Inference-based semantics in Data Exchange

Adrian Onet

Morgan Stanley, Montreal, Canada
adrian.onet@morganstanley.com

Abstract. Data Exchange is an old problem that was firstly studied from a theoretical point of view only in 2003. Since then many approaches were considered when it came to the language describing the relationship between the source and the target schema. These approaches focus on what it makes a target instance a "good" solution for data-exchange. In this paper we propose the inference-based semantics that solves many certain-answer anomalies existing in current data-exchange semantics. We show that in case of mappings represented by source-to-target tgds and safe target egds we may compute in polynomial time a target table (universal representative) able to exactly represent the inference-based semantics. We show that the new semantics agrees with the other semantics when it comes to UCQ queries. Finally we show that one may use the universal representative to compute certain-answers in tractable time for large class of non-UCQ queries even for a subclass of $\text{UCQ}^{\neg}$.

1 Introduction

The data-exchange problem is that of transforming a database existing under a source schema into another database under a different target schema. This database transformation is based on mappings that describe the relationship between the source and the target database. A mapping M can be viewed as a, possibly infinite, set of pairs (I, J), where I is a source instance and J a target instance. In this case, J is called a *data-exchange solution* for I and M. The mapping between the source and the target database is usually specified in some logic formalism. The most widely accepted mapping language is the one based on sets of tuple-generating dependencies (*tgds*) and equality-generating dependencies (*egds*).

It is very common to query the target solutions. This takes us to the problem of querying incomplete databases [2]. The *exact answers semantics* defines the answer to a query over an incomplete database as the set of all answers to the query for each instance from its semantics. This approach is rarely feasible in practice, therefore two approximations are commonly accepted: *certain answer* (answers occurring for all solutions) and *maybe answer semantics* (answers occurring for some solutions).

Based on the interpretation of the mapping language there may be several semantics in data exchange. The first semantics was introduced in [7] and

it is here referred to as the OWA-semantics (open world assumption). In order to improve the certain answer behavior for non-UCQ queries under OWA-semantics, Libkin [15] introduced the first closed-world semantics in data exchange, known as CWA-semantics. This semantics was later on extended by Hernich and Schweikardt [14] to include target *tgds*. In CWA-semantics a solution incorporates only tuples that are "justified" by the source and the dependencies. Later on Hernich [11] introduced the GCWA*-semantics for data exchange. Unfortunately most of these semantics have a rather strange behavior when looking for certain answers over the set of solutions. More recently Arenas et al. [4, **?**] introduced a new semantics for data-exchange based on bidirectional constraints. This new semantics solves most of the query anomalies present in the other semantics but it comes with price of non-tractable data complexity even for the simplest data-exchange problems.

To better understand the type of anomalies we may encounter under these semantics, consider the next simple example. A company has in the source schema a binary relation P representing the relationship between projects and employees. After some reorganization it was realized that each projects financing should be provided by one or more cost-centers, each employee belonging to one of these cost-centers. With this the company creates the target schema with two binary relations PC and CE representing the project to cost-center and cost-center to employee relationship respectively. In the case of the unidirectional semantics the process can be specified by *tgd*

$$\xi_1 : \forall p\, \forall e\, P(p,e) \rightarrow \exists cc\, PC(p,cc) \wedge CE(cc,e).$$

Let source I be $P^I = \{(p_1,e_1),(p_1,e_2),(p_2,e_3)\}$, stating that there are two employees e_1 and e_2 working on project p_1 and only employee e_3 working on project p_2. Consider query: $Q{:=}\forall p\, \forall cc\, \big(PC(p,cc) \wedge CE(cc, \text{“}e_3\text{“}) \rightarrow p = \text{“}p_2\text{“}\big)$ (*Is* e_3 *involved only in project* p_2*?*). Let instance J be $PC^J = \{(p_1,cc_1),(p_2,cc_1)\}$ and $CE^J = \{(cc_1,e_1),(cc_1,e_2),(cc_1,e_3)\}$. Because J is part of all aforementioned unidirectional semantics, the certain answer for the given query will be un-intuitively **false**. Naturally one may expect this answer to be **true**, as there is no indication from the source that employee e_3 is involved in project p_1.

Motivated by certain answer anomalies in the current data-exchange semantics, in this paper we propose a new data-exchange semantics based on logical inference for mappings represented by sets of *s-t tgds* and safe target *egds*. This semantics eliminates most anomalies related to certain-answers and keeps the same certain answers with the other semantics for union of conjunctive queries. We will also show that under this settings for any source instance one may compute a target table representation that may be queries to obtain certain-answers for any FO query. Finally we will review some complexity results regarding the certain-answers under this new semantics and present large classes of FO queries for which certain-answers can be computed in polynomial time. More details, examples and proofs can be found in the full version of this paper.

