constraints holds, then so does the dichotomy conjecture for the pointed homomorphism problems. This is a direct consequence of Proposition 3.2 proved below.

PROPOSITION 3.2. For each structure \( B \), we can compute a boolean UCQ \( q \) and a set \( \Sigma \) of GAV constraints such that \( pHom(B) \) and \( CQA(q, \Sigma) \) are polynomially equivalent, i.e. there is a polynomial reduction from \( pHom(B) \) to \( CQA(q, \Sigma) \) and vice-versa.
Note that the exact complexity of the computation of \( q \) and \( \Sigma \) (given the structure \( \mathfrak{B} \) over a schema \( \sigma \)) is irrelevant for us in order to infer \(^2\) Theorem 3.1. It is only important that

- \( p\text{Hom}(\mathfrak{B}) \) is tractable iff \( \overline{CQA}(q, \Sigma) \) is tractable,
- \( p\text{Hom}(\mathfrak{B}) \) is in \( \text{CONP} \) iff \( \overline{CQA}(q, \Sigma) \) is in \( \text{CONP} \).

This is guaranteed by the fact that \( p\text{Hom}(\mathfrak{B}) \) and \( \overline{CQA}(q, \Sigma) \) are polynomially equivalent.

**Proof.** Let \( \mathfrak{B} \) be a structure over a signature \( \sigma \). We define \( \Sigma \) and \( q \) over a schema \( \sigma' \) in the following way. The schema \( \sigma' \) consists of the following symbols

\[
\{N_b : b \in B\} \cup \{R, C_R : R \in \sigma\} \cup \{O, S\}
\]

where the \( N_b \)'s are unary, \( O \) and \( S \) are unary and \( R \) and \( C_R \) are of arity \( n \) if \( R \in \sigma \) is of arity \( n \).

Before defining \( \Sigma \) and \( q \), we give some intuition about the roles played by each constraint and by the query. For the sake of the explanation, we only focus on one reduction, from \( p\text{Hom}(\mathfrak{B}) \) to \( \overline{CQA}(q, \Sigma) \).

Suppose that we want to check for the existence of a homomorphism from a given structure \( \mathfrak{A} \) to the structure \( \mathfrak{B} \). We associate with \( \mathfrak{A} \) a database \( D \) that contains all the relations \( R^b \) and a unary relation \( S^O \) consisting of the domain of \( \mathfrak{A} \). Then, we will define the constraints \( \Sigma \) in such a way that each repair \( E \) of \( D \) encodes a partial map \( f^E : \mathfrak{A} \rightarrow \mathfrak{B} \). Moreover, if \( q \) is false in \( E \), this will ensure that \( f^E \) is a homomorphism and its domain is the domain of \( \mathfrak{A} \).

The way we encode a partial map in a repair \( E \) is by introducing a unary relation \( N_c \) for each \( c \in B \). The fact \( N_c(a) \) holds in a repair \( E \) if the map \( f^E \) sends the element \( a \) to an element that is not \( c \). For all \( b \in B \), we abbreviate the formula

\[
\bigwedge \{N_c(x) : c \in B, c \neq b\}
\]

by \( \phi_b(x) \). Hence, \( \phi_b(a) \) holds in a repair \( E \) if \( f^E \) maps \( a \) to \( b \).

Let \( R \) be a relation symbol of arity \( n \) and let \( b = (b_1, \ldots, b_n) \) be a tuple in \( B^n \). If \( R(b) \notin \mathfrak{B} \), we let \( \psi_{R(b)} \) be the following constraint

\[
\phi_{b_1}(x_1) \land \cdots \land \phi_{b_n}(x_n) \rightarrow C_R(x_1, \ldots, x_n)
\]

In the databases in which \( q \) (that we will define later) is false, we will think of \( C_R \) as being a subset of the complement of the relation \( R \). Hence, the meaning of the constraint \( \psi_{R(b)} \) is as follows. If \( f^E \) maps \( a_i \) to \( b_i \) (for all \( 1 \leq i \leq n \)) and \( R(b_1, \ldots, b_n) \) does not belong to \( \mathfrak{B} \), then the tuple \( (a_1, \ldots, a_n) \) must belong to the complement of \( R \). That is, the map \( f^E \) is a homomorphism.

For all \( b \in B \), we define \( \chi_b \) as the constraint

\[
\phi_b(x) \land S(x) \rightarrow O(x).
\]

In the databases in which \( q \) is false, the interpretation of \( O \) is the empty set. Recall that if \( \phi_b(a) \) holds in a repair \( E \) of a database \( D \), it means that the map \( f^E \) maps \( a \) to \( b \). The formulas \( \chi_b \)'s basically say that the set \( S^E \) and the domain of the map associated with \( E \) have an empty intersection.

Moreover, using the minimality condition of the repairs, we will show that this implies that those two sets not only have an empty intersection, but actually form a partition of \( S^O \).

\(^2\)Let us observe that it follows from the proof that the complexity of that computation is polynomial in the size of \( B \) and exponential in the maximal arity occurring in \( \sigma \).

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Next, we let \( \Sigma \) be the following set of constraints
\[
\{ \chi_b : b \in B \} \cup \{ \psi_{R(b)} : R(b) \notin R^B \}.
\]
We define \( q \) as the query
\[
\exists xO(x) \lor \exists xS(x) \lor \bigvee \{ \exists x(R(x) \land C_R(x)) : R \in \sigma \}.
\]
Given a database \( D \), the query \( q \) is false in a repair \( E \) iff \( O \) and \( S \) are empty in \( E \) and for all relation symbols \( R \), the intersection \( R \cap C_R \) is empty in \( E \). The fact that the intersection \( R \cap C_R \) is empty in \( E \) means that \( C_R \) is a subset of the complement of \( R \).

The intuition behind the fact that \( S \) is empty is a bit more complicated. Recall that in a repair \( E \) of a database \( D \) in which \( q \) is false, the constraints \( \chi_b \)\&s ensure that the set \( S^E \) and the domain of the map \( f^E \) form a partition of \( S^D \). In that case, the fact that \( S^E \) is empty means that all the elements of \( S^D \) have an image, or more informally, that the domain of the map associated with \( E \) is “big enough” for our purpose.

In order to prove that \( p\text{Hom}(B) \) and \( CQA(q, \Sigma) \) are polynomially equivalent, we have to show that
\[
\text{(a1) there is a polynomial reduction form } p\text{Hom}(B) \text{ to } CQA(q, \Sigma),
\]
\[
\text{(b1) there is a polynomial reduction from } CQA(q, \Sigma) \text{ to } p\text{Hom}(B).
\]
Before proving that (a) and (b) holds, we proceed with the following claims.

**Claim 1.** Let \( E \) be a repair of a database \( D \) with respect to \( \Sigma \). Then
\[
O^D \subseteq O^E, \quad S^E \subseteq S^D, \quad N_b^E \subseteq N_b^D, \quad R^E = R^D, \quad C_R^D \subseteq C_R^E,
\]
for all \( b \in B \) and all relation symbols \( R \). In particular, if \( E \models \phi_b(a) \), then \( D \models \phi_b(a) \).

**Proof.** Intuitively, the claim follows from the facts that \( R \) does not occur in \( \Sigma \), \( S \) and \( N_b \) only occur on the left sides of logical implications and \( C_R \) and \( O \) only occur on the right sides of logical implications.

Formally, let \( E \) be a repair of \( D \) with respect to \( \Sigma \). We define \( E_0 \) as the following database
\[
O^{E_0} = O^D \cup O^E, \quad S^{E_0} = S^D \cap S^E, \quad N_b^{E_0} = N_b^D \cap N_b^E, \quad R^{E_0} = R^D, \quad C_R^{E_0} = C_R^E \cup C_R^D
\]
for all \( b \in B \) and all relation symbols \( R \). We can check that if \( \Sigma \) is true in \( E_0 \), then \( \Sigma \) remains true in \( E_0 \). Moreover, \( D \oplus E_0 \subseteq D \oplus E \) by definition of \( E_0 \). We can conclude that \( E = E_0 \) since \( E \) is a repair of \( D \) with respect to \( \Sigma \). The claim follows. \( \Box \)

**Claim 2.** Let \( E \) be a repair of a database \( D \) with respect to \( \Sigma \). Suppose that \( q \) is false in \( E \). Let \( f \) be a map such that for all \( a \in \text{dom}(f) \) and for all \( b \in B \),
\[
D \models \phi_b(a) \iff b = f(a).
\]
We define \( \mathbb{A}^D \) as the induced structure with domain \( S^D \). Then there is a homomorphism \( g : \mathbb{A}^D \to \mathbb{B} \) such that for all \( a \in \text{dom}(f) \), \( g(a) = f(a) \).

