Data Exchange and Schema Mappings in Open and Closed Worlds

Leonid Libkin[a], Cristina Sirangelo[b]

[a] *University of Edinburgh*
[b] *LSV, ENS–Cachan, CNRS and INRIA*

Abstract

In the study of data exchange one usually assumes an open-world semantics, making it possible to extend instances of target schemas. An alternative closed-world semantics only moves 'as much data as needed' from the source to the target to satisfy constraints of a schema mapping. It avoids some of the problems exhibited by the open-world semantics, but limits the expressivity of schema mappings. Here we propose a mixed approach: one can designate different attributes of target schemas as open or closed, to combine the additional expressivity of the open-world semantics with the better behavior of query answering in closed worlds.

We define such schema mappings, and show that they cover a large space of data exchange solutions with two extremes being the known open and closed-world semantics. We investigate the problems of query answering and schema mapping composition, and prove two trichotomy theorems, classifying their complexity based on the number of open attributes. We find conditions under which schema mappings compose, extending known results to a wide range of closed-world mappings. We also provide results for restricted classes of queries and mappings guaranteeing lower complexity.

Key words: Data exchange, schema mappings, closed world assumption, open world assumption, incomplete information

1. Introduction

Data exchange is the problem of finding an instance of a target schema, given an instance of a source schema and a specification of a mapping between the source and the target schemas, and answering queries over target instances in a way that is consistent with the source information. Specifications between the source and the target are given in the form of a schema mapping. The study of both data exchange and schema mappings (in particular, operations on schema mappings) has been actively pursued recently (see, e.g., recent SIGMOD and PODS keynotes [18, 7]). Existing implementations [26, 28] have been incorporated into major database products.

Theoretical foundations of data exchange were first developed in [11, 12]. For a source instance S and a schema mapping M, a target instance T is a *solution* for S if S and T together satisfy the conditions of M. Target instances often contain incomplete information as mappings are rarely fully specified: for example, it is common for target schemas to have attributes that are not present in the source. To account for missing information, target instances are populated with *nulls*.

Papers [11, 12] also developed query answering techniques for data exchange that work very well for conjunctive (and positive relational algebra) queries, but have been shown to exhibit strange behavior for queries involving negation. This happens even with very simple mappings, for example, mappings specifying that each tuple from the source be copied into the target [11, 3]. There are several reasons for such unnatural behavior, stemming from handling of incomplete information. We shall outline them below.

A source instance S may have many different solutions under a mapping M. Thus, the standard approach for answering a query Q over the target schema is to find *certain answers* $\mathsf{certain}_M(Q, S)$.

These were defined in [11, 18] as the intersection of $Q(T)$'s for all solutions T:

$$\mathsf{certain}_M(Q, S) = \bigcap\{Q(T) \mid T \text{ is a solution}\}.$$

Normally, only one target instance T_0 is materialized (typically a *canonical solution* [11] or its *core* [12]). Hence, the goal of query answering in data exchange is to compute certain answers, by posing a query against that materialized instance. That is, one needs to evaluate some query Q' so that $\mathsf{certain}_M(Q, S) = Q'(T_0)$.

However, solutions T's (including the materialized solution T_0) are instances with nulls, and there is no well-defined concept of $Q(T)$ for databases with nulls [17, 22, 15]. Most commonly, to evaluate Q over an instance T with nulls, one tries to find the set $\Box Q(T)$ of answers independent of the interpretation of nulls. These are often called *certain answers* in the incomplete information literature, but they should not be confused with data exchange certain answers ($\mathsf{certain}_M(Q, S)$). In the rest of the paper, when we will talk about certain answers, it will be always clear from the context whether we refer to $\Box Q$ or to data exchange certain answers.

There are several known evaluation mechanisms for computing $\Box Q(T)$. The one used in [11, 12] is the *naive evaluation* $Q_{naive}(T)$: it treats nulls as atomic values (i.e., two nulls are equal iff they are syntactically the same) and only keeps null-free tuples in the output.

For conjunctive queries, and their unions, [11, 12] proved that $\mathsf{certain}_M(Q, S)$, defined as the intersection of $Q_{naive}(T)$ over all solutions T, can be computed as $Q_{naive}(T_0)$, where T_0 is the canonical solution. This follows from

$$\mathsf{certain}_M(Q, S) = \Box Q(T_0) \tag{1}$$
$$\Box Q(T_0) = Q_{naive}(T_0) \tag{2}$$

for such queries. However, for full relational algebra, even if (1) were to remain true, relying on (2) for finding the result of a query is impossible, as the naive evaluation no longer produces the set of certain answers [17]. Moreover, [3] showed that there are relational calculus queries Q for which $\mathsf{certain}_M(Q, S)$ cannot be expressed as $Q'_{naive}(T_0)$, where Q' is a relational calculus (or even an aggregate) query.

Furthermore, the notion of solutions is not unique (see, e.g., [11, 12, 21]) and neither is the notion of $\Box Q$ in general, as both depend on assumptions about tuples in solutions and interpretation of nulls. Papers [12, 11, 13] make the *Open World Assumption*, or *OWA* [29]. Under this assumption, tuples can be freely added to solutions. For example, if M is a mapping stating that tuples from the source S must be copied to the target T, then, under the OWA, every T that extends S is a solution. Hence, if $S = \emptyset$, then every T is a solution, and computing certain answers is as hard as finite validity (which is undecidable for relational calculus) even in such simple settings.

There is an alternative notion of solutions, proposed in [21, 16]. It is based on the *Closed World Assumption*, or *CWA* [29]. Such solutions T have "just as much as needed" to satisfy the conditions imposed by M. For example, if M states that every tuple in S must be in T, the only CWA-solution for S would be a copy of S, since instances are no longer open to adding new tuples. This approach guarantees $\mathsf{certain}_M(Q, S) = \Box Q(T_0)$ for the canonical solution T_0, and eliminates some of the anomalies that have been shown to arise under the OWA approach [3]. On the other hand, under the CWA queries may produce counterintuitive answers too, this time because of the "uniqueness of value" constraints imposed by the CWA. For example, consider a mapping stating that for each tuple *(paper#,title)* in a source S there is a tuple *(paper#,author)* in the target T. That is, we keep paper number, drop the author, and assign a null value to the *author* attribute. Let *paper#* be a key for S. Then the certain answer to a query asking whether every paper has exactly one author is *true.* This is because of the minimalistic CWA: it will create just one *(paper#,author)* tuple, which is what is needed to satisfy the mapping constraints, and will stop at that.

Fully open or fully closed mappings, being two extreme cases, are bound to have their shortcomings. Thus, *our goal* is to study mappings that are not rigidly controlled by the OWA, as in [11, 13], or by the CWA, as in [21, 16]. We adapt an old idea of [14], and permit nulls – or, more generally, attributes in targets – to be *open* or *closed*. Open attributes can be instantiated by many values, but for closed, only one value is permitted. In our example, we would declare *paper#* as closed, indicating that only papers from the source are moved to the target, and *author* as *open*, allowing instances with multiple authors of a given paper. Then the certain answer to the "one-author" query is *false*, as expected. We now further illustrate this idea by an example.

Example Consider a source schema σ with binary relations *Papers(paper#, title)* and *Assignments(paper#, reviewer)*. Each instance of σ represents the list of papers submitted to a given conference and the assignments of papers to reviewers. The target schema τ consists of two binary relations *Reviews(paper#, review)* and *Submissions(paper#, author)*. The mapping between the source and the target is provided by a set of rules below:

$$\begin{array}{lll}
\mathit{Submissions}(x^{cl}, z^{op}) & :- & \mathit{Papers}(x, y) \\
\mathit{Reviews}(x^{cl}, z^{cl}) & :- & \mathit{Assignments}(x, y) \\
\mathit{Reviews}(x^{cl}, z^{op}) & :- & \mathit{Papers}(x, y) \wedge \\
 & & \neg \exists r \, \mathit{Assignments}(x, r)
\end{array}$$

We use the syntax that will be introduced later; essentially, we formulate mappings as in [11, 13] (using rule-based notation as in [21]), with extra annotations *op* or *cl* (for open and closed) of variables in the target atoms. Intuitively, the first rule says that the target instance contains exactly the submitted papers from the source (enforced by the closed annotation of the attribute *paper#*). The *author* attribute is populated with nulls, and its open annotation models the one-to-many relationship between papers and their authors.

The second rule says that for each assigned paper and each of its reviewers, exactly one review is associated to the paper in the target. Completely closed annotation here prevents the target from having reviews of assigned papers without a corresponding reviewer in the source. The third rule deals with papers that have not been assigned, according to the source. In this case, the attribute *review* of *Reviews* is annotated as open, to allow several reviews to be generated for the same paper.

We remark that atoms of the same relation can be annotated differently in different rules. Indeed, the annotation of an atom of a given target relation R in a rule describes the way *the particular rule* allows data to be moved from the source to relation R in the target, and this may vary from a rule to a rule. □

Open/closed annotations could be an easy addition to systems that handle schema mappings [26, 28, 8] as they essentially state whether we have a one-to-one or a one-to-many relationship for a correspondence between attributes in the source and the target, and only require one-bit annotations for target attributes.

Contributions Our first goal is to study data exchange based on mappings that allow annotating target attributes as open or closed. We define their semantics via different interpretations of null values, and show the following:

- The solutions of [11, 21] are the two extreme cases: when all attributes are open (solutions of [11]), and when all are closed (solutions of [21]).
- For conjunctive (and positive relational algebra) queries, certain answers can be computed by the tractable naive evaluation, regardless of annotations.
- Under the appropriate notion of certain answers with mixed open and closed nulls, we always have (1) – that is, $\mathsf{certain}_M(Q, S) = \Box Q(T_0)$, where T_0 is the canonical solution. Thus, query answering in data exchange is reduced to query answering over a particular polynomial-time computable instance with nulls.

- For full relational algebra, computing certain answers depends on annotations. We prove a *trichotomy* result, classifying the complexity of certain answers in terms of the maximum number k of open attributes per atom in a rule of the mapping M: it is coNP-complete if $k = 0$ (under the CWA), it is coNEXPTIME-complete if $k = 1$, and undecidable for $k > 1$. Most of the work goes into the coNEXPTIME result: undecidability for $k > 1$ is an easy consequence of Trakhtenbrot's theorem, as already noticed in [1, 11], and coNP-completeness under CWA was shown in [21] by an adaptation of results in [2]. We also show how lower complexity can be achieved by putting additional restrictions on queries.

We then study schema mappings themselves. This subject too has witnessed a lot of activity recently (see [7]). A central topic is the study of operations on mappings, with perhaps the most common one being *composition*: for mappings $M_{\sigma\tau}$ and $M_{\tau\omega}$ between schemas σ and τ, and τ and ω, resp., how do we obtain a mapping $M_{\sigma\omega}$ that transforms σ-databases into ω-databases by applying $M_{\sigma\tau}$ first, followed by $M_{\tau\omega}$?

Composition is crucial for understanding schema evolution, and it has been extensively studied (see, e.g., [6, 13, 27, 23]). The idea of the standard approach of [13] is to define composition semantically, and then capture the same notion syntactically. Semantically, a schema mapping M is a binary relation with pairs (S, T) such that T is a solution for S. Then the composition of mappings is the composition of binary relations. The definition of [13] does not permit instances with nulls, and interprets both mappings and solutions under the OWA. Then, under the OWA, [13] showed how to capture the semantic notion of composition syntactically with Skolemized constraints. But it is then natural to ask what happens if a different interpretation, e.g. closed-world, is used.

As our second contribution, we study composition of schema mappings that mix open and closed attributes. The notion of [13] is obtained when all attributes are interpreted under the OWA. Our main results are:

- We classify the complexity of composition (i.e., recognizing pairs of instances that belong to the composition of two mappings) by the maximum number k of open attributes in rules of $M_{\sigma\tau}$, proving another trichotomy: NP-completeness for $k = 0$; NEXPTIME-completeness for $k = 1$; and undecidability for $k > 1$.
- If only conjunctive queries are used in mappings (as in [11, 12, 13]), then under both CWA and OWA the composition problem is NP-complete.
- We show that the Skolemized constraints of [13] are closed under composition not only under the OWA but also under the CWA, and look at other conditions that make composition work for mixed open/closed mappings.

Organization In Section 2 we review schema mappings, data exchange solutions, and the basics of incomplete information. Section 3 introduces mappings that combine open and closed-world semantics. Complexity of query answering under such mappings is studied in Section 4. In Section 5 we study the complexity and syntactic characterizations of mapping composition. Concluding remarks are in Section 6.

2. Preliminaries

We now review the notions of schema mappings, data exchange solutions (under different assumptions), and incomplete information. Throughout this paper, we deal with relational settings. If T is an instance of some relational schema τ, then we denote by D_T the active domain of T. Moreover for each relation symbol R in τ, R^T denotes the value of R in T, and all operators on databases instances are intended relation-wise.