2 Preliminaries

This section reviews the basic technical preliminaries and definitions. More information on relational database theory can be obtained from [1]. We will consider the complexity classes P, NP and coNP. For the definition of these classes we refer to [17].

A finite mapping f, where $f(a_i) = b_i$, for $i \in \{1, \ldots, n\}$, will be represented as $\{a_1/b_1, a_2/b_2, \ldots, a_n/b_n\}$. When it is clear from the context f, it will be also viewed as the following formula $a_1 = b_1 \wedge a_2 = b_2 \wedge \ldots \wedge a_n = b_n$. For a mapping f and set A, with $f|_A$ will denote the mapping f restricted to values from A. By abusing the notation, a vector $\bar{x} = (x_1, x_2, \ldots, x_n)$ will be often viewed as the set $\{x_1, x_2, \ldots, x_n\}$, thus we may have set operations like $x_i \in \bar{x}$ or $\bar{x} \cap \bar{y}$.

Databases. A *schema* $\mathbf{S}$ is a finite set $\{S_1, S_2, \ldots, S_n\}$ of relational symbols, each symbol S_i having a fixed arity $arity(S_i)$. Let Cons, Nulls and Vars be three countably infinite sets of constants, nulls and variables such that there are no common elements between any two of these sets. Elements from Cons are symbolized by lower case (possibly subscripted) characters from the beginning of the alphabet (e.g. a, b_1). Elements from Vars are represented by lower case (possibly subscripted) characters from the end of the alphabet (e.g. z, x_2). Each element from the countable set Nulls is represented with, possible subscripted, symbol $\perp$ (e.g. $\perp_i$). A *naïve table* T of $\mathbf{S}$ is an interpretation that assigns to each relational symbol S_i a finite set $S_i^T \subset (\mathsf{Cons} \cup \mathsf{Nulls})^{arity(S_i)}$, sometimes we also view S_i as a relation between elements of $\mathsf{Cons} \cup \mathsf{Nulls}$. The set $dom(T)$ means all elements that occur in T, clearly $dom(T) \subseteq \mathsf{Cons} \cup \mathsf{Nulls}$. A naïve table T is called an *instance* if $dom(T) \subset \mathsf{Cons}$. In contrast to general naïve tables, which are identified by capitalized characters from the end of the alphabet (e.g. T, V), instances are represented by capitalized characters from the middle of the alphabet (e.g. I, J). The set of all instances over schema $\mathbf{S}$ is denoted $Inst(\mathbf{S})$. A *valuation* v is a mapping over the set $\mathsf{Cons} \cup \mathsf{Nulls}$ such that $v(a) = a$, for all $a \in \mathsf{Cons}$, and $v(\perp) \in \mathsf{Cons}$, for all $\perp \in \mathsf{Nulls}$. Valuations are extended to tuples and naïve tables as follows. For each tuple $\bar{t} = (t_1, t_2, \ldots, t_n)$, let $v(\bar{t}) := (v(t_1), v(t_2), \ldots, v(t_n))$; and for a naïve table T over schema $\mathbf{S}$, define $v(T)$ as $R^{v(T)} := \{v(\bar{t}) : \bar{t} \in R^T\}$, for all $R \in \mathbf{S}$. The interpretation of a naïve table T is given by $Rep(T) := \{v(T) : v \text{ valuation}\}$.

Schema mappings. A data-exchange *schema mapping* is a triple $M = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are two disjoint schemas named the source and target schema respectively; Σ is a set of formulae expressing the relationship between the source and the target database. Most commonly, Σ is represented by a set of source-to-target tuple-generating dependencies (*s-t tgds*) and target equality-generating dependencies (*egds*). Where a source-to-target tuple-generating dependency is a FO sentence ξ of the form: $\forall \bar{x}\ (\forall \bar{y}\ \alpha(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \beta(\bar{x}, \bar{z}))$, where $\bar{x}$, $\bar{y}$ and $\bar{z}$ are vectors of variables from Vars; $\alpha(\bar{x}, \bar{y})$ (often referred to as the body of the *tgd*) is a conjunction of atoms over the source schema; and $\beta(\bar{x}, \bar{z})$ (often referred to as the head of the *tgd*) is a conjunction of atoms over the target schema. An *equality-generating* dependency is a FO sentence ξ of the form: $\forall \bar{x}\ (\alpha(\bar{x}) \rightarrow x = y)$, where $\alpha(\bar{x})$ is a conjunction of atoms over the target schema and x,y are variables from