**Proof.** We start by proving that
\[
\text{for all } a \in S^D, \text{ there is } b_a \in B \text{ s. t. } E \models \phi_{b_a}(a) \quad \text{(2)}
\]
and if \( a \in \text{dom}(f) \), then \( b_a = f(a) \). \( \text{(3)} \)
Let $a$ be an element of $D$. Suppose for contradiction that there is no $b$ such that $\phi_b(a)$ holds in $E$. We define $E_0$ as the instance obtained by adding the tuple $S(a)$ to the database $E$.

We prove that $\Sigma$ is true in $E_0$. Since $\Sigma$ is true in $E$ and $E_0$ is obtained by adding $S(a)$ to $E$, the constraints $\Sigma$ can only be false in $E_0$ if $$\phi_{b_0}(a) \land S(a) \rightarrow O(a)$$
is false in $E_0$, for some $b_0 \in B$. If this is the case, then $\phi_{b_0}(a)$ holds in $E_0$. By definition of $E_0$, this implies that $\phi_{b_0}(a)$ holds in $E$, which contradicts the fact that there is no $b$ such that $\phi_b(a)$ holds in $E$. Therefore, $\Sigma$ is true in $E_0$.

Since $q$ is false in $E$, $S^E$ is empty. Together with the fact that $S(a)$ holds in $D$ and $E_0$, this implies that $$D \oplus E_0 \subseteq D \oplus E.$$Since $\Sigma$ is true in $E_0$, this contradicts the fact that $E$ is a repair and this finishes the proof of (2).

Next we prove (3). Suppose that for some $a \in \text{dom}(f)$, we have $E \models \phi_{b_a}(a)$. By Claim 1, this implies that $D \models \phi_{b_a}(a)$. By (1), this can only happen if $b_a = f(a)$. This finishes the proof of (3).

It follows from (2) and (3) that we may pick a map $g : A^D \rightarrow \mathbb{B}$ such that

- $g(a) = f(a)$ for all $a \in \text{dom}(f)$
- $E \models \phi_{g(a)}(a)$ for all $a \in S^D$.

We prove that $g$ is a homomorphism.

Suppose for contradiction that $g$ is not a homomorphism. That is, there are a relation symbol $R$ of arity $n$ and a tuple $a = (a_1, \ldots, a_n)$ such that $R(a)$ holds in $A^D$ but $R(g(a))$ does not hold in $B^D$. By definition of $g$,

$$\phi_{g(a_1)}(a_1) \land \cdots \land \phi_{g(a_n)}(a_n)$$

holds in $E$. Since $\Sigma$ is true in $E$ and $R(g(a))$ does not belong to $B$, $\psi_{R(g(a))}$ given by

$$\phi_{g(a_1)}(x_1) \land \cdots \land \phi_{g(a_n)}(x_n) \rightarrow C_R(x_1, \ldots, x_n),$$
is true in $E$. Together with (4), we obtain that $C_R(a)$ holds in $E$.

Since $R(a)$ holds in $A^D$, the tuple $R(a)$ holds in $D$. By Claim 1, this implies that $R(a)$ holds in $E$. Putting everything together, we have

$$C_R(a) \in E \text{ and } R(a) \in E.$$This contradicts the fact that $q$ is false in $E$. $\square$

We start by proving (a1). That is, there is a polynomial reduction form $pHom(B)$ to $CQA(q, \Sigma)$. Let $A$ be a structure. Let $f$ be a partial homomorphism from $A$ to $B$. We let $D_0$ be the following database

$$\begin{align*}
S^{D_0} &= A, \\
O^{D_0} &= \emptyset, \\
N_b^{D_0} &= a \in \text{dom}(f) : f(a) \neq b \cup \text{dom}(f), \\
C_R^{D_0} &= A^n \setminus R^A, \\
R^{D_0} &= R^A,
\end{align*}$$

where $b \in B$ and $R$ is a relation symbol of arity $n$. In order to prove (a1), it is sufficient to show that

$$CQA(q, \Sigma, D_0) = \bot \quad \text{iff} \quad (A, f) \in pHom(B).$$

(5)
For the direction from left to right of (5), suppose that the consistent answer of $q$ is false. Let $E_b$ be a repair of $D_0$ with respect to $\Sigma$ in which $q$ is false. By Claim 2, there is an homomorphism $g_0 : A^{D_0} \rightarrow \mathbb{B}$ such that for all $a \in \text{dom}(f)$, $g_0(a) = f(a)$.

Hence, in order to prove that $(A,f)$ belongs to $p\text{Hom}(\mathbb{B})$, it is sufficient to prove that $\mathbb{A}^{D_0}$ is equal to $\mathbb{A}$. This follows from the definitions of $D_0$ and $\mathbb{A}^{D_0}$.

Now we show the direction from right to left of (5). Suppose that there is a homomorphism $g_1 : A \rightarrow \mathbb{B}$ such that $g_1(a) = f(a)$ for all $a \in \text{dom}(f)$. We define $F_0$ as the following database

$$
S^{F_0} = \emptyset,
O^{F_0} = \emptyset,
C^{F_0}_R = A^n \setminus R^A,
R^{F_0} = R^A,
N^{F_0}_b = \{ a \in A : g_1(a) \neq b \},
$$

where $b \in B$ and $R$ is a relation symbol of arity $n$. It is a simple exercise to prove that $\Sigma$ is true in $F_0$. Intuitively, each constraint $\chi_b$ is true because $S^{F_0}$ is empty. Each constraint $\psi_{R(b)}$ (where $R(b) \notin \mathbb{B}$) is true because $g_1$ is a homomorphism and $C^{F_0}_R$ contains the complement of $R^A$.

Since $\Sigma$ is true in $F_0$, there exists a repair $F_1$ of $D_0$ with respect to $\Sigma$ such that

$$
D_0 \oplus F_1 \subseteq D_0 \oplus F_0.
$$

(6)

We show that $q$ is false in $F_1$. This will imply that $CQA(q, \Sigma, D_0) = \bot$.

By definition, the query $q$ is false in $F_1$ iff $O^{F_1} = \emptyset$, $S^{F_1} = \emptyset$ and for all relation symbols $R$, $R^{F_1} \cap C^{F_1}_R$ is empty. Since $O^{F_0} = O^{D_0}$, it follows from (6) that $O^{F_1} = O^{D_0}$.

That is, $O^{F_1} = \emptyset$.

Next we prove that for all relation symbols $R$,

$$
R^{F_1} \cap C^{F_1}_R = \emptyset.
$$

(7)

Let $R$ be a relation symbol of arity $n$. Since $R^{F_0} = R^{D_0}$ and $C^{F_0}_R = C^{D_0}_R$, it follows from (6) that $R^{F_1} = R^{D_0}$ and $C^{F_1}_R = C^{D_0}_R$. This means that $R^{F_1} = R^A$ and $C^{F_1}_R = A^n \setminus R^A$.

It follows that (7) holds.

In order to prove that $q$ is false in $F_1$, it remains to show that $S^{F_1} = \emptyset$. Suppose for contradiction that $S(a)$ holds in $F_1$ for some $a \in A$. Since $g_1(a') = f(a')$ (for all $a' \in \text{dom}(f)$), it follows from the definition of $F_0$ and $D_0$ that $N^{F_0}_b \subseteq N^{D_0}_b$ for all $b \in B$.

Together with (6), this implies that for all $b \in B$,

$$
N^{F_0}_b \subseteq N^{F_1}_b.
$$

(8)

It also follows from the definition of $F_0$ that for all $a \in A$,

$$
\phi_{g_1(a)}(a) = \bigwedge \{ N_b(a) : b \in B, b \neq g_1(a) \}
$$

holds in $F_0$. Together with (8), we obtain that $\phi_{g_1(a)}(a)$ holds in $F_1$. Recall that we assume that $S(a)$ holds in $F_1$. Hence,

$$
\phi_{g_1(a)}(a) \land S(a)
$$

holds in $F_1$. Since $\chi_{g_1(a)}$ given by

$$
\phi_{g_1(a)}(x) \land S(x) \rightarrow O(x)
$$

is true in $F_1$, this implies that $O(a)$ holds in $F_1$, but this contradicts the emptiness of $O^{F_1}$ proved earlier. This finishes the proof that $S^{F_1} = \emptyset$. 

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Next we prove (b). That is, there is a polynomial reduction from $CQA(q, \Sigma)$ to $pHom(\mathcal{B})$. Let $D$ be a database. We define $X_0$ as the set
\[ \{ a : \text{for some } b \in B, D \models \neg N_b(a) \land \phi_b(a) \}. \]

Note that for all $a \in X_0$, there is a unique $b \in B$ such that $D \models \neg N_b(a) \land \phi_b(a)$. Indeed, suppose for contradiction that for some $c$ is distinct from $b$, we have $D \models \neg N_b(a) \land \phi_b(a)$ and $D \models \neg N_c(a) \land \phi_c(a)$. By definition of $\phi_c$ and since $b \neq c$, $D \models \phi_c(a)$ implies that $a$ belongs to $N^D_b$, which is a contradiction.