Schema mappings and data exchange

Let σ and τ be two relational database schemas; σ is thought of as a *source* schema, and τ as a *target* schema. A mapping M between schemas σ and τ is a condition that states how instances of σ and τ are related [7, 18, 19]. In data exchange, mappings are typically specified by sets of *source-to-target dependencies (STDs)* of the form

$$\psi_\tau(\bar{x}, \bar{z}) \ :\!- \ \varphi_\sigma(\bar{x}, \bar{y}),$$

where φ_σ is a first-order (FO) formula over vocabulary σ, and ψ_τ is a conjunction of atomic τ-formulae [11, 18]. A *mapping* for us is thus a triple (σ, τ, Σ), where Σ is a set of STDs. If S is a source instance, then a target τ-instance T is called a *solution* for S under Σ if $(S, T) \models \Sigma$. More precisely, for every $\psi_\tau(\bar{x}, \bar{z}) :\!- \varphi_\sigma(\bar{x}, \bar{y})$ in Σ, we have $(S, T) \models \forall \bar{x} \forall \bar{y} \big(\varphi_\sigma(\bar{x}, \bar{y}) \rightarrow \exists \bar{z} \psi_\tau(\bar{x}, \bar{z})\big)$. That is, for every pair of tuples $\bar{a}, \bar{b}$ such that $\varphi(\bar{a}, \bar{b})$ holds in S, there is a tuple $\bar{c}$ such that $\psi(\bar{a}, \bar{c})$ holds in T.

Target instances can be populated by two different kinds of elements: *constants* and *nulls*. Constants are elements that come from the source, and nulls are new elements created in targets. We assume two countably infinite disjoint domains Const and Null; elements of Const are denoted by lowercase letters, and elements of Null by $\bot$ with sub/superscripts. Source instances are interpreted as instances over Const, and targets as instances over $\mathsf{Const} \cup \mathsf{Null}$. We assume that we can distinguish nulls from constants (e.g., by a unary predicate testing for nulls, like `IS NULL` in SQL).

One particular solution plays a special role in data exchange: the *canonical* (universal) *solution* $\text{CSol}^\Sigma(S)$, for a mapping (σ, τ, Σ) and a source S [11]. As in [3, 21], it is computed as follows: for each STD $\psi(\bar{x}, \bar{z}) :\!- \varphi(\bar{x}, \bar{y})$ in Σ and for each pair of tuples $\bar{a}, \bar{b}$ such that $\varphi(\bar{a}, \bar{b})$ holds in S, create a fresh tuple of distinct nulls $\bar{\bot} = \bar{\bot}_{(\varphi, \psi, \bar{a}, \bar{b})}$ (so that $|\bar{\bot}| = |\bar{z}|$) and put tuples in the target so that $\psi(\bar{a}, \bar{\bot})$, which is a conjunction of atoms, holds. If the mapping is understood from the context, we write just $\text{CSol}(S)$. The schemas σ and τ will always be clear from the context.

For example, if $\sigma = \{E\}, \tau = \{R\}$, where E and R are binary, and Σ contains $R(x, z) :\!- E(x, y)$, then for $E = \{(a, c_1), (a, c_2), (b, c_3)\}$, the canonical solution has tuples $\{(a, \bot_1), (a, \bot_2), (b, \bot_3)\}$ in R.

Databases with incomplete information

We briefly review some standard definitions [15, 17]. A database instance with incomplete information is an instance whose domain is a subset of $\mathsf{Const} \cup \mathsf{Null}$. Nulls are treated as existing but unknown values. A *valuation* is a partial map $v : \mathsf{Null} \rightarrow \mathsf{Const}$. Given an instance T with incomplete information, and a valuation v defined on all of its nulls, $v(T)$ stands for the instance over Const in which every null $\bot$ in T is replaced by $v(\bot)$. The semantics of T, denoted by $\mathit{Rep}(T)$ [17], consists of all such instances:

$$\mathit{Rep}(T) = \{v(T) \ \mid \ v \text{ is a valuation}\}.$$

Evaluation of queries Q on databases with nulls normally means finding *certain answers* $\Box Q(T) = \bigcap \{Q(R) \mid R \in \mathit{Rep}(T)\}$, i.e. tuples that belong to $Q(R)$ for all possible R in $\mathit{Rep}(T)$.

If Q is a positive relational algebra query, then $\Box Q(T)$ is obtained by the naive evaluation of Q on T (i.e. treating nulls as values) and then discarding tuples containing nulls [17]. For full relational algebra queries one needs a rather complicated mechanism of conditional tables [17] to represent certain answers.

Data exchange under CWA

The definitions of solutions and query answering under the CWA were given in [21]. The main idea is not to open the target to arbitrary new tuples, and instead put there just what is needed to satisfy the STDs. Solutions under the CWA (called CWA-solutions in [21]) must satisfy three criteria: (a) the presence of each null must be justified by the source instance and the STDs; (b) a justification for a null should not generate multiple nulls; and (c) facts true in the target instance must be justified by the source instance and the STDs.

These were formalized in [21]. Before showing how (a) (b) and (c) were formalized, we recall a result from [21] which characterized CWA-solutions as the homomorphic images of $\text{CS}\textsc{ol}(S)$ that have a homomorphism back into $\text{CS}\textsc{ol}(S)$.

We now recall how (a), (b), (c) are formalized. Let (σ, τ, Σ) be a mapping, where Σ is a set of STDs $\{\psi_i(\bar{x}_i, \bar{z}_i) \ \text{:-}\ \varphi_i(\bar{x}_i, \bar{y}_i) \mid 1 \le i \le m\}$, and let S be a source instance. A *justification* for a null consists of an STD ψ_i :- φ_i, a tuple $(\bar{a}, \bar{b})$ so that $\varphi_i(\bar{a}, \bar{b})$ holds, and a variable among the $\bar{z}$'s. Note that justifications generate nulls in the canonical solution $\text{CS}\textsc{ol}(S)$.

Each null in a target T must have a justification for it, but the same justification should not justify different nulls. This means that there is a mapping h from justifications onto the set of nulls of T, i.e. a *homomorphism* $h : \text{CS}\textsc{ol}(S) \to T$ that maps nulls of $\text{CS}\textsc{ol}(S)$ onto the nulls of T. Such homomorphic images of $\text{CS}\textsc{ol}(S)$ were called *CWA-presolutions*. In our previous example of an STD $R(x, z)$:- $E(x, y)$ and a source $E = \{(a, c_1), (a, c_2), (b, c_3)\}$, the canonical solution has nulls $\perp_1, \perp_2, \perp_3$ given by justifications: $((a, c_1), z)$, $((a, c_2), z)$, and $((b, c_3), z)$. If we have a homomorphism $h(\perp_1) = h(\perp_2) = \perp$ and $h(\perp_3) = \perp'$, we obtain a CWA-presolution $\{(a, \perp), (b, \perp')\}$.

Requirement (c) closes instances to unjustified facts, i.e., it prohibits inventing facts based on equating nulls unless they are implied by the source and the STDs. In our example, a homomorphism h' such that $h'(\perp_1) = h'(\perp_3) = \perp$ gives us tuples $(a, \perp), (b, \perp)$ in the presolution. This says that a and b are connected to *the same* element, which is not implied by S and the STDs, and hence should not be allowed under the CWA, as we close the instance to unjustified tuples and facts.

Formally, a *fact* is a formula $f(\bar{a}) = \exists \bar{z}\ \gamma(\bar{a}, \bar{z})$, where $\bar{a}$ is over Const, and γ is a conjunction of τ-atoms; it is satisfied in a target instance T if there is a tuple of nulls $\bar{\perp}$ such that $\gamma(\bar{a}, \bar{\perp})$ is true. Then *CWA-solutions* are defined as CWA-presolutions T so that every fact true in T is also true in $\text{CS}\textsc{ol}(S)$. The presolution $\{(a, \perp), (b, \perp')\}$ is a CWA-solution.

The characterization of CWA-solutions leads to algorithms for finding certain answers, i.e. sets of tuples that belong to $Q(R)$ for every CWA-solution T for S and every $R \in \mathit{Rep}(T)$. Namely, they can be computed as $\Box Q(\text{CS}\textsc{ol}(S))$ [21]. If Q is a union of conjunctive queries (and therefore $\Box Q$ can be computed by the naive evaluation) this coincides with the semantics used in [11]. As we move beyond positive queries, the CWA semantics behaves nicer than the OWA semantics. For example, even in *copying* mappings, with all STDs of the form $R'(\bar{x})$:- $R(\bar{x})$, under the semantics of [11] there are FO-queries that cannot be answered by evaluating an FO-query over the canonical, or other, solutions [3]. Under the CWA, certain answers coincide with $Q(\text{CS}\textsc{ol}(S))$ in such mappings.

3. Mixing OWA and CWA: mappings and solutions

We define mappings that need not follow the all-OWA or the all-CWA policy: in them, attributes of target atoms of STDs can be *annotated* as open or closed. This results in target instances in which different elements have different semantics, so we define an appropriate semantics Rep_A for them.

Annotated mappings

We shall allow each variable in the left-hand side ψ of an STD to be annotated with an element of the set $\{\mathit{op}, \mathit{cl}\}$, referring to them as *open* or *closed* variables, respectively. So formally an annotated STD is a usual STD

$$\psi(x_1, \ldots, x_n,\ z_1, \ldots, z_k) \ \text{:-}\ \varphi(x_1, \ldots, x_n,\ y_1, \ldots, y_m),$$

together with an annotation mapping α that assigns each occurrence of a variable in ψ either *op* or *cl*. An *annotated mapping* consists of source and target schemas σ and τ, and a set of annotated STDs. We put annotation as a superscript, writing x^{op} or x^{cl} when $\alpha(x) = \mathit{op}$ or $\alpha(x) = \mathit{cl}$, resp.

Closed annotations specify one-to-one relationships, so closed nulls behave just as nulls in CWA-solutions. Open annotations specify one-to-many relationships and exhibit the behavior of solutions of [11]. In the earlier example, according to the STD $\mathit{Submissions}(x^{cl}, z^{op})$:– $\mathit{Papers}(x, y)$, only papers from the source are moved to the target in the exchange of data, but the *paper-author* relationship is not one-to-one, and hence multiple values are allowed in the second attribute.

Annotation in instances

Solutions under annotated mappings will be annotated instances, which we now define. A finite relation over attributes $A_1, \ldots, A_n$ with domain D is a finite set of tuples, and each tuple is a mapping $t : \{A_1, \ldots, A_n\} \to D$. An *annotated tuple* is a pair (t, α), where t is a tuple and α is a mapping $\{A_1, \ldots, A_n\} \to \{op, cl\}$. An annotated relation is a finite set of annotated tuples, and an annotated instance is a set of annotated relations. Again we use superscripts for annotations, denoting, for example, a tuple (a, b) with annotations *cl* and *op* as (a^{cl}, b^{op}).

For purely technical reasons (to deal with empty tables) we also have empty annotated tuples, denoted by $(\_, \alpha)$, where α is an annotation on the set of attributes.

If T is an annotated relation over $\mathsf{Const} \cup \mathsf{Null}$, in the semantics $\mathit{Rep}_A(T)$, after applying a valuation v to T, any tuple $(\ldots, a^{op}, \ldots)$ in $v(T)$ can be replicated arbitrarily many times with $(\ldots, b, \ldots)$, for $b \in \mathsf{Const}$. For example, $\mathit{Rep}_A(\{(a^{cl}, \bot^{op})\})$ contains all relations R whose projection on the first attribute is $\{a\}$, and $\mathit{Rep}_A(\{(a^{cl}, \bot^{cl})\})$ contains all one-tuple relations $\{(a, b)\}$ with $b \in \mathsf{Const}$.

Formally, if $T = \{(t_i, \alpha_i) \mid 1 \leq i \leq n\}$, then a relation R over Const is in $\mathit{Rep}_A(T)$ if, for some valuation v, the relation R contains the nonempty tuples among $v(t_1), \ldots, v(t_n)$, and every tuple $t \in R$ coincides with some $v(t_i)$ in all positions annotated by *closed* by α_i. Thus if α is an all-open annotation, then the tuple $(\_, \alpha)$ allows any tuple to be added to relations in $\mathit{Rep}_A(T)$; otherwise such tuples do not change the semantics. The difference between a tuple of *op*-annotated nulls and such $(\_, \alpha)$ is that the semantics of the latter also includes the empty table. Finally, $\mathit{Rep}_A(\cdot)$ extends naturally from relations to database instances.

For each annotated relation T, we denote by $\mathit{re\ell}(T)$ the pure relational part of T, i.e. if $T = \{(t_1, \alpha_1), \ldots, (t_n, \alpha_n)\}$, then $\mathit{re\ell}(T)$ is the set of non-empty tuples in $\{t_1, \ldots, t_n\}$.

Annotated canonical solution

Let $(\sigma, \tau, \Sigma_\alpha)$ be an annotated mapping (i.e., Σ_α is a set of annotated STDs). Let S be a source instance. The annotated canonical solution is defined by the same procedure as before, except that now it is populated with annotated tuples. That is, for each STD $\psi(\bar{x}, \bar{z})$:– $\varphi(\bar{x}, \bar{y})$, we evaluate φ over S, and for each tuple $(\bar{a}, \bar{b})$ in the result, we create a fresh tuple of nulls $\bar{\bot}$, and put annotated tuples in the solution to satisfy $\psi(\bar{a}, \bar{\bot})$, annotated as prescribed by α. If φ evaluates to the empty set over S, we add empty tuples for each atom in ψ, annotated according to α. The result is the *annotated canonical solution* denoted by $\text{CSOL}_{\text{A}}^{\Sigma_\alpha}(S)$, or just $\text{CSOL}_{\text{A}}(S)$, if the mapping is understood (the subscript 'A' distinguishes it from an unannotated solution).