the vector $\bar{x}$. A source instance $I \in Inst(\mathbf{S})$ and a target instance $J \in Inst(\mathbf{T})$ are said to satisfy a *s-t tgd* ξ, denoted $(I, J) \models \xi$; if $I \cup J$ is a model of ξ in the model-theoretic sense. Similarly, a target instance J satisfies an *egd* ξ, denoted $J \models \xi$, if J is a model for ξ in the model-theoretic sense. This is extended to a set of *s-t tgds* and *egds* Σ by stipulating that $(I, J) \models \xi$, for all *s-t tgd* $\xi \in \Sigma$ and $J \models \xi$, for all *egd* $\xi \in \Sigma$. A *position* in Σ is a pair (R, i), where R is a relational symbol and $1 \leq i \leq arity(R)$. A position (R, i) is said to be *affected* in Σ if it holds an existentially quantified variable somewhere in Σ. With $\mathsf{aff}(\Sigma)$ is denoted the set of all affected positions in Σ. When the schemata are known or not relevant in the context, we usually interchange the notion of schema mapping and the set of dependencies that defines it.

A data-exchange semantics $\mathcal{O}$ associates for a schema mapping $M = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a source instance I a possible infinite set of target instances $[\![(I, M)]\!]_{\mathcal{O}}$. We refer to each element of $[\![(I, M)]\!]_{\mathcal{O}}$ as a solution for I and M under semantics $\mathcal{O}$.
Queries. CQ, UCQ, $\mathsf{UCQ}^{\neg}$ and $\mathsf{UCQ}^{\neq}$ denote the classes of *conjunctive queries*, *union of conjunctive queries*, *union of conjunctive queries with negation* and *union of conjunctive queries with unequalities*, respectively. For complete definitions of these classes, please refer to [1]. For a given data-exchange semantics $\mathcal{O}$, schema mapping M and source instance I, the certain answers for a given query Q is defined as: $\mathbf{cert}^{\mathcal{O}}(Q, (I, M)) := \bigcap_{J \in [\![(I,M)]\!]_{\mathcal{O}}} Q(J)$.

3 Data-exchange semantics

As mentioned in the introduction, there are many semantics considered in data-exchange. In this section we briefly review the most prominent of these semantics. In Section 3.4 we introduce a new semantics for mappings specified by sets of *s-t tgds* and *egds* that addresses many of certain-answers anomalies occurring under the current semantics. In the final part of this section we will review the semantics based on bidirectional constraints.

3.1 OWA Data Exchange

The OWA-Semantics is the first semantics considered in data exchange [7]. This is, by far, the most studied [7, 3, 8, 6, 5, 10, 13]. Under this semantics, given a data-exchange mapping specified by Σ a set of *tgds* and *egds* and given a source instance I, the OWA-semantics for I under Σ is defined as:

$$[\![(I, \Sigma)]\!]_{\mathsf{owa}} := \{J \in Inst(\mathbf{T}) : I \cup J \models \Sigma\}. \tag{1}$$

3.2 CWA Data Exchange

To outcome the counter-intuitive behavior under the OWA-semantics Libkin [15] introduced the CWA-semantics for mappings specified by a set of *s-t tgds*. Hernich and Schweikardt [14] extended the semantics by adding target *tgds*. Given a set Σ of *s-t tgds* and a source instance I a CWA-solution for I and Σ is defined

as any naïve table T over the target schema that satisfies the following three requirements: 1) each null from $dom(T)$ is justified by some tuples from I and a *s-t tgd* from Σ; 2) each justification for nulls is used only once; and 3) each fact in T is justified by I and Σ. The CWA-semantics for certain-answers is defined as: $[\![(I,\Sigma)]\!]_{\mathsf{cwa}}$:=$\{J \in Rep(T) : T$ is a CWA-solution for I under $\Sigma\}$. Examples of certain-answer anomalies under CWA-semantics can be found in [12].

3.3 GCWA* Data Exchange

The GCWA* semantics was inspired from Minker's [16] GCWA- semantics and nicely adapted by Hernich [12] for data exchange. For a source instance I and Σ a set of *s-t tgds* and *egds* with $Sol_{min}(I,\Sigma)$ we denote all the subset-minimal target instances J such that $I \cup J \models \Sigma$. The CGWA*-semantics is defined as:

$$[\![(I,\Sigma)]\!]_{\mathsf{gcwa}^*}:=\Big\{J : J = \bigcup_{i\leftarrow 1}^{n} J_n \text{ for some } n,\ J_i \in Sol_{min}(I,\Sigma) \text{ and } I \cup J \models \Sigma\Big\}$$

Even if the GCWA* semantics solves all aforementioned anomalies, it introduces new ones exemplified hereafter.