For all $a \in X_0$, we let $f^D(a)$ be the unique element $b \in B$ such that
\[ D \models \neg N_b(a) \land \phi_b(a). \]

Next, we define $X$ as the set
\[ \{ a : \text{for some } b \in B, D \models \phi_b(a) \}. \]

Note that $X_0 \subseteq X$ and if $a$ belongs to $X \setminus X_0$, then $D \models N_b(a)$ for all $b \in B$. We will make sure that the domain of structure associated with $D$ is a subset of $X$. Intuitively, $X_0$ contains the elements $a$ that can only be mapped to $f^D(a)$, while the elements in $X \setminus X_0$ can have an arbitrary image.

We define $\mathcal{A}^D$ as in Claim 2. That is, $\mathcal{A}^D$ is the induced substructure with domain $S^D$.

In order to prove (b1), we exhibit a set of three conditions (the satisfiability of which can be checked in polynomial time) such that if $D$ satisfies one of those conditions, then it is clear that the consistent answer of $q$ is true; and if $D$ does not satisfy any of those conditions, then
\[ (\mathcal{A}^D, f^D) \in pHom(\mathcal{B}) \quad \text{iff} \quad CQA(q, D, \Sigma) = \top. \]

This will show that there is a polynomial reduction from $CQA(q, \Sigma)$ to $pHom(\mathcal{B})$.

The three conditions are given by:

\begin{enumerate}
  \item [(C1)] for some relation symbol $R$, $R^D \cap C^D_R \neq \emptyset$,
  \item [(C2)] $S^D \setminus X \neq \emptyset$,
  \item [(C3)] $O^D \neq \emptyset$.
\end{enumerate}

We prove that
\begin{enumerate}
  \item [(A1)] if (C1), (C2) or (C3) holds, then $CQA(q, D, \Sigma) = \top$,
  \item [(B1)] if neither (C1) nor (C2) nor (C3) holds, then
\end{enumerate}
\begin{align*}
  (\mathcal{A}^D, f^D) \in pHom(\mathcal{B}) & \quad \text{iff} \quad CQA(q, D, \Sigma) = \bot \quad (9).
\end{align*}

We start by showing (A1). We pick a repair $G_0$ of $D$ with respect to $\Sigma$. Suppose that (C1) holds. That is, $R(a)$ and $C_R(a)$ belong to $D$ for some relation symbol $R$ and some tuple $a$. By Claim 1, this implies that $R(a)$ and $C_R(a)$ belong to $G_0$. Hence, $q$ is true in $G_0$.

Next suppose that (C2) holds. Suppose that there exists $a$ such that $S(a)$ holds in $D$ and $a$ does not belong to $X$. Let $G_1$ be the database obtained by adding the tuple $S(a)$ to the database $G_0$. We prove that $\Sigma$ is true in $G_1$.

Since $\Sigma$ is true in $G_0$ and $G_1$ is obtained from $G_0$ by adding $S(a)$, the only way for $\Sigma$ to be false in $G_0$ is if the constraint
\[ S(a) \land \phi_b(a) \rightarrow O(a) \]

is false in $\Sigma$, for some $b \in B$. Suppose that $S(a) \land \phi_b(a)$ holds in $G_1$ for some $b \in B$. Since $\phi_b(a)$ holds in $G_1$, it follows from Claim 1 that $\phi_b(a)$ holds in $D$. Hence, $a$ belongs to $X$, which is a contradiction. This finishes the proof that $\Sigma$ is true in $G_1$.  

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Why is it hard to obtain a dichotomy for consistent query answering?

It follows from definition of $G_1$ that $D \oplus G_1 \subseteq D \oplus G_0$. Since $G_0$ is a repair of $D$ with respect to $\Sigma$, this can only happen if $G_0 = G_1$. Hence, $S(a)$ holds in $G_0$. By definition of $q$, this implies that $q$ is true in $G_0$.

Now assume that (C3) holds. That is, $O^D \neq \emptyset$. By Claim 1, this means that $O^{G_0} \neq \emptyset$. Hence, $q$ is true in $G_0$.

Next we prove (B1). Suppose that neither (C1) nor (C2) nor (C3) holds. For the direction from right to left of (9), suppose that the consistent answer of $q$ is false. It follows from Claim 2 that there is a homomorphism $h_0 : \mathcal{A}^D \rightarrow \mathcal{B}$ such that $h_0(a) = f^D(a)$ for all $a \in \text{dom}(f^D)$. Hence, $(f^D, f^D)$ belongs to $\text{pHom} (\mathcal{B})$.

For the direction from left to right of (9), suppose that there is a homomorphism $h_1 : \mathcal{A}^D \rightarrow \mathcal{B}$ such that $h_1(a) = f^D(a)$ for all $a \in \text{dom}(f^D)$. We let $H_0$ be the following database

$$
R^{H_0} = R^D, \\
C_R^{H_0} = C_R^D \cup \{a \in (S^D)^n : h_1(a) \notin R^G\}, \\
S^{H_0} = \emptyset, \\
O^{H_0} = \emptyset, \\
N_{h_0} = \{a \in S^D : h_1(a) \neq b\},
$$

where $b \in B$ and $R$ is a relation symbol of arity $n$. It is easy to show that $\Sigma$ is true in $H_0$. Basically, this follows from the facts that $S^{H_0}$ is empty and that $h_1$ is a homomorphism. Since $\Sigma$ is true in $H_0$, there is a repair $H_1$ of $D$ with respect to $\Sigma$ such that $D \oplus H_1 \subseteq D \oplus H_0$.

We show that $q$ is false in $H_1$. We start by proving that $\exists x \phi(x)$ is false in $H_1$. Since (C3) does not hold, $O^D$ is empty. Together with $O^{H_0} = \emptyset$ and $D \oplus H_1 \subseteq D \oplus H_0$, we obtain that $O^{H_1} = \emptyset$.

Next we prove that $\exists x S(x)$ is false in $H_1$. Suppose that there is a fact $S(a)$ in $H_1$. We will derive a contradiction by showing that

$$
\phi_{h_1(a)}(a) \land S(a) \rightarrow O(a)
$$

is false in $H_1$, which is impossible as $H_1$ is a repair with respect to $\Sigma$.

We proved previously that $O^{H_1} = \emptyset$. We also assume that $S(a)$ holds in $H_1$. Hence, (10) is false iff $\phi_{h_1(a)}(a)$ holds in $H_1$. Since $D \oplus H_1 \subseteq D \oplus H_0$, in order to show that $\phi_{h_1(a)}(a)$ holds in $H_1$, it is sufficient to prove that

$$
H_0 \models \phi_{h_1(a)}(a) \text{ and } D \models \phi_{h_1(a)}(a).
$$

The fact that $\phi_{h_1(a)}(a)$ holds in $H_0$ follows from the definition of $H_0$. Next, we prove that $\phi_{h_1(a)}(a)$ is true in $D$.

Since $S^{H_0} = \emptyset$, $S(a)$ holds in $H_1$ and $D \oplus H_1 \subseteq D \oplus H_0$, $S(a)$ must belong to $D$. Since (C2) does not hold, $a$ belongs to $X$.

---

If $a$ belongs to $X \setminus X_0$, then $N_b(a) \text{ holds in } D \text{ for all } b \in B$. In particular, $\phi_{h_1(a)}(a)$ is true in $D$.

---

If $a$ belongs to $X_0$, then $\phi_{f^D(a)}(a) \text{ holds in } D$. Since $f^D(a') = h_1(a')$ for all $a' \in \text{dom}(f^D)$, this implies that $\phi_{h_1(a)}(a)$ is true in $D$.

---

This finishes the proof of (11) and the proof that $\exists x S(x)$ is false in $H_1$.

Since $O^{H_1} = \emptyset$ and $S^{H_1} = \emptyset$, in order to show that $q$ is false in $H_1$, it remains to prove that for all relation symbols $R$,

$$
\exists x (R(x) \land C_R(x))
$$

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
is false in \( H_1 \). Suppose for contradiction that there exists a tuple \( a \) such that \( R(a) \) and \( C_R(a) \) belong to \( H_1 \). We prove that this implies

\[
R(a) \in D \text{ and } C_R(a) \in H_0. \tag{12}
\]

By Claim 1, since \( R(a) \) holds in \( H_1 \), \( R(a) \) holds in \( D \). Since \( D \oplus H_1 \subseteq D \oplus H_0 \) and \( C_R^D \subseteq C_R^{H_0} \), we have \( C_R^{H_1} \subseteq C_R^{H_0} \). In particular, if \( C_R(a) \) holds in \( H_1 \), then \( C_R(a) \) holds in \( H_0 \). Hence, (12) holds.