In our previous example with $\sigma = \{E\}, \tau = \{R\}$, let the STD be $R(x^{cl}, z^{op})$:– $E(x, y)$. Then, if $E = \{(a, c_1), (a, c_2), (b, c_3)\}$, the canonical solution has annotated tuples $\{(a^{cl}, \bot_1^{op}), (a^{cl}, \bot_2^{op}), (b^{cl}, \bot_3^{op})\}$ in R.

Note that the same variable can be annotated differently in different atoms. For example, if we have an STD $R(x^{op}, z_1^{cl}) \wedge R(x^{cl}, z_2^{op})$:– $E(x, y)$ and a single tuple (a, c) in the source, then $\text{CSOL}_{\text{A}}(S) = \{(a^{op}, \bot_1^{cl}), (a^{cl}, \bot_2^{op})\}$.

Open (resp., closed) versions of the canonical solution capture the semantics of solutions in [11] and [21]. For reasons to become clear soon, we call the solutions of [11] *OWA-solutions*: i.e., an OWA-solution for a source S under Σ is any target instance T over $\mathsf{Const} \cup \mathsf{Null}$ such that $(S, T) \models \Sigma$. We then define

$$\begin{aligned} [\![S]\!]^{\Sigma}_{\text{OWA}} &= \{R \in \mathit{Rep}(T) \mid T \text{ an OWA-solution for } S\} \\ [\![S]\!]^{\Sigma}_{\text{CWA}} &= \{R \in \mathit{Rep}(T) \mid T \text{ a CWA-solution for } S\} \end{aligned}$$

These semantics produce sets of relations without nulls represented by OWA and CWA-solutions, respectively.

If Σ is a set of unannotated STDs, let Σ_{op} (resp., Σ_{cl}) be the set of all Σ-STDs where each variable is annotated with *op* (resp., *cl*). The following easy observations states that the canonical solutions under these two extremes capture the semantics of the unannotated OWA- and CWA-solutions:

Lemma 1. $[\![S]\!]^{\Sigma}_{\mathrm{OWA}} = Rep_A(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{op}}_{\mathrm{A}}(S))$ *and* $[\![S]\!]^{\Sigma}_{\mathrm{CWA}} = Rep_A(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{cl}}_{\mathrm{A}}(S))$.

Proof. First observe that, by the construction of $\mathrm{CS{\scriptstyle OL}}^{\Sigma_\alpha}_{\mathrm{A}}(S)$, we have $re\ell(\mathrm{CS{\scriptstyle OL}}^{\Sigma_\alpha}_{\mathrm{A}}(S)) = \mathrm{CS{\scriptstyle OL}}^{\Sigma}(S)$.

If an instance T is all-closed annotated, then $Rep_A(T)$ coincides with $Rep(rel(T))$ and, therefore, $Rep_A(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{cl}}_{\mathrm{A}}(S)) = Rep(\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S))$. It was shown in in [21] that $Rep(T) \subseteq Rep(\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S))$ for each CWA-solution T, and thus $[\![S]\!]^{\Sigma}_{\mathrm{CWA}} = Rep(\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S))$. This proves $[\![S]\!]^{\Sigma}_{\mathrm{CWA}} = Rep_A(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{cl}}_{\mathrm{A}}(S))$.

Similarly, since $\mathrm{CS{\scriptstyle OL}}^{\Sigma_{op}}_{\mathrm{A}}(S)$ has all-open annotation, $Rep_A(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{op}}_{\mathrm{A}}(S))$ consists of all target instances over Const that contain a valuation of $re\ell(\mathrm{CS{\scriptstyle OL}}^{\Sigma_{op}}_{\mathrm{A}}(S)) = \mathrm{CS{\scriptstyle OL}}^{\Sigma}(S)$. On the other hand, $[\![S]\!]^{\Sigma}_{\mathrm{OWA}}$ consists of all target instances J over Const such that $(S, J) \models \Sigma$.

So it remains to prove that a target instance J over Const contains a valuation of $\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S)$ if and only if $(S, J) \models \Sigma$. Indeed if $J \supseteq v(\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S))$, for some valuation v, then J contains an OWA-solution, and therefore $(S, J) \models \Sigma$. Conversely if $(S, J) \models \Sigma$, for each Σ-STD $\psi(\bar{x}, \bar{z}) \text{ :- } \varphi(\bar{x}, \bar{y})$ and for each pair of tuples $\bar{a}, \bar{b}$ such that $\varphi(\bar{a}, \bar{b})$ holds in S, let $\bar{c}$ the tuple of constants such that $\psi(\bar{a}, \bar{c})$ holds in J. Now define a valuation v of nulls of $\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S)$ such that $v(\bar{\bot}_{(\varphi,\psi,\bar{a},\bar{b})}) = \bar{c}$. Then clearly $v(\mathrm{CS{\scriptstyle OL}}^{\Sigma}(S)) \subseteq J$. □

Annotated solutions

We now define a general notion of solutions under annotated mappings using an approach similar to the CWA-solutions in Section 2, except that now we distinguish open and closed nulls. A *homomorphism* of annotated instances $h : T \to T'$ is a mapping from Null to Null so that for each annotated tuple (t, α) in a relation R in T, the tuple $(h(t), \alpha)$ is in R' – that is, homomorphisms preserve annotations (by $h(t)$ we denote the tuple obtained from t by replacing each null $\bot$ with $h(\bot)$).

Given an annotated mapping $(\sigma, \tau, \Sigma_\alpha)$ and a source S, each null in a target solution still needs to be justified by an STD $\psi \text{ :- } \varphi$ and a witness for φ. It is the annotation that will account for differences in the semantics: while closed nulls behave as nulls in CWA-solutions, open nulls can be instantiated by many values. Hence, we still define *presolutions* as homomorphic images of $\mathrm{CS{\scriptstyle OL}}_{\mathrm{A}}(S)$, since homomorphisms preserve annotations.

Our last requirement for CWA-solutions was that facts true in them must be implied by the source and the STDs, and thus true in $\mathrm{CS{\scriptstyle OL}}(S)$. We still want to apply this restriction, but only to closed nulls. For that, we use *annotated facts*, i.e. pairs $(f(\bar{a}), \alpha)$ where $f(\bar{a}) = \exists \bar{z}\ \gamma(\bar{a}, \bar{z})$ is a fact over the target schema, and α is an annotation over all atoms in γ. The notion of satisfaction is restricted to closed positions of T. That is, $T \models_{cl} \big(f(\bar{a}), \alpha\big)$ if there exists a tuple $\bar{\bot}$ of nulls such that for each atom $R(t)$ in $\gamma(\bar{a}, \bar{\bot})$, there is a tuple (t_0, α_0) in relation R of instance T which coincides with (t, α) in all positions annotated as *closed* in α_0.

Then a presolution T is a Σ_α-*solution* for S if each annotated fact that is true in T under $\models_{cl}$ is also true, under $\models_{cl}$, in the canonical solution $\mathrm{CS{\scriptstyle OL}}_{\mathrm{A}}(S)$.

If all annotations in Σ_α are *cl*, then $\models_{cl}$ is the usual notion of satisfaction, and thus Σ_α-solutions are precisely the CWA-solutions. If all annotations in Σ_α are *op*, then every fact is true under $\models_{cl}$ which means that under the OWA arbitrary facts could be true in solutions. We shall see soon that the semantics of all-open solutions is equivalent to the semantics of [11].

Example Consider an STD $R(x^{op}, z_1^{cl}) \wedge R(y^{cl}, z_2^{cl}) \text{ :- } S(x, y)$ and a source $S = \{(a, b)\}$ generating $\mathrm{CS{\scriptstyle OL}}_{\mathrm{A}}(S) = \{(a^{op}, \bot_1^{cl}), (b^{cl}, \bot_2^{cl})\}$. Let R be the presolution obtained by equating the two nulls:

$R = \{(a^{op}, \bot_1^{cl}), (b^{cl}, \bot_1^{cl})\}$. The fact $\exists z\ R(a^{op}, z^{cl}) \wedge R(b^{cl}, z^{cl})$, which is trivially true in R (under $\models_{cl}$) is also true in $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$ (under $\models_{cl}$) with $z = \bot_1$. In fact both atoms $R(a^{op}, \bot_1^{cl})$ and $R(b^{cl}, \bot_1^{cl})$ – obtained by assigning z the value $\bot_1$ in the fact – coincide with atom $R(a^{op}, \bot_1^{cl}))$ of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$ over closed positions of the latter. One similarly proves the same property for all other facts satisfied by the presolution R. Thus R is a Σ_α-solution. □

Annotated mappings: basic properties

We know that CWA-solutions have a homomorphism back into the canonical solution. A similar result is true for Σ_α-solutions, except that we need to expand the canonical solution, allowing for the open nulls to be replicated. We say that $T' \supseteq T$ is an *expansion* of T if every annotated tuple $t' \in T' - T$ coincides with some tuple $t \in T$ in all closed positions of t.

Proposition 1. *An annotated instance T is a Σ_α-solution iff it is a homomorphic image of* $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$*, and there is a homomorphism from T to an expansion of* $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$.

Proof. Assume that T is a Σ_α-solution and that $\bar{a} = (a_1, \ldots, a_n)$ lists the constants in T. We associate to each null $\bot$ occurring in T a distinct variable $z_\bot$, and let $\bar{z}$ be the tuple of variables associated to all nulls of T. Then with T we associate an annotated fact with $f(\bar{a}) = \exists \bar{z}\ diag_T^+(\bar{a}, \bar{z})$. Here $diag_T^+(\bar{a})$ is the positive diagram of T, i.e. the conjunction of all atoms $R(\bar{t})$ from T, where R is a relation of τ and $(\bar{t}, \alpha)$ is a non-empty tuple of R in T; furthermore, each null $\bot$ is replaced with $z_\bot$. The annotation α just follows the annotation of tuples in T.

Clearly $T \models_{cl} (f(\bar{a}), \alpha_T)$ with satisfying assignment $z_\bot = \bot$ for each variable $\bar{z}_\bot$ in $\bar{z}$. Since T is a Σ_α-solution, we know that $(f(\bar{a}), \alpha_T)$ is also satisfied (under $\models_{cl}$) in $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$, with some satisfying assignment $z_\bot = \bot'$ for each variable $z_\bot$ in $\bar{z}$.

Therefore, if we define a homomorphism h such that $h(\bot) = \bot'$ then, for each non-empty tuple $(\bar{t}, \alpha)$ in a relation R of T, $(h(\bar{t}), \alpha)$ coincides with some tuple (t', α') of R in $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$ on positions annotated as *cl* by α'. Moreover, since T is a homomorphic image of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$, each empty tuple occurring in some relation R of T also occurs in relation R of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$. In other words, h is a homomorphism from T to an expansion of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$.

Conversely assume that there exists a homomorphism h from T to an expansion C of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$. Take an arbitrary annotated fact $(f(\bar{a}), \alpha)$ satisfied in T under $\models_{cl}$, and let $f(\bar{a})$ be $\exists \bar{z}\ \gamma(\bar{a}, \bar{z})$. Let $\bar{\bot}$ be the assignment for which T satisfies $f(\bar{a})$.

We construct a target instance T_f from the satisfied fact as follows. For each annotated atom $(R(\bar{t}_0), \alpha_0)$ of $(\gamma(\bar{a}, \bar{\bot}), \alpha)$, add the tuple $(\bar{t}_0, \alpha_0)$ to relation R of T_f. Let h' be a mapping obtained by extending arbitrarily h to nulls occurring in $T_f - T$. We next prove that h' is a homomorphism from T_f to some expansion of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$. This will directly imply that $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S) \models_{cl} f(\bar{a})$ via the assignment $\bar{z} = h'(\bar{\bot})$.

Clearly, h' is an homomorphism from T_f to $h'(T_f) \cup C$. We now prove that $h'(T_f) \cup C$ is an expansion of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$. We know that for each annotated tuple $(\bar{t}_0, \alpha_0)$ of some relation R in T_f:

1. $(\bar{t}_0, \alpha_0)$ coincides with some tuple $(\bar{t}_1, \alpha_1)$ of R in T on positions annotated as *cl* by α_1 (since the fact $f(\bar{a})$ is satisfied in T);
2. $(h'(\bar{t}_1), \alpha_1)$ (i.e., $= (h(\bar{t}_1), \alpha_1)$) coincides with some tuple $(\bar{t}_2, \alpha_2)$ of R in $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$ on positions annotated as *cl* by α_2.

By 2, for each attribute A of R such that $\alpha_2(A) = cl$, we have $t_2(A) = h'(t_1(A))$ and $\alpha_1(A) = cl$. Therefore, by 1, $h'(t_1(A)) = h'(t_0(A))$ and $\alpha_0(A) = cl$. This shows that $(h'(\bar{t}_0), \alpha_0)$ coincides with $(\bar{t}_2, \alpha_2)$ on closed positions of α_2, implying that $h'(T_f) \cup C$ is an expansion of $\mathrm{CS}\textsc{ol}_{\mathrm{A}}(S)$. This concludes the proof of Proposition 1. □

Similarly to the semantics $[\![\cdot]\!]_{\text{CWA}}$ and $[\![\cdot]\!]_{\text{OWA}}$, we define the semantics for arbitrary annotated mappings:

$$[\![S]\!]^{\Sigma_\alpha} \;=\; \{R \in Rep_A(T) \mid T \text{ is a } \Sigma_\alpha\text{-solution}\}.$$

If α and α' are annotations of a set Σ of STDs, we write $\alpha \preceq \alpha'$ if for each occurrence of a variable in a Σ-STD, either both α and α' annotations are *cl*, or α' annotation is *op* (i.e., closed annotations can be extended to open). The following states that changing closed annotations to open makes the semantics larger, that the extreme points are the OWA and the CWA semantics of [11] and [21], and that for every annotated mapping, $[\![S]\!]^{\Sigma_\alpha}$ is determined by the annotated canonical solution.