Example 1. Consider source instance with binary relation *DeptC* for departments and names of consultant employees working in that department and ternary relation *DeptFTE* for departments and full-time employees (name and id) from the given department. Suppose the company hires all the consultants as full-time employees, thus the target schema will be the ternary relation *DeptEmp* with the same structure as *DeptFTE*. The exchange mapping is represented as:

$$DeptC(did, name) \rightarrow \exists eid\ DeptEmp(did, name, eid);$$
$$DeptFTE(did, name, eid) \rightarrow DeptEmp(did, name, eid).$$

Consider source instance with consultants "*john*" and "*adam*" part of the "*hr*" department and full-time employee "*adam*" with employee id 1 part of "*hr*". Let target query be: *Is there exactly one employee named* adam *in* hr *department?*. Under the CGWA*-semantics the query will return the counter-intuitive answer **true**, even if based on the source instance and given mapping one would expect the answer to be **false**, because beside full-time employee "*adam*" there maybe a consultant named "*adam*" in the "*hr*" department.

3.4 Inference-based semantics

In order to avoid the certain-answer anomalies presented in this section and in the introduction, we present a new closed-world semantics for mappings specified by a set of *s-t tgds* and *egds*.

Let $\xi : \alpha(\bar{x},\bar{y}) \rightarrow \exists\bar{z}\ \beta(\bar{x},\bar{z})$ be a *s-t tgd* and I a source instance. A set of facts J' is said to be *inferred* from I and ξ with function f denoted with $I \xrightarrow{\xi}_f J'$ if $f(\alpha(\bar{x},\bar{y})) \subseteq I$ and $f(\beta(\bar{x},\bar{z})) = J'$. Tuple $t \in J'$ is said to be *strongly inferred* from I and ξ with function f if $t \in f|_{\bar{x}}(\beta(\bar{x},\bar{z}))$, otherwise we say that t

is *weakly inferred*. Intuitively a tuple t is strongly inferred from I and ξ with f if t does not depend on the assignment given by f to existential variables from ξ. Let Σ be a set of *s-t tgds*, I a source instance and J a target instance. A function κ that assigns for each $\xi \in \Sigma$ a set of functions $\{f_1, f_2, \ldots, f_n\}$ such that $I \xrightarrow{\xi}_{f_i} J_i \subseteq J$ is called an *inference strategy* for I and J with Σ. Note that the function that allocates the empty set for each $\xi \in \Sigma$ is an inference strategy for any I and Σ. Given an inference strategy κ and $\xi \in \Sigma$, a tuple $t \in J$ is said to be: *strongly inferred* with strategy κ for ξ if there exist $f \in \kappa(\xi)$ such that t is strongly inferred from I and σ with f; *not inferred* with strategy κ for ξ if there is no $f \in \kappa(\xi)$ with $I \xrightarrow{\xi}_f J' \ni t$; *weakly inferred* with strategy κ for ξ otherwise. A tuple t is said to be *inferred* with strategy κ for I, Σ and J if there exists $\xi \in \Sigma$ such that t is strongly or weakly inferred with κ for ξ. With this we are now ready to introduce the inferred-based semantics.

Definition 1. *Given a source instance I and Σ a set of* s-t tgds *and* egds *the* inference-based semantics *for I and Σ, denoted with $[\![(I, \Sigma)]\!]_{\mathsf{inf}}$, is the set of all target instances J for which there exists an inference strategy κ such that:*

1. *Every tuple $t \in J$ is inferred with κ for I, Σ and J;*
2. *For every $\xi : \alpha \rightarrow \beta \in \Sigma$ and every function f with $f(\alpha) \subseteq I$ there exists $f' \in \kappa(\xi)$ such that f' is an extension of f;*
3. *For every $\xi : \alpha \rightarrow \beta \in \Sigma$ and $J_{\kappa,\xi} = \bigcup_{I \xrightarrow{\xi}_g J_g, g \in \kappa(\xi)} J_g$, there is no function f with $f(\beta) \in J_{\kappa,\xi}$ and $f(\alpha) \not\subseteq I$, where $f(\beta)$ contains at least one weakly inferred tuple with strategy κ for ξ.*
4. *$(I, J) \models \Sigma$ in the model-theoretic sense.*

Intuitively the first rule from the definition states that all tuples in the target instance needs to be inferred from the source instance and the mapping, this is taking care of the tuples not inferred present under OWA-semantics. This also allows the same nulls to be matched to different constant as long as there exists an inference for each of these. This takes care of the query anomalies present under CWA-semantics. The second condition makes sure that all possible source triggers are fired. With this all tuples that can be inferred will be present in at least in one instance in the semantics, this solves the anomaly presented in Example 1 for GCWA*. The third condition ensures that the assignment of nulls does not contradict with the inference strategy used, thus taking care of the query anomaly from the introduction. The last condition is needed in order to guarantee that the instances from the semantics are models for the *egds*. Need to mention here that by renouncing to Condition 3 from the previous definition we obtain the semantics defined in [9] in the context of exchange recovery. The following theorem reveals the complexities for the *solution existence* (Is $[\![(I, \Sigma)]\!]_{\mathsf{inf}} = \emptyset$? for a fixed set Σ) and *solution check* (Is $J \in [\![(I, \Sigma)]\!]_{\mathsf{inf}} = \emptyset$ for a fixed set Σ) problems.