Since (C1) does not hold, (12) can only happen if

\[
R(a) \in D \text{ and } C_R(a) \in H_0 \text{ and } C_R(a) \notin D.
\]

By definition of \( H_0 \), this means that \( h_1(a) \) does not belong to \( R^D \). Since \( h_1 \) is a homomorphism, it follows that \( R(a) \) does not belong to \( A^D \). This contradicts the fact that \( R(a) \) holds in \( D \). This completes the proof that (B1) holds, hence the proof of the existence of a polynomial reduction from \( \overline{CQA}(q, \Sigma) \) to \( pHom(\mathcal{B}) \).

4. OTHER RELATED RESULTS

As mentioned in the introduction, we were not able to adapt the proof of Proposition 3.2 to the setting of key constraints. However, if we restrict our attention to conservative CSP, we can prove a similar result.

**Theorem 4.1.** There is a key constraint \( \phi \) such that for each structure \( \mathcal{B} \), we can compute a boolean UCQ \( q \) using constants such that \( cHom(\mathcal{B}) \) and \( \overline{CQA}(q, \phi) \) are polynomially equivalent.

As a consequence, a dichotomy result for consistent query answering with respect to keys and UCQs with constants would provide an alternative proof for the dichotomy theorem for conservative CSP.

If we accept to trade keys for egds, we can prove a similar result without using constants in the queries.

**Theorem 4.2.** For each structure \( \mathcal{B} \), we can compute a boolean UCQ \( q \) and a set of egds \( \Sigma \) such that \( cHom(\mathcal{B}) \) and \( \overline{CQA}(q, \Sigma) \) are polynomially equivalent.

We provide now the proof of the two above results. We start with Theorem 4.1.

**Proof.** (of Theorem 4.1) Let \( \mathcal{B} \) be a structure over a signature \( \sigma \). We define \( \phi \) and \( q \) over a schema \( \sigma' \) in the following way. The schema \( \sigma' \) consists of the following symbols

\[
\{ F \} \cup \{ R : R \in \sigma \}
\]

where \( F \) is binary and \( R \) is of arity \( n \) if \( R \in \sigma \) is of arity \( n \).

Before defining \( q \) and \( \phi \), we give some intuition and for that purpose, we only focus on the reduction from \( cHom(\mathcal{B}) \) to \( \overline{CQA}(q, \phi) \). Fix a structure \( \mathcal{A} \) and a family \( \mathcal{L} = \{ L_a \subseteq B : a \in A \} \). Suppose that we want to check whether \( (\mathcal{A}, \mathcal{L}) \in cHom(\mathcal{B}) \).

We associate with \( (\mathcal{A}, \mathcal{L}) \) a database \( D \). The database \( D \) contains all the relations \( R^A \) and for each \( (a, b) \) such that \( b \in L_a \), \( D \) contains the fact \( F(a, b) \). In other words, the presence of \( F(a, b) \) in \( D \) means that we are allowed to map \( a \) to \( b \). The key \( \phi \) is defined in such a way that each repair \( E \) of the database encodes a map \( f^E : A \to \mathcal{B} \) such that \( f^E(a) = b \) iff \( F(a, b) \in E \). So the key must express that for each \( a \), there is at most one element \( b \) such that \( F(a, b) \in E \). We let \( \phi \) be the following key

\[
F(x, u) \land F(x, v) \to u = v.
\]

Next, the query \( q \) is defined such that \( q \) is false in a repair \( E \) iff then \( f^E \) is a homomorphism. For all \( R(b) \) with \( b = (b_1, \ldots, b_n) \), we let \( q_{R(b)} \) be the following conjunctive
Why is it hard to obtain a dichotomy for consistent query answering?

query

\[ \exists x_1, \ldots, x_n(R(x_1, \ldots, x_n) \land F(x_1, b_1) \land \cdots \land F(x_n, b_n)). \]

We define \( q \) by

\[ \bigvee \{ q_{R(b)} : R(b) \notin \mathbb{B} \}. \]

We will show that \( q \) is false in a repair \( E \) iff \( f^E \) is an homomorphism. This finishes the definition of \( q \) and \( \phi \). Now we show that \( cHom(\mathbb{B}) \) and \( \overline{CQA}(q, \phi) \) are polynomially equivalent. That is, we have to prove

(a2) there is a polynomial reduction from \( cHom(\mathbb{B}) \) to \( \overline{CQA}(q, \phi) \),
(b2) there is a polynomial reduction from \( \overline{CQA}(q, \phi) \) to \( cHom(\mathbb{B}) \).

The proof that (a2) and (b2) hold is based on the following claim. Given a database \( D \), we define \( A^D \) as the set

\[ \{ a : \text{for some } b \in B, F(a, b) \in D \} \]

and we define \( \mathbb{A}^D \) as the induced substructure with domain \( A^D \). For all \( a \in A^D \), let \( L^D_a \) be the set \( \{ b \in B : F^D(a, b) \} \) and we let \( L^D \) be the set \( \{ L^D_a : a \in A^D \} \).

**Claim 3.** Let \( D \) be a database. We assume that \( \mathbb{A}^D \) and \( L^D \) are defined as above. Then,

\[ CQA(q, D, \phi) = \bot \iff (\mathbb{A}^D, L^D) \in cHom(\mathbb{B}). \]  

(13)

**Proof.** Suppose first that \( A^D \) is empty. Then it is clear that \( (\mathbb{A}^D, L^D) \) belongs to \( cHom(\mathbb{B}) \). Moreover, it can easily be seen that in case \( A^D \) is empty, \( CQA(q, D, \phi) \) is false.

So assume that \( A^D \) is not empty. For the implication from left to right, suppose that \( CQA(q, D, \phi) = \bot \). Hence, there is a repair \( E \) of \( D \) such that \( E \not\models q \). First, we show that for all \( a \in A^D \), there is a unique \( b \) such that \( F(a, b) \in E \).

\[ \text{(14)} \]

Since \( \phi \) is true in \( E \), for all \( a \in A^D \), there is at most one element \( b \) such that \( F(a, b) \) holds in \( E \).

Next suppose for contradiction that for some \( a \in A^D \), there is no \( b \) such that \( F(a, b) \) holds in \( E \). By definition of \( A^D \), there exists \( b_0 \in B \) such that \( F(a, b_0) \) holds in \( D \). We let \( E_0 \) be the database obtained from the database \( E \) by adding the tuple \( F(a, b_0) \). The key constraint \( \phi \) remains true in \( E_0 \), and moreover, \( E \subseteq E_0 \subseteq D \). This contradicts the fact that \( E \) is a repair of \( D \). This finishes the proof of (14).

It follows that there is a unique map \( f : \mathbb{A}^D \rightarrow \mathbb{B} \) such that

\[ F(a, f(a)) \text{ holds in } E \text{ for all } a \in \mathbb{A}. \]  

(15)

In order to show that \( (\mathbb{A}^D, L^D) \) belongs to \( cHom(\mathbb{B}) \), it is sufficient to prove that \( f(a) \) belongs to \( L^D_a \) for all \( a \in A^D \) and \( f \) is a homomorphism.

We start by proving that \( f(a) \) belongs to \( L^D_a \) for all \( a \in A^D \). Let \( a \) be an element of \( A^D \). Since \( F(a, f(a)) \) belongs to \( E \), this implies that \( f(a) \) also belongs to \( D \). Therefore, \( f(a) \) belongs to \( L^D_a \).

Next we prove that \( f \) is a homomorphism. Suppose for contradiction that \( f \) is not a homomorphism. That is, there is a tuple \( a = (a_1, \ldots, a_n) \) and a relation symbol \( R \) such that \( R(a) \) holds in \( \mathbb{A}^D \) and \( R(f(a)) \) does not belong to \( \mathbb{B} \). By definition of \( \mathbb{A}^D \), if \( R(a) \) holds in \( \mathbb{A}^D \), then \( R(a) \) belongs to \( D \). Since \( R \) does occur in the constraint \( \phi \), this implies that \( R(a) \) holds in \( E \). Together with (15), we obtain

\[ E \models R(a) \land F(a_1, f(a_1)) \land \cdots \land F(a_n, f(a_n)). \]
That is, $q_{R(f(a))}$ is true in $E$. Since $R(f(a))$ does not belong to $\mathbb{B}$, this implies that $q$ is true in $E$, which is a contradiction. This finishes the proof that $f$ is a homomorphism.

We show now the implication from right to left of (13). Assume that there is a homomorphism $g : A^D \to \mathbb{B}$ such that for all $a \in A^D$, $g(a) \in L_a^D$. We define $X^D$ as the set

$$\{ r \notin A^D : \text{for some } s, F(r, s) \in D \}.$$

We pick an arbitrary map $h$ with domain $X^D$ such that for all $a \in X^D$, $F^D(a, h(a))$ holds. We define the database $G$ by

$$F^G = \{(a, g(a)) : a \in A^D \} \cup \{(a, h(a)) : a \in X^D \},$$

$$R^G = R^D,$$

for all relation symbols $R$. The database $G$ is a repair of $D$ with respect to $\phi$. Hence, in order to prove the implication from right to left of (13), it is sufficient to show that $q$ is false in $G$.