Theorem 1. *If Σ is a set of STDs and S is a source instance, then*

1. $[\![S]\!]^{\Sigma_{cl}} = [\![S]\!]^{\Sigma}_{\text{CWA}}$.
2. $[\![S]\!]^{\Sigma_{op}} = [\![S]\!]^{\Sigma}_{\text{OWA}}$.
3. *If $\alpha \preceq \alpha'$ then $[\![S]\!]^{\Sigma_\alpha} \subseteq [\![S]\!]^{\Sigma_{\alpha'}}$.*
4. $[\![S]\!]^{\Sigma_\alpha} = Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$.

Proof. We start by proving 4). It suffices to prove that $Rep_A(T) \subseteq Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$ for each Σ_α-solution T. For an arbitrary Σ_α-solution T and an arbitrary $J \in Rep_A(T)$, let h be a homomorphism such that $h(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)) = T$, and v a valuation witnessing $J \in Rep_A(T)$. We next prove that the valuation $v\ \circ h$ on nulls of $\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)$ witnesses $J \in Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$.

We know that $J \supseteq v(re\ell(T))$, thus $J \supseteq v \circ h(re\ell(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)))$. Moreover, for each tuple $\bar{t}$ in a relation R of J there exists an annotated tuple $(\bar{t}_0, \alpha)$ of R in T such that $\bar{t}$ and $v(\bar{t}_0)$ coincide on positions annotated as closed by α, and there exists an annotated tuple (t_1, α) of R in $\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)$ such that $h(\bar{t}_1) = \bar{t}_0$.

Then $v \circ h(\bar{t}_1)$ coincides with $\bar{t}$ on positions annotated as closed by α. This shows that $J \in Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$ via the homomorphism $v \circ h$ and concludes the proof of 4.

Items 1) and 2) follow directly from 4) and Lemma 1.

We now prove 3). Assume that $\alpha \preceq \alpha'$ and $J \in [\![S]\!]^{\Sigma_\alpha}$. By 4), $J \in Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$, and therefore there exists a valuation v such that

$$J \;\supseteq\; v(re\ell(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))).$$

By the construction of the canonical solution, we have $re\ell(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)) = re\ell(\text{CSol}_{\text{A}}^{\Sigma_{\alpha'}}(S))$, and thus $J \supseteq v(re\ell(\text{CSol}_{\text{A}}^{\Sigma_{\alpha'}}(S)))$.

Now consider an arbitrary tuple $\bar{t}$ of some relation R of J, and let $(\bar{t}_0, \alpha_0)$ be an annotated tuple of relation R in $\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S)$ such that $\bar{t}$ and $v(\bar{t}_0)$ coincide on closed positions of α_0. We know, again by the construction of the canonical solution, that there exists a tuple $(\bar{t}_0, \alpha_1)$ in relation R of $\text{CSol}_{\text{A}}^{\Sigma_{\alpha'}}(S)$ with $\alpha_0 \preceq \alpha_1$. Observe that positions annotated as closed by α_1 are also annotated as closed by α_0. Hence, $\bar{t}$ and $v(\bar{t}_0)$ coincide on all closed positions of α_1. This proves that $J \in Rep_A(\text{CSol}_{\text{A}}^{\Sigma_{\alpha'}}(S))$, that is by 4), $J \in [\![S]\!]^{\Sigma_{\alpha'}}$, thus showing 3) and concluding the proof of Theorem 1. □

There is a natural decision problem of recognizing instances in $[\![S]\!]^{\Sigma_\alpha}$. We next show that this problem is in PTIME if all annotations in Σ_α are open, but any presence of closed annotations leads to NP-completeness. More precisely we introduce the parameter $\#_{cl}(\Sigma_\alpha)$ denoting the *maximum number of closed positions* per atom in an STD in a set of annotated STDs Σ_α. For example, for the rule $T(x^{cl}, y^{op}) \wedge T(x^{cl}, z^{op})$:– φ, the value of $\#_{cl}(\Sigma_\alpha)$ is 1.

The complexity of recognizing instances in $[\![S]\!]^{\Sigma_\alpha}$ can be classified based on the parameter $\#_{cl}(\Sigma_\alpha)$ as follows.

Theorem 2. *The problem of checking, for source and target instances S and T, whether $T \in [\![S]\!]^{\Sigma_\alpha}$ is always in* NP*, and furthermore:*

- *it is in* PTIME *if all annotations in* Σ_α *are open (that is if* $\#_{cl}(\Sigma_\alpha) = 0$*);*
- *for each* $k > 0$*, there is a mapping* Σ_α *with* $\#_{cl}(\Sigma_\alpha) = k$*, such that the problem of checking* $T \in [\![S]\!]^{\Sigma_\alpha}$ *is* NP*-complete.*

Proof. If all variables are annotated as open by α, then by Theorem 1 (item 2) one can check that $T \in [\![S]\!]^{\Sigma_\alpha}$ by simply checking that $(S, T) \models \Sigma$ which can be done in polynomial time.

Otherwise by Theorem 1 (item 4) one can check $T \in [\![S]\!]^{\Sigma_\alpha}$ by first guessing a valuation v on nulls of $\mathrm{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ and then checking that 1) $T \supseteq v(rel(\mathrm{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)))$ and 2) each tuple of T coincides with some tuple of $\mathrm{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ on closed positions. All this can be checked in polynomial time since $\mathrm{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ has size polynomial in S.

This shows that checking whether T belongs to $[\![S]\!]^{\Sigma_\alpha}$ is in NP for an arbitrary annotation α; we now show it is NP-hard when $\#_{cl}(\Sigma_\alpha) = k$, for all $k > 0$. We only show the reduction for the case $\#_{cl}(\Sigma_\alpha) = 1$, for all other values of $k > 1$, the same reduction will hold after replicating k times one of the closed variables in Σ_α.

We use a reduction from the *tripartite matching*. The input of *tripartite matching* is given by three disjoint sets B_0, G_0 and H_0 of the same size n, and a compatibility relation $C_0 \subseteq B_0 \times G_0 \times H_0$. The problem asks whether there exists a subset X_0 of n triples of C_0 such that all elements of B_0, G_0 and H_0 occur in X_0.

From an input $\langle B_0, G_0, H_0, C_0\rangle$ of *tripartite matching* we construct a pair of source and target instances (S, T) for the following annotated schema mapping:

- $\sigma = \{N, C'\}$, where N is unary and C' is ternary;
- $\tau = \{B, G, H, C\}$, where B, G, H are unary and C is ternary;
- Σ_α:

$$\begin{array}{rcl} C(x^{op}, y^{op},\ z^{op}),\ B(x^{cl}),\ G(y^{cl}),\ H(z^{cl}) & \text{:-} & N(w) \\ C(x^{op}, y^{op}, z^{op}) & \text{:-} & C'(x, y, z) \end{array}$$

Note that the maximum number of closed-annotated attributes per atom is 1, that is $\#_{cl}(\Sigma_\alpha) = 1$.

The source instance S interprets the relation N as $\{1, \ldots, n\}$, and the relation C' as C_0. The target instance interprets the relations C, B, G and H as C_0, B_0, G_0 and H_0, respectively (i.e., the input of the *tripartite matching* problem). We now prove that $T \in [\![S]\!]^{\Sigma_\alpha}$ iff there exists a subset of n triples of C_0 covering all elements of $B_0 \cup G_0 \cup H_0$.

First, $T \in [\![S]\!]^{\Sigma_\alpha}$ iff $T \in Rep_A(\mathrm{CSol}_{\mathrm{A}}(S)))$. We compute $\mathrm{CSol}_{\mathrm{A}}(S)$, in which the values of relations B, G, H and C are:

- $B = \{\bot_{bi}^{cl} \mid i = 1, \ldots, n\}$;
- $G = \{\bot_{gi}^{cl} \mid i = 1, \ldots, n\}$;
- $H = \{\bot_{hi}^{cl} \mid i = 1, \ldots, n\}$;
- $C = C_0 \cup \{(\bot_{bi}^{op}, \bot_{gi}^{op}, \bot_{hi}^{op}) \mid i = 1, \ldots, n\}$.

Intuitively, relation N in the source instance represents the set of n choices of triples, and for each choice $i \in \{1, \ldots, n\}$, the tuple $(\bot_{bi}^{op}, \bot_{gi}^{op}, \bot_{hi}^{op})$ in relation C of $\mathrm{CSol}_{\mathrm{A}}(S)$ represents the chosen triple; the values occurring in the three components of the chosen triples are collected, with *closed* annotation, in relations B, G and H of $\mathrm{CSol}_{\mathrm{A}}(S)$, respectively.

Assume now that the instance of *tripartite matching* has a solution, witnessed by a subset $\{\langle b_i, g_i, h_i\rangle \mid i = 1, \ldots, n\}$ of C_0. Define a valuation v on nulls of $\text{CS}\text{OL}_{\text{A}}(S)$ such that, for $i = 1, \ldots, n$, we have

$$v(\bot_{bi}) = b_i, \quad v(\bot_{gi}) = g_i, \quad v(\bot_{hi}) = h_i.$$

Then one can easily check that $v(\mathit{re\ell}(\text{CS}\text{OL}_{\text{A}}(S))) = T$, and thus $T \in \mathit{Rep}_A(\text{CS}\text{OL}_{\text{A}}(S))$.

Conversely, assume that $T \in \mathit{Rep}_A(\text{CS}\text{OL}_{\text{A}}(S)))$, and let v a valuation witnessing this. Then we have:

$$\begin{array}{rcl} B_0 & = & v(\{\bot_{bi} \mid i = 1, \ldots, n\}), \\ G_0 & = & v(\{\bot_{gi} \mid i = 1, \ldots, n\}), \\ H_0 & = & v(\{\bot_{hi} \mid i = 1, \ldots, n\}), \\ C_0 & \supseteq & C_0 \cup v(\{(\bot_{bi}, \bot_{gi}, \bot_{hi}) \mid i = 1, \ldots, n\}). \end{array}$$

Therefore, $v(\{(\bot_{bi}, \bot_{gi}, \bot_{hi}), i = 1, \ldots, n\})$ is a subset of C_0 whose triples cover all elements of B_0, G_0 and H_0, and thus it gives a solution of the *tripartite matching* problem. This completes the proof of Theorem 2. □

Notice that the reduction shown in the proof of Theorem 2 is still valid if all annotations in Σ_α are turned to closed. Hence the following corollary holds:

Corollary 1. *There exists a mapping Σ_α, having all-closed annotation, such that the problem of checking, for source and target instances S and T, whether $T \in [\![S]\!]^{\Sigma_\alpha}$ is* NP*-complete.*

Note that the complexity of recognizing instances representing tables with incomplete information normally increases with additional constraints on nulls: for example, checking if an instance R is in $\mathit{Rep}(S)$ is in PTIME if S is a Codd table (which cannot equate nulls), but the same problem is NP-complete for naive tables, which can equate nulls [2]. Thus, it is natural that the complexity of this particular recognition problem increases as one allows closed variables, which introduce extra constraints on nulls. But as we shall see soon, most of the time it suffices to work with the canonical solution, which can be constructed in PTIME regardless of annotation, and thus the higher complexity of $[\![\cdot]\!]^{\Sigma_\alpha}$ with closed annotations will not affect problems such as query answering.

4. Query answering

Query answering in data exchange normally means finding certain answers. Since the notion of $Q(T)$, where T is a solution, is not well-defined due to T containing nulls, we must find certain answers to Q over each solution T, and then find tuples that belong to such certain answers over all solutions T. That is, given an annotated mapping with STDs Σ_α, a source instance S and a query Q, we define

$$\mathsf{certain}_{\Sigma_\alpha}(Q, S) = \bigcap_{T \text{ is a } \Sigma_\alpha-\text{solution}} \quad \bigcap_{R \in \mathit{Rep}_A(T)} Q(R)$$

We compare this with two existing notions of certain answers in data exchange. The original open-world notion $\mathsf{certain}^{\text{OWA}}_{\Sigma}(Q, S)$ of [11] and many others was defined as the set of tuples that belong to $Q(T)$ for every OWA-solution T, where Q is evaluated under the naive semantics. In [21, 16], $\mathsf{certain}^{\text{CWA}}_{\Sigma}(Q, S)$ was defined as the set of tuples in all $Q(R)$'s where R ranges over $\mathit{Rep}(T)$ for CWA-solutions T. Using a simple observation that in the definition of [11] it suffices to look only at instances over Const, we show:

Proposition 2. *If Σ is an arbitrary set of STDs, and Σ_{op} and Σ_{cl} are its annotations that assign op (resp., cl) to each variable, then*

$$\begin{array}{rcl} \mathsf{certain}^{\text{OWA}}_{\Sigma}(Q, S) & = & \mathsf{certain}_{\Sigma_{op}}(Q, S) \\ \mathsf{certain}^{\text{CWA}}_{\Sigma}(Q, S) & = & \mathsf{certain}_{\Sigma_{cl}}(Q, S) \end{array}$$

Furthermore, for an arbitrary annotation α,

$$\mathsf{certain}_{\Sigma_{op}}(Q,S) \subseteq \mathsf{certain}_{\Sigma_\alpha}(Q,S) \subseteq \mathsf{certain}_{\Sigma_{cl}}(Q,S).$$

Proof. The statement for *closed* annotation is a direct corollary of Theorem 1 (item 1). In the case of *open* annotation, the result is based on the following claim:

Claim 1. *For each OWA-solution T containing nulls there exist OWA-solutions R_1 and R_2 over* Const *such that*

$$Q(T) \cap Q(R_1) \cap Q(R_2) \ = \ Q(R_1) \cap Q(R_2),$$

where $Q(T)$ is computed according to the naive evaluation.