Theorem 1. *For a fixed set of* s-t tgds *the solution-existence problem can be solved in polynomial time and the solution-check is an* NP*-complete problem under inference-based semantics.*

3.5 Bidirectional constraints

Arenas et al. in [4] considered another approach to the certain answer anomaly problem by changing the language used to express the schema mapping. For this, the authors proposed the language of bidirectional constraints. Where a bidirectional constraints is a FO sentence of the form: $\forall \bar{x}\ \alpha(\bar{x}) \leftrightarrow \beta(\bar{x})$;, where α and β are FO formulae over atoms from the source and target schema respectively with free variables $\bar{x}$. If the language of α is $\mathcal{L}_S$ and of β is $\mathcal{L}_T$, then we are talking about a $\langle \mathcal{L}_S, \mathcal{L}_T \rangle$-dependency. With this, given a source instance I and $\Sigma^{\leftrightarrow}$ a set of $\langle \mathcal{L}_S, \mathcal{L}_T \rangle$-dependencies, the bidirectional semantics is defined as:

$$[\![(I, \Sigma^{\leftrightarrow})]\!]_{\leftrightarrow} := \{J \in Inst(\mathbf{T}) : I \cup J \models \Sigma^{\leftrightarrow}\}. \tag{2}$$

This approach did solve most of the anomalies related to the other semantics. Unfortunately, this is achieved at a high cost, as even the most common data-exchange problems became non-tractable. For example, testing if the semantics is empty is an NP-hard problem [4] even for a set of $\langle \mathsf{CQ}, \mathsf{CQ} \rangle$-dependencies.

Another issue with bidirectional semantics is that there are simple unidirectional mappings for which neither of the presented closed-world semantics are not expressible using bidirectional constraints without changing the target schema, as shown in the example below.

Example 2. Consider source schema with two binary relations $PFTE$, for projects and the full-time employees assigned on the project, and PT, that contains the tasks associated with each project. Let target schema consist of a binary relation PE, for projects and employee assigned to that project, and binary relation TM, for employee and the task they manage. Consider source instance I, with $PFTE^I = \{(hr, adam)\}$ and $PT^I = \{(hr, comp)\}$, stating that full-time employee 'adam' works on the 'hr' project and the 'hr' project consists of one task 'comp'. Consider the following mapping Σ stating that each full-time employee working on a project is also an employee working on the project (ξ_1) and that for each project task there exists an employee working on that project that manages that task (ξ_2).

$$\xi_1 : PFTE(pid, eid) \rightarrow PE(pid, eid)$$
$$\xi_2 : PT(pid, tid) \rightarrow \exists eid\ PE(pid, eid), TM(eid, tid).$$

It is easy to observe that for any set $\Sigma^{\leftrightarrow}$ of bidirectional constraints there is no possibility to differentiate between the tuples from relation PE as being inferred from $PFTE$ or PT source relation. Thus, for $J_1 = \{PE(hr, adam)\}$, $J_2 = J_1 \cup \{TM(adam, comp)\}$ and $J_3 = J_1 \cup \{PE(hr, sal), TM(sal, comp)\}$, we have that either $J_1 \in [\![(I, \Sigma^{\leftrightarrow})]\!]_{\leftrightarrow}$ or $J_2 \notin [\![(I, \Sigma^{\leftrightarrow})]\!]_{\leftrightarrow}$ or $J_3 \notin [\![(I, \Sigma^{\leftrightarrow})]\!]_{\leftrightarrow}$, where "sal" is a consultant not a full-time employee. On the other hand, we have that $J_1 \notin [\![(I, \Sigma)]\!]_*$ and $J_2, J_3 \in [\![(I, \Sigma)]\!]_*$, for any semantics $* \in \{\mathsf{cwa}, \mathsf{gcwa}^*, \mathsf{inf}\}$.