Suppose for contradiction that $q$ is true in $G$. By definition of $q$, this means that there is a tuple $R(b) \notin \mathbb{B}$ with $b = (b_1, \ldots, b_n)$ such that $q_{R(b)}$ is true in $G$. That is, there exists $a_1, \ldots, a_n$ such that

$$G \models R(a_1, \ldots, a_n) \land F(a_1, b_1) \land \ldots \land F(a_n, b_n).$$

We prove that this implies that

$$R(a_1, \ldots, a_n) \in A^D \text{ and } R(g(a_1), \ldots, g(a_n)) \notin \mathbb{B},$$

(16)

which contradicts the fact that $g$ is a homomorphism. For all $1 \leq i \leq n$, since $b_i$ belongs to $B$ and $F(a_i, b_i)$ belongs to $G$, $a_i$ belongs to $A^D$. Since $(a_1, \ldots, a_n)$ belongs to $(A^D)^n$ and $R(a_1, \ldots, a_n)$ holds in $G$, $R(a_1, \ldots, a_n)$ holds in $A^D$.

In order to prove (16), it remains to show that $R(g(a_1), \ldots, g(a_n))$ does not belong to $\mathbb{B}$. Recall that we proved that $a_i$ belongs to $A^D$ for all $1 \leq i \leq n$. By definition of $F^G$, if $a_i$ belongs to $A^D$ and $F(a_i, b_i)$ holds in $G$, then $b_i = g(a_i)$. Recall also that $R(b)$ does not belong to $\mathbb{B}$. Together with $b_i = g(a_i)$, this implies that $R(g(a_1), \ldots, g(a_n))$ does not belong to $\mathbb{B}$. $\square$

Now that we finished the proof of the claim, we start properly the proof of the fact that $cHom(\mathbb{B})$ and $CQA(q, \phi)$ are polynomially equivalent. First we prove that there is a polynomial reduction from $cHom(\mathbb{B})$ to $CQA(q, \phi)$. Let $A$ be a structure and for all $a \in A$, let $L_a$ be a subset of $B$. We let $L$ be the set $\{ L_a : a \in A \}$. Without loss of generality, we may assume that $L_a \neq \emptyset$ for all $a \in A$. We define a database $D_0$ by

$$F^D_0 = \{(a, b) \in A \times B : b \in L_a \},$$

$$R^D_0 = R^A,$$

for all relation symbols $R$. In order to prove (a), it is sufficient to show that

$$CQA(q, D_0, \phi) = \bot \iff (A, L) \in \text{cHom}(\mathbb{B}).$$

(17)

It follows from the claim that

$$CQA(q, D_0, \phi) = \bot \iff (A^D_0, L^D_0) \in \text{cHom}(\mathbb{B}).$$

It also follows from the definition of $D_0$ that $A^D_0 = A$ and $L^D_a = L_a$ for all $a \in A$. Together with the previous equivalence, we obtain (17). This finishes the proof of the existence of polynomial reduction from $cHom(\mathbb{B})$ to $CQA(q, \phi)$. 

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Why is it hard to obtain a dichotomy for consistent query answering?

Next we prove that there is a polynomial reduction from $CQA(q, \phi)$ to $cHom(\mathbb{B})$. Let $D$ be a database. It follows from the previous claim that

$$CQA(q, D, \phi) = \perp \iff (\mathbb{A}^D, \mathcal{L}^D) \in cHom(\mathbb{B}).$$

This implies that there is a polynomial reduction from $CQA(q, \phi)$ to $cHom(\mathbb{B})$.

We prove now Theorem 4.2. Recall that Theorem 4.2 is the following result. For each structure $\mathbb{B}$, we can compute a boolean UCQ $q$ and a set of egds $\Sigma$ such that $cHom(\mathbb{B})$ and $CQA(q, \Sigma)$ are polynomially equivalent.

**Proof.** (of Theorem 4.2) Let $\mathbb{B}$ be a structure over a signature $\sigma$. We define $\Sigma$ and $q$ over a schema $\sigma'$ in the following way. The schema $\sigma'$ consists of the following symbols

$$\{ F_b : b \in B \} \cup \{ R : R \in \sigma \} \cup \{ Q \}$$

where $Q$ is unary, $F_b$ is unary and $R$ is of arity $n$ if $R \in \sigma$ is of arity $n$.

We give some intuition about the constraints $\Sigma$ and the query $q$ that we introduce, and we focus first on the reduction from $cHom(\mathbb{B})$ to $CQA(q, \Sigma)$. Fix a structure $\mathbb{A}$ and a family $\mathcal{L} = \{ L_a : a \in A \}$. We want to check whether $(\mathbb{A}, \mathcal{L})$ belongs to $cHom(\mathbb{B})$.

We define a database $D^\mathbb{A}$ containing the facts $R(a)$ for all $a \in R^\mathbb{A}$ and the facts $F_b(a)$, where $a \in A$ and $b \in L_a$. Moreover, $D^\mathbb{A}$ contains a special fact $Q(\perp_0)$ where $\perp_0 \notin A \cup B$.

The idea is that in each repair $E$, either $Q^E$ is empty or $E$ encodes a map $f^E : \mathbb{A} \to \mathbb{B}$ such that $f^E(a) = b$ iff $F_b(a) \in E$.

If $Q^E = \emptyset$, the way we ensure that $E$ encodes a map is by introducing for all $b, c \in B$ such that $b \neq c$, the egd $\phi_{b,c}$ given by

$$F_b(x) \land F_c(x) \land Q(y) \to x = y.$$ 

Since $Q^E$ is not empty and for all $b$, $Q^E \cap F_b^E = \emptyset$, the constraints $\phi_{b,c}$ express that for each $a$, there is at most one $b$ such that $F_b(a)$ holds in $E$. If $Q^E$ consists of exactly one element and for all $b$, $Q^E \cap F_b = \emptyset$, we say that $Q^E$ is well-behaved.

In general, if we are given an arbitrary database $D$ (and not a database of the form $D^\mathbb{A}$), there is no guarantee that in each repair $E$ of $D$, either $Q^E$ is empty or $Q^E$ is well-behaved. We enforce this by introducing the following constraint and query. We let $\phi$ be the egd given by:

$$Q(x) \land Q(y) \to x = y.$$ 

The egd $\phi$ ensures that $Q$ has at most one element in each repair. Next, we define $q_1$ as the query

$$\bigvee \{ \exists x(Q(x) \land F_b(x)) : b \in B \}.$$ 

If a repair $E$ satisfies $\phi$ and falsifies $q_1$, then either $Q^E = \emptyset$ or $Q^E$ is well-behaved.

Next we introduce a query $q_2$ such that $q_2$ is false in a repair $E$ encoding a map $f^E$ as defined above) iff $f^E$ is an homomorphism. For all $R(b)$ with $b = (b_1, \ldots, b_n)$, we let $q_{R(b)}$ be the following conjunctive query

$$\exists x_1, \ldots, x_n(R(x_1, \ldots, x_n)) \land F_{b_1}(x_1) \land \cdots \land F_{b_n}(x_n)$$

and we let $q_2$ by the query given by

$$\bigvee \{ q_{R(b)} : R(b) \notin \mathbb{B} \}.$$ 

Let $E$ be a repair for which there is a map $f^E : \mathbb{A} \to \mathbb{B}$ such that $f^E(a) = b$ iff $F_b(a) \in E$. We can prove that $q_2$ is true in $E$ iff $f^E$ is not a homomorphism.
Finally, we define $\Sigma$ as the set of constraints 
\[ \{ \phi_{b,c} : b, c \in B, b \neq c \} \cup \{ \phi \} \]
and we let $q$ be the query $q_1 \lor q_2$. To summarize our informal intuition: in the repairs $E$ of a database $D$ in which $Q^E \neq \emptyset$ and $q_1$ is false, $Q^E$ is well-behaved, $E$ encodes a map $f^E : A \to B$ and $f^E$ is a homomorphism iff $q_2$ is false.

Now we prove formally that $\text{cHom}(B)$ and $\mathcal{CQA}(q, \Sigma)$ are polynomially equivalent. That is, we have to show that

(a3) there is a polynomial reduction from $\text{cHom}(\text{str}B)$ to $\mathcal{CQA}(q, \Sigma)$,

(b3) there is a polynomial reduction from $\mathcal{CQA}(q, \Sigma)$ to $\text{cHom}(B)$.

We now proceed with the proof that (a3) and (b3) hold. We start with the following Claim. Given a database $D$, we define $A^D$ as the set
\[ F_{b_1}^D \cup \cdots \cup F_{b_k}^D. \]
and we define $\mathcal{A}^D$ as the structure with domain $A^D$ and
\[ R_{\mathcal{A}^D} = R^D \cap (A^D)^n, \]
for all relation symbols $R$ of arity $n$.