Proof of Claim 1. We consider OWA-solutions $v_1(T)$ and $v_2(T)$, where v_1 and v_2 are valuations of nulls of T which map distinct nulls to distinct constants not occurring in T and have disjoint ranges. By the genericity of Q we conclude $Q(v_1(T)) = v_1(Q(T))$ and $Q(v_2(T)) = v_2(Q(T))$.

Let $Q(T)\!\downarrow$ denote the set of tuples of $Q(T)$ which do not contain nulls. Since v_1 and v_2 have disjoint ranges, we have $v_1(Q(T)) \ \cap \ v_2(Q(T)) = Q(T)\!\downarrow$ and thus $Q(T) \ \cap \ Q(v_1(T)) \ \cap \ Q(v_2(T)) = Q(T)\!\downarrow = Q(v_1(T)) \ \cap \ Q(v_2(T))$, as claimed. □

From Claim 1 it follows that

$$\bigcap_{T \text{ is an OWA-solution}} Q(T) \ = \ \bigcap_{T \text{ is an OWA-solution over } \mathsf{Const}} Q(T)$$

and thus

$$\mathsf{certain}^{\mathrm{OWA}}_{\Sigma}(Q,S) \ = \ \bigcap_{T \in [\![S]\!]^{\Sigma}_{\mathrm{OWA}}} Q(T) \ = \ \mathsf{certain}_{\Sigma_{op}}(Q,S)$$

(the last equation follows from Theorem 1, item 2), which proves Proposition 2. □

Hence the semantics of [11] and [21] are indeed the two extreme semantics. For one class of queries, which was the focus of several papers on data exchange [11, 12, 3], the semantics coincide, regardless of annotations (we recall that positive relational algebra refers to the fragment of relational algebra allowing only projection, union, product and selection with positive Boolean combinations of equalities).

Proposition 3. *Let (σ, τ, Σ) be a mapping, Σ_α an arbitrary annotation of Σ, and Q a positive relational algebra query. Then*

$$\mathsf{certain}_{\Sigma_\alpha}(Q,S) \ = \ \Box Q(\mathrm{CSOL}^{\Sigma}(S)).$$

Proof. We prove the statement in the more general case that Q is a monotone query. By Theorem 1 (item 4), each $R \in [\![S]\!]^{\Sigma_\alpha}$ contains a valuation $v(\mathit{rel}(\mathrm{CSOL}_{\mathrm{A}}(S)))$; as $\mathit{rel}(\mathrm{CSOL}_{\mathrm{A}}(S)) = \mathrm{CSOL}(S)$, each $R \in [\![S]\!]^{\Sigma_\alpha}$ contains an instance of $\mathit{Rep}(\mathrm{CSOL}(S))$. Therefore, by monotonicity of Q

$$\bigcap_{R \in [\![S]\!]^{\Sigma_\alpha}} Q(R) \ \supseteq \ \bigcap_{R \in \mathit{Rep}(\mathrm{CSOL}(S))} Q(R),$$

which implies $\mathsf{certain}_{\Sigma_\alpha}(Q,S) \supseteq \Box Q(\mathrm{CSOL}(S))$.

On the other hand, each instance in $\mathit{Rep}(\mathrm{CSOL}(S))$ is also in $\mathit{Rep}_A(\mathrm{CSOL}_{\mathrm{A}}(S))$, thus in $[\![S]\!]^{\Sigma_\alpha}$. This proves the reverse inclusion, and hence $\Box Q(\mathrm{CSOL}(S)) = \mathsf{certain}_{\Sigma_\alpha}(Q,S)$. □

Thus, to compute certain answers for positive queries, one can simply construct the canonical solution and apply the standard naive evaluation [17] to compute $\Box Q$ over it, as was done in [11].

We now study the general case. Our goal is to find $\mathsf{certain}_{\Sigma_\alpha}(Q,S)$ using one materialized target. We can use the annotated canonical solution as this target. Indeed, under the natural notion of certain answers in annotated instances defined as $\Box Q(T) \ = \ \bigcap\{Q(R) \ \mid \ R \in \mathit{Rep}_A(T)\}$, we conclude, from Theorem 1:

Corollary 2. $\mathsf{certain}_{\Sigma_\alpha}(Q,S) = \Box Q(\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S))$.

We know that $\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ can be constructed in polynomial time. Thus, to describe the complexity of query answering in data exchange, we need to determine the complexity of finding $\Box Q$. We do this for relational algebra (i.e., FO) queries. Consider the problem $\mathrm{DEQA}(\Sigma_\alpha, Q)$ of data exchange query answering for an annotated mapping $(\sigma, \tau, \Sigma_\alpha)$ and a query Q:

Problem	$\mathrm{DEQA}(\Sigma_\alpha, Q)$
Input:	a source database S, a tuple t
Question:	is $t \in \mathsf{certain}_{\Sigma_\alpha}(Q,S)$.

Some partial answers under CWA or OWA are known [1, 4, 11, 21]. We now classify the complexity of $\mathrm{DEQA}(\Sigma_\alpha, Q)$ for FO queries Q using, as the main parameter, the *maximum number of open positions* per atom in an STD in a set of annotated STDs Σ_α. It is denoted by $\#_{op}(\Sigma_\alpha)$.

In a CWA mapping, $\#_{op}(\Sigma_\alpha) = 0$ (since there are no open positions), and in an all-open mapping, it is the maximum arity of a relation. Note that we measure the number of open annotations per atom and not per rule. For example, for the rule $T(x^{cl}, y^{op}) \wedge T(x^{cl}, z^{op}) \coloneq \varphi$, the value of $\#_{op}(\Sigma_\alpha)$ is 1, even though two variables occur with an open annotation.

We prove the following *trichotomy* result – the complexity of $\mathrm{DEQA}(\Sigma_\alpha, Q)$ for FO queries is:

- coNP-complete if $\#_{op}(\Sigma_\alpha) = 0$;
- coNEXPTIME-complete if $\#_{op}(\Sigma_\alpha) = 1$;
- undecidable if $\#_{op}(\Sigma_\alpha) > 1$.

Theorem 3. *The complexity of* $\mathrm{DEQA}(\Sigma_\alpha, Q)$ *for* FO *queries is as follows:*

1. *if* $\#_{op}(\Sigma_\alpha) = 0$*, then* $\mathrm{DEQA}(\Sigma_\alpha, Q) \in \mathrm{coNP}$*, and there exists a mapping with* $\#_{op}(\Sigma_\alpha) = 0$ *and an* FO *query* Q *so that* $\mathrm{DEQA}(\Sigma_\alpha, Q)$ *is* coNP*-hard;*
2. *if* $\#_{op}(\Sigma_\alpha) = 1$*, then* $\mathrm{DEQA}(\Sigma_\alpha, Q)$ *is in* coNEXPTIME*, and there exists a mapping with* $\#_{op}(\Sigma_\alpha) = 1$ *and an* FO *query* Q *so that* $\mathrm{DEQA}(\Sigma_\alpha, Q)$ *is* coNEXPTIME*-hard;*
3. *if* $k > 1$*, then there is a mapping with* $\#_{op}(\Sigma_\alpha) = k$ *and an* FO *query* Q *so that* $\mathrm{DEQA}(\Sigma_\alpha, Q)$ *is undecidable.*

The main result is the decidable case 2 (others are easy adaptations of known techniques [1, 2, 11, 21]). Below presenting the proof, we give a sketch of a simpler result showing that for $\#_{op}(\Sigma_\alpha) = 1$, the query answering problem could be hard for an arbitrary level of the polynomial hierarchy (PH). This indicates the main source of complexity (and a more detailed proof tightens it to coNEXPTIME-completeness).

Suppose the source database is a graph with vertices $V(\cdot)$ and edges $E(\cdot,\cdot)$, the target schema has two binary relations, and the STDs are

$$\begin{array}{lcl} E'(x^{cl}, y^{cl}) & \coloneq & E(x,y) \\ P(x^{cl}, z^{op}) & \coloneq & V(x) \end{array}$$

That is, E' is a copy of the graph, and P assigns open nulls to vertices: the semantics of P is any relation whose first projection is V.

We next consider a sentence Φ_p saying that P encodes the powerset of the set of vertices (i.e., for each value a of the first attribute of P, there is a c so that $P(a,c)$ holds, and no other $P(\cdot, c)$ holds; and, for any c_1, c_2, there is a c so that $\{a \mid P(a,c)\} = \{a \mid P(a,c_1)\} \cup \{a \mid P(a,c_2)\}$ – all these are easily stated in FO). Let Ψ be an arbitrary monadic second-order sentence over E. If P encodes the powerset on V, we can easily restate Ψ as an FO sentence ψ over the schema $\{E', P\}$. Thus, the certain answer of $\Phi_p \to \psi$ is true iff the original graph satisfies Ψ. But it is well-known that in monadic second-order logic one can encode problems complete for all levels of PH [25] – hence query answering is hard for every level of PH.

Proof of Theorem 3. We start with the easy cases of $\#_{op}(\Sigma_\alpha) = 0$ and $\#_{op}(\Sigma_\alpha) > 1$.

For $\#_{op}(\Sigma_\alpha) = 0$, the statement follows directly from results of [21]: in fact if $\#_{op}(\Sigma_\alpha) = 0$ then, by Proposition 2, $\mathsf{certain}_{\Sigma_\alpha}(Q, S) = \mathsf{certain}_\Sigma^{\mathrm{CWA}}(Q, S)$, and the problem of checking $\bar{t} \in \mathsf{certain}_\Sigma^{\mathrm{CWA}}(Q, S)$, for any fixed first order query Q and mapping Σ, is in coNP [21]. Moreover in [21] a mapping $\bar{\Sigma}$ and an FO query $\bar{Q}$ are constructed, such that checking $\bar{t} \in \mathsf{certain}_{\bar{\Sigma}}^{\mathrm{CWA}}(\bar{Q}, S)$ is coNP-hard.

The undecidability for $\#_{op}(\Sigma_\alpha) > 1$ can be proved by reduction from finite validity of first order sentences, similarly to [1]. More precisely, for a fixed first order sentence φ of relational vocabulary τ with at least one symbol of arity 2 of greater, we consider the following problem which we denote by $\mathrm{VAL}(\varphi)$: given an input finite structure M_0 of vocabulary τ, it is to decide whether all finite structures $M \supseteq M_0$ are models of φ. It follows easily from the proof of Trakhtenbrot's theorem that for each $k > 1$, there exists a relational vocabulary τ, whose maximum arity of relations is k, and a first order sentence φ_k of vocabulary τ for which $\mathrm{VAL}(\varphi_k)$ is undecidable.

Now from each first order sentence φ of relational vocabulary τ we show how to construct a set of annotated STDs $\Sigma_\alpha(\varphi)$ and a first order query Q_φ so that $\#_{op}\Sigma_\alpha(\varphi)$ coincides with the maximum arity in τ, and $\mathrm{VAL}(\varphi)$ has a reduction to $\mathrm{DEQA}(\Sigma_\alpha(\varphi), Q_\varphi)$.

The target schema of $\Sigma_\alpha(\varphi)$ is $\tau \cup \{U\}$, with U unary, and source schema is $\tau' \cup \{U'\}$, where τ' has a relation R' for each R in τ, with the same arity as R. The mapping given by $\Sigma_\alpha(\varphi)$ is a copying schema mapping from each relation V' in the source schema to the corresponding relation V in the target schema, and α annotates each position as *open*. The query Q_φ is a unary query $U(x) \wedge \varphi$.

The problem $\mathrm{VAL}(\varphi)$ can be reduced to $\mathrm{DEQA}(\Sigma_\alpha, Q)$, for $\Sigma_\alpha = \Sigma_\alpha(\varphi)$ and $Q = Q(\varphi)$, as follows: for each input structure M_0 of universe U_0 we construct a source instance S having $U' = U_0$, $R' = R^{M_0}$ for each relational symbol R of τ. Of course, instances $J \in Rep_A(\mathrm{CSOL}_\mathrm{A}(S))$ represent all possible finite structures containing M_0, where U^J represents the universe of the structure and R^J, $R \in \tau$, the interpretation of relations. Moreover for each $J \in Rep_A(\mathrm{CSOL}_\mathrm{A}(S))$, we have $Q(J) = U^J \supseteq U_0$ if $J \models \varphi$, and $Q(J)$ is empty otherwise. Therefore if we let t be an arbitrary value in U_0, then $t \in \mathsf{certain}_{\Sigma_\alpha}(Q, S)$ if and only if φ holds in all $J \in Rep_A(\mathrm{CSOL}_\mathrm{A}(S))$, that is if and only if φ holds in all structures containing M_0. This completes the reduction and proves that for all $k > 1$ the problem $\mathrm{DEQA}(\Sigma_\alpha(\varphi_k), Q_{\varphi_k})$, where $\#_{op}\Sigma_\alpha(\varphi_k) = k$, is undecidable.