4 Universal Representatives

As mentioned in the introduction, data exchange transforms a database existing under a source schema into another database under the target schema. This

means that for a given semantics it would be preferable to be able to materialize one or more table representations on a target. The materialized table(s) could later be used to obtain answers for different queries over the target database. In this section we will introduce a new type of table capable of representing all solutions for the inference-based semantics. Thus this new table can be used to obtain the certain answers for any FO query.

Definition 2. *Let* Nulls *be partitioned in two infinite sets* $\mathsf{Nulls}^{\mathsf{o}}$ *and* $\mathsf{Nulls}^{\mathsf{c}}$. *A* semi-naïve table *is a naïve table T for which each null is identified as being either from* $\mathsf{Nulls}^{\mathsf{o}}$ *or* $\mathsf{Nulls}^{\mathsf{c}}_n$. *The semi-naïve table T has the following interpretation:*

$$Rep(T) := \{J = v(\bigcup_{i \leftarrow 1} v_i(T)) \text{ : } v \text{ valuation over } \mathsf{Nulls}^{\mathsf{c}}, v_i \text{ valuation over } \mathsf{Nulls}^{\mathsf{o}}\}$$

The nulls from $\mathsf{Nulls}^{\mathsf{o}}$ are called *open* and denoted $\perp^{\mathsf{o}}$ (possibly subscripted). The ones from $\mathsf{Nulls}^{\mathsf{c}}$ are called *closed* and denoted $\perp^{\mathsf{c}}$ (possibly subscripted).

Example 3. Let T be the semi-naïve table with $R^T = \{(a, \perp^{\mathsf{o}}_1, \perp^{\mathsf{c}}_1, \perp^{\mathsf{c}}_2)\}$. We have $I_1, I_2, I_3 \in Rep(T)$, where $R^{I_1} = \{(a, a, b, c)\}$, $R^{I_2} = \{(a, a, b, c), (a, b, b, c)\}$ and $R^{I_3} = \{(a, a, b, a), (a, b, b, a), (a, c, b, a)\}$. For $R^{I_4} = \{(a, a, b, c), (a, a, b, d)\}$, we have that $I_4 \notin Rep(T)$, because closed null $\perp^{\mathsf{c}}_2$ was valued to both c and d.

To a semi-naïve table T we add a global condition φ^*, denoted (T, φ^*), as a conjunction $\delta_1 \wedge \delta_2 \wedge \ldots \wedge \delta_n$ where each conjunct δ_i, $1 \leq i \leq n$, is a disjunction of unequalities over the elements from $dom(T)$. Given v a valuation over $\mathsf{Nulls}^{\mathsf{c}}$ and $v_1, v_2, \ldots, v_n$ valuations over $\mathsf{Nulls}^{\mathsf{o}}$, for some integer n, we say that $(v, \{v_1, v_2, \ldots, v_n\})$ satisfies $x \neq y$, denoted $(v, \{v_1, v_2, \ldots, v_n\}) \models (x \neq y)$, iff:

- $v(x) \neq a$, when $x \in \mathsf{Nulls}^{\mathsf{c}}$ and $y = a \in \mathsf{Cons}$;
- $v_i(x) \neq a$, for all $i \leq n$, when $x \in \mathsf{Nulls}^{\mathsf{o}}$ and $y = a \in \mathsf{Cons}$;
- $v(x) \neq v_i(y)$, for all $i \leq n$, when $x \in \mathsf{Nulls}^{\mathsf{c}}$ and $y \in \mathsf{Nulls}^{\mathsf{o}}$;
- $v(x) \neq v(y)$, when $x, y \in \mathsf{Nulls}^{\mathsf{c}}$; and
- $v_i(x) \neq v_j(y)$, for all $i, j \leq n$, when $x, y \in \mathsf{Nulls}^{\mathsf{o}}$.

The previous notion is naturally extended to a disjunction of unequalities and to the conjunctive formula φ^* where each conjunct represents a disjunction of unequalities. This is denoted $(v, \{v_1, v_2, \ldots, v_n\}) \models \varphi^*$. With this we can define:

$$Rep(T, \varphi^*) := \{J = v(\bigcup_{i \leftarrow 1}^{n} v_i(T)) : \ n \text{ an integer, } v \text{ valuation over } \mathsf{Nulls}^{\mathsf{c}}, \\ v_i \text{ valuation over } \mathsf{Nulls}^{\mathsf{o}} \text{ and } (v, \{v_1, \ldots, v_n\}) \models \varphi^*\}$$

Example 4. Consider (T, φ^*), where T is the same as in Example 3 and global condition $\varphi^* := (\perp^{\mathsf{o}}_1 \neq a \vee \perp^{\mathsf{c}}_2 \neq \perp^{\mathsf{o}}_1)$. It can be verified that $I_1, I_2 \in Rep(T, \varphi^*)$ and $I_3 \notin Rep(T, \varphi^*)$ because the tuple $R(a, a, b, a)$ in I_3 was obtained from valuations $v = \{\perp^{\mathsf{c}}_1/b, \perp^{\mathsf{c}}_2/a\}$ and $v_1 = \{\perp^{\mathsf{o}}_1/a\}$ and $(v, \{v_1\}) \not\models \varphi^*$.