**Claim 4.** Let $D$ be a database. The structure $\mathcal{A}^D$ is defined as above. Assume that $Q^D \neq \emptyset$ and $\Sigma$ is true in $D$. Then, if $q_1$ is false in $D$, there is a unique map $f^D : \mathcal{A}^D \to B$ such that for all $a \in A^D$, $F_b(a)$ holds in $D$, where $b = f^D(a)$. Moreover,
\[ f^D \text{ is a homomorphism } \iff D \models q_2. \]

**Proof.** It follows from the definition of $\mathcal{A}^D$ that there is a map $f : \mathcal{A}^D \to B$ such that for all $a \in A^D$, $F_b(a)$ holds in $D$. Suppose that $q_1$ is false in $D$. We prove that such a map is uniquely defined. If this is not the case, there exist $a \in A^D$ and $b, c \in B$ such that $b \neq c$ and $F_b(a)$ and $F_c(a)$ belong to $D$. Since $Q^D \neq \emptyset$, we can pick $\bot_0$ such that $\bot_0 \in Q^D$. Thus,
\[ F_b(a) \land F_c(a) \land Q(\bot_0) \]
holds in $D$. Since $\Sigma$ is true in $D$, this implies that $a = \bot_0$. That is,
\[ D \models F_b(a) \land Q(a). \]
This contradicts the fact that $q_1$ is false in $D$.

Next we prove that
\[ D \models q_2 \iff f \text{ is not a homomorphism.} \quad (18) \]
Hence, we may define $f^D$ as the map $f$.

The formula $q_2$ is true in $D$ iff there is a tuple $R(b) \notin B$ with $b = (b_1, \ldots, b_n)$ such that $q_{R(b)}$ is true in $D$. The query $q_{R(b)}$ is true in $D$ iff there exists a tuple $a = (a_1, \ldots, a_n)$ such that
\[ D \models R(a_1, \ldots, a_n) \land F_{b_1}(a_1) \land \cdots \land F_{b_n}(a_n). \]
Observe that since $F_{b_i}(a_i) \in D$, the element $a_i$ belongs to $A^D$ for all $1 \leq i \leq n$. Hence, by definition of $\mathcal{A}^D$,
\[ R(a_1, \ldots, a_n) \in D \iff R(a_1, \ldots, a_n) \in \mathcal{A}^D. \]
Moreover, it follows from the unicity of $f$ that $F_{b_i}(a_i) \in D$ iff $b_i = f(a_i)$ for all $1 \leq i \leq n$. 

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Putting everything together, we obtain that \( q_2 \) is true in \( D \) iff there are a tuple \( R(b) \notin B \) and a tuple \((a_1,\ldots,a_n)\) such that
\[
R(a_1,\ldots,a_n) \in \mathbb{A}^D, \quad f(a_1) = b_1, \ldots, f(a_n) = b_n,
\]
where \( b = (b_1,\ldots,b_n) \). This happens iff \( f \) is not a homomorphism. \( \square \)

**Claim 5.** Let \( E \) be a repair of a database \( D \) with respect to \( \Sigma \) such that \( Q^E = \emptyset \). Then, \( R^E = R^D \) and \( F^E_b = F^D_b \) for all \( b \in B \) and for all relation symbols \( R \).

**Proof.** Let \( G \) be the following database
\[
\begin{align*}
Q^G &= \emptyset, \\
R^G &= R^D, \\
F^G_b &= F^D_b,
\end{align*}
\]
where \( b \in B \) and \( R \) is a relation symbol. Since \( Q^E = \emptyset \), we have \( E \subseteq G \subseteq D \). Moreover, since \( Q^G = \emptyset \), it is easy to check that \( \Sigma \) is true in \( G \). As \( E \) is a repair of \( D \) with respect to \( \Sigma \), this can only happen if \( E = G \). The claim follows. \( \square \)

Now that we finished the proof of the two claims, we start properly the proof of the fact that \( c\text{Hom}(B) \) and \( \text{CQA}(q,\Sigma) \) are polynomially equivalent. First we show that there is a polynomial reduction from \( c\text{Hom}(B) \) to \( \text{CQA}(q,\Sigma) \). Let \( \mathbb{A} \) be a structure and for all \( a \in A \), let \( L_a \) be a subset of \( A \). Without loss of generality, we may assume that \( L_a \neq \emptyset \) for all \( a \in A \). We let \( \mathcal{L} \) be the set \( \{ L_a : a \in A \} \). We define a database \( D_0 \) by
\[
\begin{align*}
Q^{D_0} &= \{ \bot_0 \}, \\
F^{D_0}_b &= \{ a \in A : b \in L_a \}, \\
R^{D_0} &= R^\mathbb{A},
\end{align*}
\]
where \( b \in B \) and \( R \) is a relation symbol. In order to show (a), it is sufficient to prove that
\[
\text{CQA}(q, D_0, \Sigma) = \bot \quad \text{iff} \quad (\mathbb{A}, \mathcal{L}) \in c\text{Hom}(B). \tag{19}
\]
Suppose that \( \text{CQA}(q, D_0, \Sigma) = \bot \). Hence, there is a repair \( E_0 \) of \( D_0 \) such that \( E_0 \not\vDash q \). We make the following case distinction.

---
Suppose that \( Q^{E_0} = \emptyset \). Let \( f : \mathbb{A} \to \mathbb{B} \) be an arbitrary map such that for all \( a \in A \), \( f(a) \in L_a \). We prove that \( f \) is a homomorphism. Suppose for contradiction that \( f \) is not a homomorphism. Hence, there are tuples \( b = (b_1,\ldots,b_n) \) and \( a = (a_1,\ldots,a_n) \) such that
\[
a \in R^\mathbb{A}, \quad b \notin R^\mathbb{B} \quad \text{and} \quad f(a_i) = b_i,
\]
for all \( 1 \leq i \leq n \). As \( f(a_i) \) and \( b_i \) are equal, \( b_i \) belongs to \( L_{a_i} \). By definition of \( F^{D_0}_{b_i} \), this implies that \( F^{D_0}_{b_i}(a_i) \in D_0 \). Together with the facts \( a \in R^\mathbb{A} \) and \( R^{D_0} = R^\mathbb{A} \), we obtain
\[
D_0 \vDash R(a_1,\ldots,a_n) \land F_{b_i}(a_1) \land \ldots \land F_{b_i}(a_n).
\]
Since \( Q^{E_0} = \emptyset \), by Claim 5, this implies that
\[
E_0 \vDash R(a_1,\ldots,a_n) \land F_{b_i}(a_1) \land \ldots \land F_{b_i}(a_n).
\]
That is, \( q_{R(b)} \) is true in \( E_0 \). Since \( b \notin R^\mathbb{B} \), this implies that \( q \) is true in \( E_0 \), which is a contradiction.
Next suppose that $Q^{E_0} \neq \emptyset$. It follows from Claim 4 that there is a homomorphism $f^{E_0} : \mathcal{A}^{E_0} \to B$ such that $F_b(a)$ holds in $E_0$, for all $a \in A^{E_0}$ and where $b = f^{E_0}(a)$. Hence, in order to prove that $(\mathcal{A}, \mathcal{L})$ belongs to $c\text{Hom}(\mathcal{B})$, it is sufficient to show that

$$\mathcal{A}^{E_0} = \mathcal{A} \quad \text{and for all } a \in A, f^{E_0}(a) \in L_a.$$ 

We prove that for all $a \in A$, $f^{E_0}(a)$ belongs to $L_a$. Let $a$ be an element of $A$ and let $b$ be the image $f^{E_0}(a)$. Since $F_b(a)$ holds in $E_0$ and $E_0$ is a subset of $D_0$, $F_b(a)$ holds in $D_0$. By definition of $F^{D_0}$, $b$ belongs to $L_a$.

Next we show that $\mathcal{A}^{E_0} = \mathcal{A}$. By definition of $\mathcal{A}^{E_0}$, this is equivalent to show that $A^{E_0}$ is equal to $A$. Since $E_0$ is a subset of $D_0$, it is immediate that $A^{E_0}$ is a subset of $A^{D_0}$. Moreover, since $F^D_0$ is a subset of $A$ (for all $b$), it is clear that $A^{D_0}$ is a subset of $A$. Hence, $A^{E_0}$ is a subset of $A$.

Now suppose for contradiction that $A^{E_0}$ is a proper subset of $A$. That is, for some $a \in A$, there is no $b$ such that $F_b(a)$ holds in $E_0$. Since $L_a \neq \emptyset$, there exists $b_0 \in A$ such that $F_{b_0}(a)$ holds in $D_0$. We let $E_1$ be the database obtained from the database $E_0$ by adding the tuple $F_{b_0}(a)$. The constraint $\Sigma$ remain true in $E_1$ and moreover, $E_0 \subsetneq E_1 \subseteq D_0$. This contradicts the fact that $E_0$ is a repair of $D_0$. This finishes the proof that $\mathcal{A}^{E_0} = \mathcal{A}$.