We now move to the main case of the theorem – the proof for $\#_{op}(\Sigma_\alpha) = 1$. We first use a games argument to establish an exponential bound on the number of replicated open nulls in a possible witness for $t \notin \mathsf{certain}_{\Sigma_\alpha}(Q, S)$, and then code a version of the tiling problem.

Membership. We need to show that the problem of checking whether $\bar{t} \in \mathsf{certain}_{\Sigma_\alpha}(Q, S)$ is in coNEXPTIME. If Q is expressed by an FO formula φ, let $\bar{Q}$ be the query expressed by $\neg\varphi$. Given an input instance S and a tuple $\bar{t}$, the complexity of checking $\bar{t} \notin \mathsf{certain}_{\Sigma_\alpha}(Q, S)$ coincides with the complexity of checking whether there exists an instance $I \in Rep_A(\mathrm{CSOL}_\mathrm{A}(S))$ such that $\bar{t} \in \bar{Q}(I)$. This is proved to be in NEXPTIME in Lemma 2 below (with X empty). The case of nonempty X will be used when we apply this lemma in the proof of Theorem 4.

Lemma 2. *For a fixed first order query Q over schema τ and a subset X of attributes of Q, given an input consisting of:*

1. *an annotated instance T of schema τ whose tuples have at most one position annotated as open;*
2. *a relation W over* Const *of the same arity as Q;*

the problem of checking whether there exists an instance $I \in Rep_A(T)$ such that

- $W \subseteq Q(I)$ *and*
- $\pi_X(W) = \pi_X(Q(I))$

is in NEXPTIME.

Proof. Let Q be expressed by an FO formula $\varphi(\bar{x}, \bar{y})$, where the variables $\bar{x}$ correspond to attributes X of Q. In the rest of the proof we denote by $qr(\psi)$ the quantifier rank of an FO formula ψ, and we let k be $qr(\varphi(\bar{x}, \bar{y})) + |\bar{y}|$. We let C_φ be the set of constants occurring in $\varphi(x)$ and n and m the number of tuples in T and W, respectively.

We first prove that if there exists $I \in Rep_A(T)$ satisfying the properties required by the lemma, then there exists also $I' \in Rep_A(T)$ with $\|I'\|$ exponential in the size of the input, still satisfying the same properties. This will prove that there exists a NEXPTIME algorithm that guesses I' and checks $W \subseteq Q(I')$ and $\pi_X(W) = \pi_X(Q(I'))$.

We now give the intuition for the existence of I' (at least in the case that X is empty), before proving it formally. If $I \in Rep_A(T)$, this is witnessed by some valuation v of nulls of T. Then tuples of I are of the form $(v(\bar{t}_1), a, v(\bar{t}_2))$, where $(\bar{t}_1^{cl}, s^{op}, \bar{t}_2^{cl})$ is a tuple of T for some constant or null s. The subset of I where a is a "known" constant (that is occurring in either $v(rel(T))$ or W or φ) is clearly of polynomial size. Now consider tuples of I where a is an external (that is not "known") constant; in principle there is no bound on the size of this subset of I. Nevertheless we show that some of these tuples can be safely removed, to get an instance I' which is clearly still in $Rep_A(T)$, and still satisfies $\varphi(\bar{t})$ for all $\bar{t}$ in W. The idea is that each external constant in I is "connected" (that is occurs together) with some subset of tuples of the form $(v(\bar{t}_1), v(\bar{t}_2))$. To each subset S of such tuples we can then associate the set of external constants connected precisely to S. Intuitively, these external constants are all "equivalent" in the sense that the substructures of I where they occur are isomorphic. It is then natural to expect – as we will prove formally later – that if the set of constants connected precisely to S is "very large", then one can bound its size to a constant depending on the quantifier rank of φ, by removing tuples from I. It will then be possible to play $qr(\varphi)$ rounds of the Ehrenfeucht-Fraïssé game with I and its reduced version. In fact when the spoiler plays an external constant a connected to S in I, the duplicator will always be able to reply with another external constant connected to S.

If the restriction of I is done for every subset S of tuples of the form $(v(\bar{t}_1), v(\bar{t}_2))$, one then gets an instance I' that φ cannot distinguish from I. The instance I' has exponential size since there are exponentially many subsets of tuples of the form $(v(\bar{t}_1), v(\bar{t}_2))$, and each of them is possibly connected only to a constant number of external values in I'.

Now we formalize this and prove the existence of I'. Assume that $I \in Rep_A(T)$, that $W \subseteq Q(I)$ and $\pi_X(W) = \pi_X(Q(I))$. Let v be a valuation of nulls of T witnessing $I \in Rep_A(T)$. We denote by K the union of the following two sets of triples:

- the set of triples $\langle R; v(\bar{t}_1); v(\bar{t}_2)\rangle$ such that R is a relation symbol in τ and the non-empty annotated tuple $(\bar{t}_1^{cl}, a^{op}, \bar{t}_2^{cl})$ is in R^T, for some $a \in \mathsf{Const} \cup \mathsf{Null}$ (with $\bar{t}_1$ and/or $\bar{t}_2$ possibly empty);
- the set of triples $\langle R; \_; \_\rangle$ such that R is a unary relation symbol in τ and $(\_, op) \in R^T$.

Moreover let V stand for $v(re\ell(T))$ and let C stand for $D_V \cup D_W \cup C_\varphi$ (recall that D_J denotes the active domain of instance J). Note that $|K| \leq n$ and constants occurring in triples of K are all in D_V thus in C.

Given a subset U of Const and $X \subseteq K$, we denote by $X \times U$ the following target instance. For each relation symbol $R \in \tau$, its interpretation in $X \times U$ is defined as

$$R^{X \times U} := \{(\bar{c}_1, c, \bar{c}_2) \mid \langle R; \bar{c}_1; \bar{c}_2\rangle \in X \text{ and } c \in U\}.$$

The following claim shows how I and K are related.

Claim 2. *There exist $E_0 \subseteq K \times C$ and $E \subseteq K \times (D_I - C)$, with E_0, E and V pairwise disjoint, such that $I = V \cup E_0 \cup E$.*

Proof of Claim 2. Since I is in $Rep_A(T)$ via the valuation v, then $I \supseteq V$. Moreover each tuple $\bar{t}$ of some relation R in $I - V$ has to coincide with the valuation of some tuple of R^T, on positions annotated as closed. We know that this tuple of R^T cannot be empty, unless it has an all-open annotation, and it cannot have an all-closed annotation, otherwise $\bar{t}$ would be in V. Then $\bar{t}$ is either of the form $(v(\bar{t}_1), c, v(\bar{t}_2))$, where $c \in D_I$ and $(\bar{t}_1^{cl}, a^{op}, \bar{t}_2^{cl})$ is a non-empty tuple of R^T, or of the form (c) where $c \in D_I$ and R^T is a unary relation containing $(\_, op)$. In both cases $\bar{t} \in K \times D_I$. As a consequence $I - V$ can be partitioned into two sub-instances: E_0, whose tuples are in $K \times C$, and E whose tuples are in $K \times (D_I - C)$. This completes the proof of the claim. □

Now we give more details about the structure of E. For each element $d \in D_I - C$, let $X(d)$ be the maximal subset of K such that $X(d) \times \{d\} \subseteq E$. Then $E = \bigcup_{d \in D_I - C} X(d) \times \{d\}$.

For each $X \subseteq K$, we define a set C_X as $\{d \in D_I - C \mid X(d) = X\}$. Note that these sets form a partition of $D_I - C$. Now for each set C_X such that $|C_X| > k + arity(Q)$ we choose arbitrarily a subset $C'_X \subseteq C_X$ such that $|C'_X| = k + arity(Q)$. For all sets C_X such that $|C_X| \leq k + arity(Q)$, we let $C'_X = C_X$. Let I' be the instance obtained from I by removing all tuples containing constants in $\bigcup_{X \subseteq K} C_X - C'_X$. Observe that $D_{I'} = (D_I \cap C) \cup \bigcup_{X \subseteq K} C'_X$. Moreover, since constants removed from I are not in C, we have

$$I' = V \cup E_0 \cup E',$$

where

$$E' = \bigcup_{d \in D_{I'} - C} X(d) \times \{d\}$$

If we measure the size of an instance as the number of tuples in it, then $\|I'\| = \|V\| + \|E_0\| + \|E'\|$, where:

- $\|V\| = n$;
- $\|E_0\| \leq |K| \times |C| = O(n(n+m))$;
- $\|E'\| \leq (k + arity(Q)) \cdot n \cdot 2^n$.

To see the last point, note that $\|E'\| = \sum_{d \in D_{I'} - C} |X(d)| \leq |D_{I'} - C| \cdot n$. Since sets C'_X are pairwise disjoint and of size at most $k + arity(Q)$, we have $|D_{I'} - C| = \sum_{X \subseteq K} |C'_X| \leq (k + arity(Q)) \cdot 2^n$, from which the bound follows.

The instance I' is still in $Rep_A(T)$, as it contains $V = v(re\ell(T))$ and is contained in I. Its size, as shown above, is at most exponential in n.

In what follows we prove that, if $\psi(\bar{z})$ stands either for $\varphi(\bar{x}, \bar{y})$ or for $\exists \bar{y} \varphi(\bar{x}, \bar{y})$, then $I' \models \psi(\bar{t})$ if and only if $I \models \psi(\bar{t})$, for each tuple $\bar{t}$ over $D_{I'} \cup C_\varphi$.

First note that the set of constants occurring in $\psi(\bar{z})$ is C_φ; the quantifier rank $qr(\psi)$ is at most k and the arity $|\bar{z}|$ is at most $arity(Q)$. Now fix an arbitrary tuple $\bar{t}$ over $D_{I'} \cup C_\varphi$. Let $\beta = \psi(\bar{t})$ and let $\bar{c}_\beta$ be the sequence of constants occurring in β (that is the sequence of all constants occurring either in C_φ or in $\bar{t}$). We view I and I' as first-order structures over vocabulary $\langle \tau, \bar{c}_\beta \rangle$ and with universes $D_I \cup C_\varphi$ and $D_{I'} \cup C_\varphi$, respectively. We prove that the duplicator has a winning strategy in the k-round Ehrenfeucht-Fraïssé game on I and I'. Since $qr(\beta) \leq k$, this will prove that I and I' agree on β.

Observe that the universe of I is $D_I \cup C_\varphi = C \cup \bigcup_{X \subseteq K} C_X$ and the universe of I' is $D_{I'} \cup C_\varphi = C \cup \bigcup_{X \subseteq K} C'_X$. Therefore, given an arbitrary value c in the universe of I or I', exactly one of the following holds:

- either $c \in C$;
- or $c \in C_X$, for some class C_X with $|C_X| \leq k + arity(Q)$;
- or $c \in C_X$, for some class C_X with $|C_X| > k + arity(Q)$.

The strategy of the duplicator is as follows. Assume that $i < k$ rounds have been played and assume that at round $i+1$ the spoiler picks a structure A (either I or I'). Assume that the sequence of moves played in structure A in former rounds is $(a_1, \ldots, a_i)$. Let B be the other structure (either I' or I), and $(b_1, \ldots, b_i)$ the values played on B in previous rounds. At round $i+1$, if the spoiler picks a value a_{i+1} in the universe of A such that $a_{i+1} = a_j$ for some $j \leq i$, then the duplicator responds with $b_{i+1} = b_j$. Otherwise, if a_{i+1} has never been played on A in former rounds:

- if either a_{i+1} occurs in $\bar{c}_\beta$, or $a_{i+1} \in C$, or $a_{i+1} \in C_X$, with $|C_X| \leq k + arity(Q)$, then the duplicator responds with $b_{i+1} = a_{i+1}$;
- otherwise (if $a_{i+1} \in C_X$, with $|C_X| > k + arity(Q)$, and a_{i+1} does not occur in $\bar{c}_\beta$), the duplicator responds with an arbitrary value $b_{i+1} \in C'_X$ which does not occur in $\bar{c}_\beta$ and has not been played on B yet. We are guaranteed that such a value exists because because cardinalities of the sets C'_X are sufficiently large. That is, there are at most $arity(Q)$ constants from $\bar{c}_\beta$ occurring in C'_X, and fewer than k rounds have been played, so with $|C'_X| = k + arity(Q)$ there is an element to choose from.

If $(q^I_1, \ldots q^I_k)$ and $(q^{I'}_1, \ldots, q^{I'}_k)$ are the sequences of values played after k rounds of the game, on I and I' respectively, we define the sequences $\bar{c}^I = (\bar{c}_\beta, q^I_1, \ldots, q^I_k)$ and $\bar{c}^{I'} = (\bar{c}_\beta, q^{I'}_1, \ldots, q^{I'}_k)$ of size $|\bar{c}_\beta| + k$.