For the next result, let's define the notion of *safe* egds. For a set Σ of *s-t tgds* and *egds* we say that it contains *safe* egds if each variable that occurs more than once in the body of an *egd* it occurs only in positions different than $\mathsf{aff}(\Sigma)$.

The following result shows that for any set Σ of *s-t tgds* and safe *egds* there exists an exact representation for the inference-based semantics.

Theorem 2. *Let Σ be a set of* s-t tgds *and safe* egds. *Then one may compute in polynomial time (for a fixed Σ) (T, φ^*) such that $[\![(I, \Sigma)]\!]_{\mathsf{inf}} = Rep(T, \varphi^*)$.*

The pair (T, φ^*) from the previous theorem is called *universal representative* for I and Σ. In Theorem 2 the universal representative is computed using a 3-step chase algorithm (for a detailed description please check the full version of the paper).

Example 5. Consider $\Sigma = \{\xi_1, \xi_2, \xi_3\}$, where:

$$\begin{aligned} \xi_1 : &\quad S(x,y) \rightarrow K(x,z), V(z,y); \\ \xi_2 : &\quad R(x) \rightarrow U(x,y); \\ \xi_3 : &\quad U(x,y), K(x,z) \rightarrow y = z. \end{aligned}$$

Let instance I be $S^I = \{(a,b),(c,d)\}$ and $R^I = \{(a)\}$. In this case a universal representative for I and Σ is the pair (T, φ^*), where $K^T = \{(a, \bot_1^{\mathsf{c}}), (c, \bot_1^{\mathsf{o}})\}$, $V^T = \{(\bot_1^{\mathsf{c}}, b), (\bot_1^{\mathsf{o}}, d)\}$, $U^T = \{(a, \bot_1^{\mathsf{c}})\}$ and $\varphi^*{:=}(\bot_1^{\mathsf{c}} \neq \bot_1^{\mathsf{o}})$.

5 Query Answering

If in the previous section we showed how we can compute universal representatives for inference-based semantics. in this section we will focus on when and how these representatives can be used to compute certain answers for different query classes and check the complexities of such evaluations. Let us first start by defining the certain-answer evaluation problem.

PROBLEM	$\text{EVAL}_{\mathsf{inf}}(\Sigma, Q)$
INPUT:	$I \in Inst(\mathbf{S})$ and $\bar{t} \in (dom(I))^{arity(\bar{t})}$.
QUESTION:	Is $\bar{t} \in \mathbf{cert}^{\mathsf{inf}}(Q, (I, \Sigma))$?

The following theorem ensures that for any mapping Σ of *s-t tgds* and safe *egds* the OWA and inference-based semantics agrees on UCQ certain-answers for any source instance.

Theorem 3. *Let Σ be a set of* s-t tgds *and safe* egds *. Then for any source instance I and $q \in \mathsf{UCQ}$ we have $\mathbf{cert}^{\mathsf{owa}}(Q, (I, \Sigma)) = \mathbf{cert}^{\mathsf{inf}}(Q, (I, \Sigma))$ and one may use the universal representative for I and Σ to compute $\mathbf{cert}^{\mathsf{inf}}(Q, (I, \Sigma))$ in polynomial time (for a fixed Σ).*

The following negative result shows that not all queries are tractable under inference-based semantics.

Theorem 4. *There exists a set Σ of* s-t tgds *and safe* egds *and there exists a query $Q \in \mathsf{CQ}^{\neg}$ such that the problem* $\text{EVAL}_{\mathsf{inf}}(\Sigma, Q)$ *is* coNP*-complete.*

From this it follows that for tractable query evaluation under inference-based semantics we need to restrict either the set Σ or the query class used or both. In the last part of this section we will present such restrictions that ensure tractability for certain answers evaluation under inference-based semantics.

Proposition 1. *Let Σ be a set of full* s-t tgds *and safe* egds. *Then for any* FO *query Q the* $\text{EVAL}_{\mathsf{inf}}(\Sigma,Q)$ *is tractable and one may use the universal representative to compute it.*

With $\mathsf{UCQ}^{\neq,\mathsf{n}}$ is denoted the set of $\mathsf{UCQ}^{\neq}$ queries with at most n unequalities per disjunct.