We show now the implication from right to left of (19). Assume that there is a homomorphism $g : \mathcal{A} \to \mathcal{B}$ such that for all $a \in A$, $g(a) \in L_a$. We have to find a repair $G_0$ of $D_0$ with respect to $\Sigma$ in which $q$ is false. We define the database $G_0$ by

$$Q^{G_0} = \{ \perp_0 \},$$

$$F_b^{G_0} = \{ a \in A : g(a) = b \},$$

$$R^{G_0} = R^{D_0},$$

where $b \in B$ and $R$ is a relation symbol. The instance $G_0$ is a repair of $D_0$ with respect to $\Sigma$. We show that $q$ is false in $G_0$.

Since $Q^{G_0} \cap F_b^{G_0}$ is empty (for all $b$), $q_1$ is false in $G_0$. Next we prove that $q_2$ is false in $G_0$. Since $q_1$ is false in $G_0$ and $Q^{G_0} \neq \emptyset$, it follows from Claim 4, that in order to prove that $q_2$ is false in $G_0$, it is enough to show that $f^{G_0}$ is a homomorphism. As $g$ is a homomorphism, it is sufficient to prove that $f^{G_0} = g$. Recall that $f^{G_0}$ is the unique map such that $F_b(a)$ holds in $G_0$, for all $a \in A^{G_0}$ and where $b = f^{G_0}(a)$. By definition of $G_0$,

$$F_{g(a)}(a) \in G_0 \quad \text{for all } a \in A.$$ 

Hence, $f^{G_0} = g$ and this finishes the proof that $q$ is false in $G_0$.

Next we prove (b3). That is, there is a polynomial reduction from $\overline{CQA}(q, \Sigma)$ to $c\text{Hom}(\mathcal{B})$. Let $D_1$ be a database. We let $\mathcal{A}^{D_1}$ and $\{ L_a^{D_1} : a \in A^{D_1} \}$ be as defined in Claim 4. That is,

$$A^{D_1} = \bigcup \{ F_b : b \in B \},$$

$$R^{\mathcal{A}^{D_1}} = R^{D_1} \cap (A^{D_1})^n,$$

$$L_a = \{ b \in B : F_b(a) \in D_1 \},$$

where $R$ is a relation symbol of arity $n$ and $a \in A$. In order to make notation easier, we abbreviate $A^{D_1}$ by $A^1$, $R^{\mathcal{A}^{D_1}}$ by $R^1$ and $L_a^{D_1}$ by $L_a^1$. We let $L^1$ be the set $\{ L_a^1 : a \in A^1 \}$.

In the proof we may use the notion of $Q$-compatibility which we define as follows. We say that an element $x$ is $Q$-compatible if $x$ belongs to $Q^{D_1}$ and for all $a \in A^1 \setminus \{ x \}$,
there is a unique $b$ such that $F_b(a)$ holds in $D_1$. The intuition behind the notion of $Q$-compatibility is as follows: a database $D$ admits a $Q$-compatible element iff in each repair $E$, $Q^E$ is not empty. We prove this property later.

It is clear that $CQA(q, D_1, \Sigma) = \perp$ iff

- either there is a repair $H_1$ such that $Q^{H_1} = \emptyset$ and $H_1 \neq q$,
- or there is a repair $H_2$ such that $Q^{H_2} \neq \emptyset$ and $H_2 \neq q$.

We will show that

(A3) [there is a repair $H_1$ such that $H_1 \neq q$ and $Q^{H_1} = \emptyset$ if $D_1 \neq q_2$ and there is no $Q$-compatible $x$],

(B3) [there is a repair $H_2$ such that $H_2 \neq q$ and $Q^{H_2} \neq \emptyset$ iff $[(A^1, L^1) \in cHom(\mathcal{B})$ and $D_1 \models \exists x(Q(x) \land \neg F_{b_1}(x) \land \cdots \land \neg F_{b_k}(x))]$. (20)

Recall that $\{b_1, \ldots, b_k\}$ is the domain of $\mathcal{B}$.

Provided that (A3) and (B3) hold, we obtain that $CQA(q, D_1, \Sigma) = \perp$ iff

- either $D_1 \neq q_2$ and there is no $x$ $Q$-compatible,
- or $(A^1, L^1) \in cHom(\mathcal{B})$ and (20) holds.

Since checking for the existence of a $Q$-compatible element, the satisfaction of (20) and the fact that $D \neq q_2$ can be performed in polynomial time, this equivalence shows that there is a polynomial reduction from $CQA(q, \Sigma)$ to $cHom(\mathcal{B})$.

Hence, in order to prove (b3), it is sufficient to show that (A3) and (B3) hold. We start by proving that (A3) holds. We do so by showing that

(i) there is a repair $H_1$ such that $Q^{H_1} = \emptyset$ iff there is no $Q$-compatible element,

(ii) if $H_1$ is a repair such that $Q^{H_1} = \emptyset$, $H_1 \neq q$ iff $D_1 \neq q_2$.

First we prove (i). For the direction from left to right, let $H_1$ be a repair such that $Q^{H_1} = \emptyset$ and suppose for contradiction that there is an element $\bot_1$ that is $Q$-compatible. We define $I_1$ as the database obtained by adding the fact $Q(\bot_1)$ to $H_1$.

Using the fact that $\bot_1$ is $Q$-compatible, we check that the constraints $\Sigma$ remain true in $I_1$. Since $\bot_1$ is the only element in $Q^{\bot_1}$, the egd $\phi$ is true in $I_1$. Next let $b, c \in B$ be such that $b \neq c$. We have to prove that $\phi_{b, c}$ is true in $I_1$. Suppose that $I_1 \models F_b(a) \land F_c(a) \land Q(a')$. (21)

We have to prove that $a = a'$. Suppose for contradiction that $a \neq a'$. By definition of $I_1$, we have that $I_1 \models Q(a')$ implies $a' = \bot_1$. Hence, $a \neq \bot_1$. Together with the fact that $\bot_1$ is $Q$-compatible, this implies that there is a unique element $b_0 \in B$ such that $F_{b_0}(a) \in D_1$. Since $I_1$ is a subset of $D_1$, (21) implies that $F_b(a)$ and $F_c(a)$ belong to $D_1$, which contradicts the unicity of $b_0$. This finishes the proof that the constraints of $\Sigma$ are true in $I_1$.

Moreover, since $Q^{H_1} = \emptyset$, we have $H_1 \subseteq I_1 \subseteq D_1$. This contradicts the fact that $H_1$ is a repair of $D_1$ with respect to $\Sigma$.

Next we prove the implication from right to left of (i). Suppose that there is no element $Q$-compatible. We define $H_1$ as the following database

$$
Q^{H_1} = \emptyset, \\
R^{H_1} = R^{D_1}, \\
F_b^{H_1} = F_b^{D_1}.
$$

ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
where \( b \in B \) and \( R \) is a relation symbol. We show that \( H_1 \) is a repair. Suppose for contradiction that \( H_1 \) is not a repair. Since \( H_1 \models \Sigma \), there is a repair \( I_2 \) such that \( H_1 \subseteq I_2 \subseteq D_1 \). By definition of \( H_1 \), this can only happen if there is a fact of the form \( Q(\bot_2) \) in \( I_2 \).

We prove that \( \bot_2 \) is \( Q \)-compatible, which is a contradiction. Recall that \( \bot_2 \) is \( Q \)-compatible iff \( Q(\bot_2) \) holds in \( D_1 \) and for all \( a \in A^1 \setminus \{\bot_2\} \), there is a unique \( b \) such that \( F_b(a) \in D_1 \).

By definition, \( Q(\bot_2) \) holds in \( I_2 \) and since \( I_2 \subseteq D_1 \), we have that \( Q(\bot_2) \) belong to \( D_1 \). Next we prove that for all \( a \in A^1 \setminus \{\bot_2\} \), there is a unique \( b \) such that \( F_b(a) \). Take \( a \in A^1 \setminus \{\bot_2\} \). For all \( c,d \in B \) such that \( c \neq d \),

\[
F_c(a) \land F_d(a) \land Q(\bot_2) \rightarrow \bot_2 = a
\]

holds in \( I_2 \). Since \( a \neq \bot_2 \), this implies that there is a unique \( b \) such that \( F_b(a) \in I_2 \). Together with \( F_b^{D_1} = F_b^{I_2} \) (for all \( b \in B \)), this means that there is a unique \( b \) such that \( F_b(a) \in D_1 \). This finishes the proof of \( Q \)-compatibility.