We prove that if the duplicator adopts the above strategy, after k rounds of the game, the following holds:

Claim 3. *For each $l = 1, \ldots, |\bar{c}_\beta| + k$*

- *if the pair $(c^I_l, c^{I'}_l)$ contains a value occurring either in $\bar{c}_\beta$ or in C or in a class C_X, with $|C_X| \leq k + arity(Q)$, then $c^I_l = c^{I'}_l$;*
- *if the pair $(c^I_l, c^{I'}_l)$ contains a value of a class C_X, then both c^I_l and $c^{I'}_l$ are in C_X.*

Proof of Claim 3. We prove the statement by induction on l: it is trivially true for each $l \leq |\bar{c}_\beta|$; we now assume that the claim holds for each $l \leq i-1$ and prove it for $l = i$ (with $|\bar{c}_\beta| < i \leq |\bar{c}_\beta| + k$). Since $i > |\bar{c}_\beta|$, the pair $(c^I_i, c^{I'}_i)$ represents the values played at round $i - |\bar{c}_\beta|$. As usual we denote by A the structure picked by the spoiler in this round and B the other structure. There are two cases:

1. If $c^A_i = c^A_j$ for some $|\bar{c}_\beta| < j < i$ then, by the strategy of the duplicator, $c^B_i = c^B_j$. The statement for c^A_i and c^B_i follows by the induction hypothesis on c^A_j and c^B_j.
2. Otherwise, if c^A_i has never been played on A in previous rounds, then directly by the strategy of the duplicator:
   (a) if either c^A_i occurs in $\bar{c}_\beta$ or $c^A_i \in C$ or $c^A_i \in C_X$ with $|C_X| \leq k + arity(Q)$, then $c^B_i = c^A_i$;
   (b) if c^A_i belongs to some class C_X then both c^A_i and c^B_i are in C_X.

   It remains to prove 2a and 2b also for c^B_i. Assume that either c^B_i occurs in $\bar{c}_\beta$ or $c^B_i \in C$ or c^B_i is in a class of cardinality at most $k + arity(Q)$. Assume, by contradiction, that $c^B_i \neq c^A_i$. Then, by 2a, c^A_i is in a class C_X with $|C_X| > k + arity(Q)$ and c^A_i does not occur in $\bar{c}_\beta$. Therefore, by the strategy of the duplicator, $c^B_i \in C_X$ and c^B_i does not occur in $\bar{c}_\beta$. This contradicts the initial assumption on c^B_i, and proves 2a also for c^B_i.

   Now assume c^B_i belongs to some class C_X and, by contradiction, that $c^A_i \notin C_X$. Then either $c^A_i \in C$ or $c^A_i \in C_{X_0}$, for some $C_{X_0} \neq C_X$. In the first case, by 2a, $c^B_i = c^A_i$; thus $c^B_i \in C$. In the other case, by 2b, $c^B_i \in C_{X_0}$, thus $c^B_i \notin C_X$. In both cases we reach contradiction, thus 2b holds also for c^B_i.

This ends the proof of Claim 3. □

Claim 4. *The pair $\langle \bar{c}^I, \bar{c}^{I'} \rangle$ forms a partial isomorphism between I and I' (that is, the duplicator has a winning k-round strategy).*

Proof of Claim 4. We split the proof into two parts.

1. We prove that for each $j, l \in [1, \ldots, |\bar{c}_\beta| + k]$, we have $c_j^I = c_l^I$ if and only if $c_j^{I'} = c_l^{I'}$. This holds trivially if $j, l \leq |\bar{c}_\beta|$. Now assume that the statement holds for $j, l \in [1, \ldots, i-1]$ (with $|\bar{c}_\beta| < i \leq |\bar{c}_\beta| + k$), the we prove that it also holds for $j, l \in [1, \ldots, i]$. We only need to prove that for an arbitrary j in $[1, \ldots, i-1]$, $c_i^I = c_j^I$ if and only if $c_i^{I'} = c_j^{I'}$. There are two cases:
   (a) If c_i^A has already been played on A in some previous round, then there exists $|\bar{c}_\beta| < l < i$ such that $c_i^A = c_l^A$. In this case the choice of the duplicator is $c_i^B = c_l^B$. By the induction hypothesis, $c_l^A = c_j^A$ if and only if $c_l^B = c_j^B$; thus $c_i^A = c_j^A$ if and only if $c_i^B = c_j^B$.
   (b) Assume now that c_i^A has never been played on A in previous rounds. If $c_i^A = c_j^A$, then it must be the case that $j \leq |\bar{c}_\beta|$ and $c_j^A = c_j^B = c$, for some constant c in $\bar{c}_\beta$. Therefore the duplicator chooses $c_i^B = c$, then $c_i^B = c_j^B$.
   Conversely if $c_i^A \neq c_j^A$, there are two cases:
      - If either c_i^A occurs in $\bar{c}_\beta$ or $c_i^A \in C$ or $c_i^A \in C_X$, with $|C_X| \leq k + \mathit{arity}(Q)$, the duplicator chooses $c_i^B = c_i^A$. If by contradiction $c_i^B = c_j^B$, then $c_j^A \neq c_j^B$. Therefore by Claim 3, $c_j^B (= c_i^A)$ must be in a class of size greater than $k + \mathit{arity}(Q)$ and must not occur in $\bar{c}_\beta$. This contradicts the original hypothesis about c_i^A.
      - Otherwise, if $c_i^A \in C_X$ with $|C_X| > k + \mathit{arity}(Q)$ and c_i^A does not occur in $\bar{c}_\beta$ then, directly by the strategy of the duplicator, c_i^B does not occur in $\bar{c}_\beta$ and it has never been played before on B. Thus $c_i^B \neq c_j^B$.
2. Next we prove that, if $\bar{u} = (c_{i_1}^I, \ldots, c_{i_l}^I)$ with $i_1, \ldots, i_l \in [1, \ldots, |\bar{c}_\beta| + k]$, if $\bar{u}' = (c_{i_1}^{I'}, \ldots, c_{i_l}^{I'})$ and R is a relation symbol of τ, then $\bar{u} \in R^I$ if and only if $\bar{u}' \in R^{I'}$. Assume that $\bar{u} \in R^I$, then there are two cases: either $\bar{u}$ is a tuple of R in the sub-instance $V \cup E_0$, or $\bar{u} \in R^E$. In the first case, all constants in $\bar{u}$ are in C thus, by Claim 3, $\bar{u}' = \bar{u}$. Therefore $\bar{u}'$ is a tuple of R in the sub-instance $V \cup E_0 \subseteq I'$. In the case that $\bar{u} \in R^E$, it is of the form $\bar{u} = (\bar{k}_1, d, \bar{k}_2)$ with $\langle R; \bar{k}_1; \bar{k}_2 \rangle \in X(d)$ and $d \in C_{X(d)}$. By Claim 3, as $\bar{k}_1$ and $\bar{k}_2$ are tuples over C, we have that $\bar{u}' = (\bar{k}_1, d', \bar{k}_2)$ with $d' \in C_{X(d)}$ (that is $X(d') = X(d)$). Now, the value d' is of course in the universe of I', that is $D_{I'} \cup C_\varphi$. Moreover, since d' belongs to $C_{X(d)}$, we have $d' \notin C$. Therefore $d' \in D_{I'} - C$ (due to the fact that $C_\varphi \subseteq C$). This implies, by the definition of E', that $X(d') \times \{d'\} \subseteq E'$ and thus $(\bar{k}_1, d', \bar{k}_2) \in R^{E'}$, that is $\bar{u}' \in R^{E'}$.
   If we start with $\bar{u}' \in R^{I'}$, the proof that $\bar{u} \in R^I$ is symmetric.

This concludes the proof of Claim 4. □

Claim 4 proves that the duplicator has a winning strategy for the k-round game. Therefore, since $qr(\beta) \leq k$, I and I' agree on $\beta = \psi(\bar{t})$.

For an arbitrary tuple $\bar{t}$ over $D_{I'} \cup C_\varphi$, we proved that $I \models \psi(\bar{t})$ iff $I' \models \psi(\bar{t})$. The proof holds both for $\psi = \varphi(\bar{x}, \bar{y})$ (representing the query Q) and $\psi = \exists \bar{y} \varphi(\bar{x}, \bar{y})$ (representing the query $\pi_X Q$). Based on this result, we now show that $W \subseteq Q(I')$ and $\pi_X(W) = \pi_X(Q(I'))$.

By the construction of I', the domain D_W is included in $D_{I'} \cup C_\varphi$. Therefore, for each tuple $\bar{t} \in W$, since $I \models \varphi(\bar{t})$, also $I' \models \varphi(\bar{t})$. As a consequence $W \subseteq Q(I')$.

Furthermore each tuple $\bar{t} \in \pi_X(W)$ is a tuple over $D_{I'} \cup C_\varphi$ and is such that $I \models \exists \bar{y} \varphi(\bar{t}, \bar{y})$. Therefore also $I' \models \exists \bar{y} \varphi(\bar{t}, \bar{y})$ and thus $\bar{t} \in \pi_X(Q(I'))$. Conversely, if $\bar{t} \in \pi_X(Q(I'))$ then $\bar{t}$ is is a tuple over $D_{I'} \cup C_\varphi$ and $I' \models \exists \bar{y} \varphi(\bar{t}, \bar{y})$. Then also $I \models \exists \bar{y} \varphi(\bar{t}, \bar{y})$ and thus $\bar{t} \in \pi_X(Q(I'))$.

A NEXPTIME algorithm guesses I' by guessing:

1. a valuation v of $\mathit{re\ell}(T)$, from which the target instance V and the sets C and K can be computed as described above;
2. a target instance $E_0 \subseteq K \times C$;
3. disjoint sets $C'_X \subseteq \mathsf{Const} - C$ with $|C'_X| \leq k + \mathit{arity}(Q)$, for each $X \subseteq K$, giving $E' = \bigcup_{X \subseteq K} \bigcup_{d \in C'_X} X \times \{d\}$.

Then $I' = V \cup E_0 \cup E'$ is computed and $W \subsetneq Q(I)$ and $\pi_X(W) = \pi_X(Q(I))$ are checked. Given the exponential bound on $\|I'\|$ shown earlier, we have membership in NEXPTIME. This concludes the proof of Lemma 2. □

Hardness. We reduce an NEXPTIME-complete version of the TILING problem to the complement of DEQA(Σ_α, Q) for a particular first order query Q and a mapping Σ_α with $\#_{op}(\Sigma_\alpha) = 1$. We are given an input instance of the tiling problem, that is

- a set of tile types $T = \{t_0, \ldots, t_k\}$,
- horizontal and vertical compatibility relations among tiles $H, V \subseteq T \times T$
- an integer n in unary

The tiling problem is the problem of telling whether there exists a *tiling* of the $2^n \times 2^n$ grid, that is a mapping $f : \{0, \ldots, 2^n - 1\} \times \{0, \ldots, 2^n - 1\} \to T$ which associates a tile to each position of the grid, in such a way that horizontally consecutive tiles respect H, vertically consecutive tiles respect V, and $f(0, 0) = t_0$.

We fix the following annotated schema mapping:

- the source schema consists of binary relations H_s and V_s (intended to represent horizontal and vertical constraints), a unary relation N_s (representing n in unary), a unary relation T (the set of tiles), a unary relation Empty_s representing a constant for the empty set, and a binary relation $<_s$ (linear order over elements of N_s);
- the target schema consists of relations H, V, N, Empty and $<$, having the same arity and intended meaning of H_s, V_s, N_s, Empty_s and $<_s$ respectively; a binary relation F (the tiling function), whose intended semantics is to associate to each tile a subset of positions of the grid; two binary relations G_h and G_v, intended to represent horizontal and vertical coordinates of grid positions – as we will explain later in more detail.
- the STDs Σ_α are as follows:

$$\begin{array}{rcl}
H(x^{cl}, y^{cl}) & \text{:-} & H_s(x, y) \\
V(x^{cl}, y^{cl}) & \text{:-} & V_s(x, y) \\
N(x^{cl}) & \text{:-} & N_s(x) \\
G_h(x^{cl}, y^{op}) & \text{:-} & N_s(x) \\
G_v(x^{cl}, y^{op}) & \text{:-} & N_s(x) \\
F(x^{cl}, y^{op}) & \text{:-} & T(x) \\
\mathit{Empty}(x^{cl}) & \text{:-} & \mathit{Empty}_s(x) \\
x^{cl} < y^{cl} & \text{:-} & x <_s y
\end{array}$$

The input instance for the tiling problem can be translated directly into a source instance S for Σ_α by interpreting: H_s and V_s as H and V respectively, N_s as the unary relation $\{1, \ldots, n\}$, T as $\{t_0, \cdots, t_k\}$, Empty_s as the singleton containing only the symbol '$\emptyset$', and $<_s$ as $\{(i, j) | 1 \leq i < j \leq n\}$.

In $\text{CSol}_{\text{A}}(S)$, relations H, V, N, Empty and $<$ are copies of H_s, V_s, N_s, Empty_s and $<_s$, annotated as *closed* on each position, while G_h consists of tuples $\{(i^{cl}, \bot^{op}_{hi})|1 \le i \le n\}$, G_v is $\{(i^{cl}, \bot^{op}_{vi})|1 \le i \le n\}$, and F is interpreted as $\{(t^{cl}_i, \bot^{op}_i)|0 \le i \le k\}$.

In the rest of the proof we construct a query Q, represented by a FO formula $\varphi(\bar{x})$ over the target schema, and an input tuple $\bar{t}$ such that there exists a tiling if and only if $\bar{t} \notin \mathsf{certain}_{\Sigma_\alpha}(Q, S)$ – that is if and only if there exists an instance $I \in \mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$ such that $I \models \neg\varphi(\bar{t})$. This will complete the reduction.