Theorem 5. *Let Σ be a set of* s-t tgds *and safe* egds. *Then for any* $\mathsf{UCQ}^{\neq,1}$ *query Q the* $\text{EVAL}_{\mathsf{inf}}(\Sigma,Q)$ *problem is tractable and one may use only the universal representative to evaluate the query. And the* $\text{EVAL}_{\mathsf{inf}}(\Sigma,Q)$ *for $q \in \mathsf{UCQ}^{\neq,2}$ is* coNP*-complete.*

In [12] Hernich showed that if the mapping is given by a restricted set of *s-t tgds* (packed *s-t tgds*), then the certain answers evaluation problem may be answered in polynomial time for universal queries under the GCWA*-semantics. Where a universal query is one of the form $Q(\bar{x}):=\forall \bar{y}\ \beta(\bar{x},\bar{y})$, with β a quantifier-free FO formula over the target schema. In our next result we show that similar polynomial time can be achieved under the inference-based semantics even without any restriction on the *s-t tgds* and also by adding safe target *egds*.

Theorem 6. *Let I be a source instance and Σ a set of* s-t tgds *and safe* egds. *Then for any universal query Q,* $\text{EVAL}_{\mathsf{inf}}(\Sigma,Q)$ *can be solved in* PTIME.

Next we will present a subclass of $\mathsf{CQ}^{\neg}$ that has tractable query evaluation properties for a restricted class of *s-t tgds* and safe *egds*. Let $\mathsf{CQ}^{\neg,1}$ denote the subclass of $\mathsf{CQ}^{\neg}$ such that each query of this class has exactly one positive atom.

Theorem 7. *Let Σ be a set of* s-t tgds *and safe* egds*, such that for all* s-t tgds *each existentally quantified variable occurs in only one atom and each* egd *does not equate two variables both occurring in affected positions. Then for any* $\mathsf{CQ}^{\neg,1}$ *query the* $\text{EVAL}_{\mathsf{inf}}(\Sigma, Q)$ *problem is polynomial and can be decided using a universal representative.*

Intuitively, the restrictions on the mapping language from the previous theorem ensure that the universal representative does not have any global condition and it contains only open nulls.

6 Conclusions

In this paper we introduced the inference-based semantics for data exchange. We showed that the inference-based semantics solves most of the certain-answers anomalies existing in the existing semantics and that one may compute a universal representative that exactly represents the semantics. For the certain answer semantics it remained an open problem if one can evaluate certain-answers for any $\mathsf{CQ}^{\neg,1}$ queries in polynomial time for any Σ and not only for the restricted class of dependencies presented here. As further work we intend to increase the language for this semantics to included target *tgds* too.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. Serge Abiteboul, Paris C. Kanellakis, and Gösta Grahne. On the representation and querying of sets of possible worlds. In *SIGMOD Conference*, pages 34–48, 1987.
3. Marcelo Arenas, Pablo Barceló, Ronald Fagin, and Leonid Libkin. Locally consistent transformations and query answering in data exchange. In *Proceedings of the Twenty-third ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 14-16, 2004, Paris, France*, pages 229–240, 2004.
4. Marcelo Arenas, Gabriel Diéguez, and Jorge Pérez. Expressiveness and complexity of bidirectional constraints for data exchange. In *Proceedings of the 8th Alberto Mendelzon Workshop on Foundations of Data Management, Cartagena de Indias, Colombia, June 4-6, 2014.*, 2014.
5. Marcelo Arenas, Jorge Pérez, and Juan L. Reutter. Data exchange beyond complete data. In *PODS*, pages 83–94, 2011.
6. Alin Deutsch, Alan Nash, and Jeffrey B. Remmel. The chase revisited. In *PODS*, pages 149–158, 2008.
7. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
8. Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.
9. Gösta Grahne, Ali Moallemi, and Adrian Onet. Recovering exchanged data. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems, PODS 2015, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 105–116, 2015.
10. Gösta Grahne and Adrian Onet. Representation systems for data exchange. In *ICDT*, pages 208–221, 2012.
11. André Hernich. *Foundations of query answering in relational data exchange*. PhD thesis, 2010.
12. André Hernich. Answering non-monotonic queries in relational data exchange. *Logical Methods in Computer Science*, 7(3), 2011.
13. André Hernich. Computing universal models under guarded tgds. In *ICDT*, pages 222–235, 2012.
14. André Hernich and Nicole Schweikardt. Cwa-solutions for data exchange settings with target dependencies. In *PODS*, pages 113–122, 2007.
15. Leonid Libkin. Data exchange and incomplete information. In *PODS*, pages 60–69, 2006.
16. Jack Minker. On indefinite databases and the closed world assumption. In *CADE*, pages 292–308, 1982.
17. Christos H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.