Next we prove (ii). That is, if \( H^1 \) is a repair such that \( Q^{H^1} = \emptyset \), then \( H^1 \not\models q \) iff \( D_1 \not\models q_2 \). Let \( H_1 \) be a repair such that \( Q^{H_1} = \emptyset \). For the implication from right to left, suppose that \( q_2 \) is false in \( D_1 \). Since \( \Sigma \) consists of egds, a repair of \( D_1 \) with respect to \( \Sigma \) is a substructure of \( D_1 \). Intuitively, an egd is not a constraint that can help us “generate” new facts. Since \( H_1 \) is a substructure of \( D_1 \) and \( q_2 \) is false in \( D_1 \), \( q_2 \) is also false in \( H_1 \). Moreover, as \( Q^{H_1} = \emptyset \), \( q_1 \) is also false in \( H_1 \). Hence, \( H_1 \not\models q \).

Next we prove the implication from left to right of (ii), suppose that \( q \) is false in \( H_1 \). Since \( Q^{H_1} = \emptyset \), it follows from Claim 5 that

\[
R^{H_1} = R^{D_1} \text{ and } F_b^{H_1} = F_b^{D_1}, \tag{22}
\]

for all relation symbols \( R \) and for all \( b \in B \). Since \( q \) is false in \( H_1 \), \( q_2 \) is false in \( H_2 \). Observe that the only symbols occurring in \( q_2 \) are the relation symbols \( R \) and \( F_b \). Together with (22) and the fact that \( q_2 \) is false in \( H_1 \), this means that \( q_2 \) is false in \( D_1 \).

We show now that (B3) holds. That is, there is a repair \( H_2 \) such that \( H_2 \not\models q \) and \( Q^{H_2} \neq \emptyset \) iff \( (A^1, L^1) \in cHom(\mathbb{B}) \) and

\[
D_1 \models \exists x(Q(x) \land \neg F_{b_1}(x) \land \cdots \land \neg F_{b_k}(x)). \tag{23}
\]

First we prove the direction from left to right of (B3). Suppose that there is a repair \( H_2 \) such that \( H_2 \not\models q \) and \( Q^{H_2} \neq \emptyset \). Let \( \bot_2 \) be an element in \( Q^{H_2} \). We start by showing that \( (A^1, L^1) \) is \( cHom(\mathbb{B}) \). Since \( Q^{H_2} \neq \emptyset \) and \( q \) is false in \( H_2 \), it follows from Claim 4 that there is a homomorphism \( f^{H_2} : A^{H_2} \rightarrow \mathbb{B} \) such that for every \( a \in A^{H_2} \), \( f_b(a) \) holds in \( H_2 \) and \( b = f^{H_2}(a) \). In order to prove that \( (A^1, L^1) \) is \( cHom(\mathbb{B}) \), it is enough to show that

\[
A^{H_2} = A^{D_1}, \tag{24}
\]

and for all \( a \in A^{H_2} \), \( f^{H_2}(a) \in L^1_a \). \tag{25}

By definitions of \( A^{H_2} \) and \( A^{D_1} \), (24) holds iff \( A^{H_2} = A^{D_1} \). Since \( \Sigma \) is a set of egds and \( H_2 \) is repair of \( D_1 \), \( H_2 \) is a subset of \( D_1 \). Hence, \( A^{H_2} \subseteq A^{D_1} \).

Suppose for contradiction that \( A^{H_2} \) is a proper subset of \( A^{D_1} \). That is, there is a fact \( F_b(a) \) in \( D_1 \) and there is no \( c \in B \) such that \( F_c(a) \in H_2 \). We let \( H_3 \) be the database obtained from \( H_2 \) by adding the tuple \( F_b(a) \). Since there is no \( c \in B \) such that \( F_c(a) \in H_2 \), \( \Sigma \) remains true in \( H_3 \). Moreover,

\[
H_2 \subseteq H_3 \subseteq D_1.
\]

This contradicts the fact that \( H_2 \) is a repair and proves (24).
Why is it hard to obtain a dichotomy for consistent query answering?

Next, we show (25). Let $a$ be an element of $A^{H_2}$. By definition of $f^{H_2}$, if $b = f^{H_2}(a)$, then $F_b(a)$ holds in $H_2$. Since $H_2$ is a subset of $D_1$, this implies that $F_b(a)$ holds in $D_1$. By definition of $L_a^1$, this implies that $b$ belongs to $L_a^1$. This finishes the proof that $(\mathcal{A}^1, L^1) \in cHom(\mathcal{B})$.

Next, we show (23) by proving that

$$D_1 \models Q(\bot_3) \land \neg F_{b_1}(\bot_3) \land \cdots \land \neg F_{b_k}(\bot_3).$$

(26)

Since $\bot_3$ belongs to $Q^{H_2}$, the element $\bot_3$ also belongs to $Q^{D_1}$. Suppose for contradiction that $F_{b}(\bot_3)$ holds in $D_1$ for some $b$ in $B$. Hence, $\bot_3$ belongs to $A^{D_1}$. By (24), $\bot_3$ belongs to $A^{H_2}$. That is, for some $c \in B$, $F_c(\bot_3)$ holds in $H_2$. Since $\bot_3$ belongs to $Q^{H_2}$, this implies that

$$\exists x (Q(x) \land F_c(x))$$

is true in $H_2$. This is not possible, as $q$ is false in $H_2$. This contradiction finishes the proof of (26) and the proof of the implication from left to right of (B3).

Now we prove the implication from right to left of (B3). Suppose that $(\mathcal{A}^1, L^1)$ belongs to $cHom(\mathcal{B})$ and that there is an element $\bot_4$ such that

$$D_1 \models Q(\bot_4) \land \neg F_{b_1}(\bot_4) \land \cdots \land \neg F_{b_k}(\bot_4).$$

(27)

We pick a homomorphism $g_1 : \mathcal{A}^1 \to \mathcal{B}$ such that $g_1(a)$ belongs to $L_a^1$ for all $a \in A^1$. We define $J_1$ as the following subset of $D_1$:

$$Q^{J_1} = \{ \bot_4 \},$$

$$R^{J_1} = R^{D_1},$$

$$F_{b_1}^{J_1} = \{ a \in A^1 : g_1(a) = b \},$$

where $R$ is a relation symbol and $b \in B$. The database $J_1$ is a repair of $D_1$ with respect to $\Sigma$. Next we prove that $J_1 \not\models q$.

First, we prove that $q_1$ is false in $J_1$. Suppose for contradiction that $q_1$ is true in $J_1$. Then there are $b \in B$ and $a \in A^1$ such that

$$J_1 \models Q(a) \land F_b(a).$$

Since $Q^{J_1} = \{ \bot_4 \}$, this implies that $J_1 \models F_b(\bot_4)$. Since $J_1$ is a subset of $D_1$, we also have that $D_1 \models F_b(\bot_4)$, which contradicts (27).

Next we prove that $q_2$ is false in $J_1$. Since $Q^{J_1} \neq \emptyset$ and $q_1$ is false in $J_1$, it follows from Claim 4 that there is a unique map $f^{J_1} : \mathcal{A}^{J_1} \to \mathcal{B}$ such that $F_b(a)$ holds in $J_1$ for all $a \in A^{J_1}$ and where $b = f^{J_1}(a)$. By definition of $J_1$, this implies that $f^{J_1} = g_1$. Moreover, we obtain from Claim 4 that

$$f^{J_1} \text{ is a homomorphism if and only if } J_1 \not\models q_2.$$

Since $g_1$ is a homomorphism and $f^{J_1} = g_1$, this implies that $q_2$ is false in $H_1$.

5. CONCLUSION

We proved that if the dichotomy conjecture holds for consistent query answering with respect to GAV constraints and unions of conjunctive queries, then so does the dichotomy conjecture for CSP. One question left open is whether a similar result could be achieved for other classes of constraints and queries. The case of key constraints and conjunctive queries would be of particular interest, as this is the setting of the original dichotomy conjecture stated by Afrati and Kolaitis [Afrati and Kolaitis 2009].

Another open question is whether we can prove the opposite implication of our main result. That is, is it true that if there is a dichotomy result for CSP, then there is
a dichotomy result for consistent query answering with respect to given classes of constraints and queries?

ACKNOWLEDGMENTS

The author thanks Phokion Kolaitis for suggesting this line of research and Phokion Kolaitis and Balder ten Cate for many useful discussions during the early stages of this work. Many thanks also to Pablo Barceló, Amélie Gheerbrant and anonymous referees for comments on earlier versions of the paper.

The author is funded by Fondecyt grant 3130491 of Conicyt and by Millenium Nucleus Center for Semantic Web Research under grant NC120004. Part of this work was done while doing a postdoctoral stay at the University of California Santa Cruz, supported by NSF Grant IIS-0905276.

REFERENCES


ACM Transactions on Database Systems, Vol. 1, No. 1, Article 1, Publication date: September 2014.
Why is it hard to obtain a dichotomy for consistent query answering?


Received February 2014; revised June 2014; accepted September 2014