We first construct a sentence β over the target schema such that there exists a tiling if and only if there exists an instance $I \in \mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$ satisfying β. Then we let $\varphi(x) = \neg(\beta \wedge \mathit{Empty}(x))$ and $\bar{t} =$ '$\emptyset$'. As all instances in $\mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$ satisfy $\mathit{Empty}(\emptyset)$, there exists $I \in \mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$ such that $I \models \neg\varphi(\emptyset)$ if and only if there exists a tiling.

We now write β as a sentence – to be interpreted over instances in $\mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$ – which forces the intended semantics of relations F, G_h and G_v, and forces F to represent a tiling of a $2^n \times 2^n$ grid. This intended semantics is as follows. The idea is that open nulls introduced in rules as second attributes of relations define subsets of values of the first attributes, that is, $\{1, \dots, n\}$. Thus, they are in one-to-one correspondence with the values of the axes of the grid. The relations G_h and G_v code pairs of coordinates, and the relation F gives the assignment of tiles. The sentence checks if the coding is correct, and the assignment is indeed a tiling.

More precisely, in what follows, for a given instance in $\mathit{Rep}_A(\text{CSol}_{\text{A}}(S))$, for each value c of its active domain, we let $X_h(c)$ be the set of all $i \in \{1, \dots, n\}$ such that $(i, c) \in G_h$. If we interpret this subset as the set of 1-positions in a vector of n bits, $X_h(c)$ represents an integer between 0 to $2^n - 1$. Similarly we define $X_v(c)$. Then each domain value c represents the pair of integers $(X_h(c), X_v(c))$, that is, some grid position. We shall use the notation $R.x$ for $\{y \mid (x, y) \in R\}$ for a binary relation R. Observe that each value c which does not occur in $G_h.y \cup G_v.y$ represents position $(0, 0)$.

β is the conjunction of the following sentences:

- a sentence β_1 checking that F associates to each tile either only the value $\emptyset$ or a set of values different from $\emptyset$:

$$\beta_1 = \neg\exists t, y_1, y_2(F(t, y_1) \wedge F(t, y_2) \wedge \mathit{Empty}(y_1) \wedge \neg\mathit{Empty}(y_2))$$

- a sentence β_2 checking that F represents a function mapping each distinct value of $F.y$ different from $\emptyset$ to exactly one tile.

$$\beta_2 = \forall x, t, t' \left(\neg\mathit{Empty}(x) \wedge F(t, x) \wedge F(t', x) \rightarrow t = t'\right)$$

- a sentence β_3 forcing a one-to-one mapping between distinct values of $F.y - \{\emptyset\}$ and grid positions $\{0, \dots, 2^n - 1\} \times \{0, \dots, 2^n - 1\}$. In particular β_3 checks that all grid positions are represented by exactly one value of $F.y - \{\emptyset\}$, and is given by $\beta_3 = \beta_{31} \wedge \beta_{32}$, where:
  - β_{31} checks that position $(2^n - 1, 2^n - 1)$ is represented by exactly one value of $F.y - \{\emptyset\}$,
  - β_{32} checks that, if position (i, j) is represented then: if $i > 0$, then also position $(i - 1, j)$ is represented by exactly one value of $F.y - \{\emptyset\}$, and if $j > 0$, then also position $(i, j - 1)$ is represented by exactly one value of $F.y - \{\emptyset\}$.

Let $\mathit{Pos}(y) = \neg\mathit{Empty}(y) \wedge \exists t F(t, y)$, then:

$$\beta_{31} = \exists! y\, (\mathit{Pos}(y) \wedge \forall i(N(i) \rightarrow G_h(i, y) \wedge G_v(i, y)))$$

$$\beta_{32} = \forall y\,(Pos(y) \to (Pred_h(y) \wedge Pred_v(y)))$$

where for $a = h, v$:

$$Pred_a(y) = (\exists i G_a(i,y)) \to (\exists! z(Pos(z) \wedge a\text{-}succ(z,y))$$

and $h\text{-}succ(z,y)$ and $v\text{-}succ(z,y)$ check that y represents a position of the grid which is the successor, in horizontal and vertical direction respectively, of the position represented by z. For $a = h$, $\bar{a} = v$ or $a = v$, $\bar{a} = h$, $a\text{-}succ(z,y)$ first checks that $X_{\bar{a}}(y) = X_{\bar{a}}(z)$ and then compares $X_a(y)$ and $X_a(z)$ viewed as bit vectors, to verify that one is the successor of the other in the usual way:

$$\begin{array}{rlll} a\text{-}succ(z,y) = & \forall i & (& G_{\bar{a}}(i,z) \leftrightarrow G_{\bar{a}}(i,y) \quad) \;\wedge \\ & \exists i & (& G_a(i,y) \wedge \neg G_a(i,z) \wedge \\ & & & \forall j(j < i \to (G_a(j,z) \wedge \neg G_a(j,y))) \wedge \\ & & & \forall j(i < j \to (G_a(j,z) \leftrightarrow G_a(j,y))) \quad) \end{array}$$

- a sentence β_4 requiring that F is a tiling. We write β_4 as $\beta_{41} \wedge \beta_{42}$, where β_{41} checks that F associates tile t_0 to position $(0,0)$, and β_{42} verifies horizontal and vertical constraints.

$$\beta_{41} = \exists y(\;\; F(\text{`}t_0\text{'}, y) \wedge \neg Empty(y) \wedge \neg\exists i(G_h(i,y) \vee G_v(i,y)) \;\;)$$

$$\begin{array}{rl} \beta_{42} = \forall x,y,t,t'(& (F(t,x) \wedge F(t',y) \wedge \neg Empty(x) \wedge \neg Empty(y)) \to \\ & (\; (h\text{-}succ(x,y) \to H(t,t')) \wedge \\ & \;\; (v\text{-}succ(x,y) \to V(t,t')) \;) \;\;) \end{array}$$

It is straightforward to verify that there exists an instance in $Rep_A(\text{CSol}_{\text{A}}(S))$ satisfying β if and only if there exists a tiling. This concludes the proof of coNEXPTIME-hardness and thus the proof of Theorem 3. □

We now look at some special cases when we can guarantee better complexity of query answering. The hardness results for $\#_{op}(\Sigma_\alpha) = 1$ are achieved in simple mappings with all STDs either copying, i.e. $R'(\bar{x}^{cl})$:– $R(\bar{x})$, or the simplest open null introductions $U'(x^{cl}, z^{op})$:– $U(x)$. Combining several relations into one, we can also see that hardness is witnessed by a two-rule mapping of the form $R'_1(\bar{x}^{cl})$:– $R_1(\bar{x})$, $R'_2(\bar{x}^{cl}, z^{op})$:– $R_2(\bar{x})$. Thus, to achieve better complexity we should look at subclasses of queries rather than mappings.

We start with positive relational algebra queries. From Proposition 3, we obtain

Corollary 3. *If Q is a positive relational algebra query, then* DEQA(Σ_α, Q) *is in* PTIME.

But adding inequalities even to conjunctive queries takes us to a larger class. Combining results of [24, 21] with properties of annotated solutions, we derive:

Proposition 4. *Let Σ be a set of STDs, α an arbitrary annotation, and Q a monotone polynomial-time query. Then* DEQA(Σ_α, Q) *is in* coNP. *Moreover, there exists a set Σ of STDs and a conjunctive query with two inequalities Q so that* DEQA(Σ_α, Q) *is* coNP*-complete for every annotation α.*

Proof. Membership. First observe that the proof of Proposition 3 is based only on the assumption of monotonicity of Q. Therefore, for an arbitrary monotone query Q,

$$\mathsf{certain}_{\Sigma_\alpha}(Q,S) \;=\; \Box Q(\mathrm{CS{\scriptsize OL}}(S)) \;=\; \mathsf{certain}_{\Sigma}^{\mathrm{CWA}}(Q,S).$$

As a consequence the complexity of DEQA(Σ_α, Q) coincides with the complexity of checking $\bar{t} \in \mathsf{certain}_{\Sigma}^{\mathrm{CWA}}(Q,S)$ which was shown to be in coNP for each mapping Σ and arbitrary query Q with polynomial data complexity [21].

Hardness. It was shown in [24] that there exists a set of dependencies Σ, forming a *LAV setting* (cf. [11]) and a boolean conjunctive query with two inequalities Q, such that checking $\mathsf{certain}_{\Sigma}^{\mathrm{OWA}}(Q,S) = \mathit{true}$, for an input S, is coNP-hard.

Now notice that, by Proposition 2, $\mathsf{certain}_{\Sigma}^{\mathrm{OWA}}(Q,S) = \mathsf{certain}_{\Sigma_{op}}(Q,S)$ and, since Q is monotone,

$$\mathsf{certain}_{\Sigma_{op}}(Q,S) = \mathsf{certain}_{\Sigma}^{\mathrm{CWA}}(Q,S) = \mathsf{certain}_{\Sigma_\alpha}(Q,S)$$

for an arbitrary α. Since LAV settings are special cases of schema mappings considered here, this proves that for every α the problem DEQA(Σ_α, Q) is coNP-hard and completes the proof of Proposition 4. □

Finally, we describe the complexity of universal or $\forall^*\exists^*$ queries. It can also be viewed as the complexity of validating constraints in data exchange, since most commonly used integrity constraints, equality- or tuple-generating, are expressed as $\forall^*$ or $\forall^*\exists^*$ sentences.

Proposition 5. *If Q is a $\forall^*\exists^*$ query, and Σ_α is an arbitrary annotated set of STDs, then* DEQA(Σ_α, Q) *is in* coNP.

Proof. Let Q be expressed by a $\forall^*\exists^*$ formula $\varphi(\bar{x})$. Given an input consisting of a source instance S for Σ_α and a constant tuple $\bar{t}$, we next prove that it is in NP to check whether $\bar{t} \notin \mathsf{certain}_{\Sigma_\alpha}(Q,S)$, that is to check whether there exists a target instance $I \in \mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$ such that $I \models \neg\varphi(\bar{t})$. In the rest of the proof β will denote the sentence $\neg\varphi(\bar{t})$; in particular we let:

$$\beta = \exists x_1, \ldots, x_l \forall y_1, \ldots, y_m \gamma(x_1, \ldots, x_l, y_1, \ldots, y_m)$$

where γ is a boolean combination of atomic formulae. Moreover we let C_β be the set of constants occurring in β and, for any target instance J over Const, we let U_J stand for $D_J \cup C_\beta$.

Assume that $I \in \mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$ and $I \models \beta$. Next we show that there exists an instance $I' \in \mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$ of size polynomial in $\|S\|$ such that $I' \models \beta$. Let v be a valuation witnessing $I \in \mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$, and $V = v(\mathit{re\ell}(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S)))$. As $I \models \beta$, there exists a tuple $\bar{k} = (k_1, \ldots, k_l)$ over U_I such that, for all tuples $(c_1, \ldots, c_m) \in U_I^m$, the formula $\gamma(k_1, \ldots, k_l, c_1, \ldots, c_m)$ holds in I. Observe that $\gamma(k_1, \ldots, k_l, c_1, \ldots, c_m)$ is a boolean combination of atoms with terms over the set $D_{\bar{k},\bar{c}} = \{k_1, \ldots, k_l, c_1, \ldots, c_m\} \cup C_\beta$. Moreover, as $\{k_1, \ldots, k_l\} \subseteq U_I$, there exists an instance $I_0 \subseteq I$, with at most l tuples, such that $D_{I_0} \cup C_\beta \supseteq \{k_1, \ldots, k_l\}$.

Now let I' be the target instance obtained by restricting I to the domain $U_V \cup D_{I_0}$. As $U_{I'} \subseteq U_I$, for each tuple $\bar{c} = (c_1, \ldots, c_m)$ over $U_{I'}$, the formula $\gamma(k_1, \ldots, k_l, c_1, \ldots, c_m)$ holds in I. Since $\gamma(k_1, \ldots, k_l, c_1, \ldots, c_m)$ is a boolean combination of atoms over $D_{\bar{k},\bar{c}}$ and $D_{\bar{k},\bar{c}} \subseteq U_V \cup D_{I_0}$, we see that $\gamma(k_1, \ldots, k_l, c_1, \ldots, c_m)$ also holds in I'. Furthermore $k_1, \ldots, k_l$ are values occurring in $U_{I'}$, thus $I' \models \beta$.

The instance I' is still in $\mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$ (via the valuation v), as it has been obtained from I by removing no tuple of $V = v(\mathit{re\ell}(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S)))$ (since it includes all tuples over the domain U_V), and I' is polynomial in $\|\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S)\|$ (thus in $\|S\|$), since the domain $U_V \cup D_{I_0}$ has linear size. In particular D_{I_0} is of fixed size $|D_{I_0}| \leq l \times \mathit{arity}(\tau)$, where $\mathit{arity}(\tau)$ is the maximum arity of relations of τ.

An NP algorithm can check whether there exists an instance in $\mathit{Rep}_A(\mathrm{CS{\scriptsize OL}}_{\mathrm{A}}(S))$ satisfying β by guessing:

- a valuation v on nulls of $\text{CSol}_{\text{A}}(S)$, generating $V = v(\mathit{re\ell}(\text{CSol}_{\text{A}}(S)))$;
- a set D_{I_0} of at most $l \times \mathit{arity}(\tau)$ constants;
- a target instance E as follows: for each relation symbol R of τ and each annotated tuple $(\bar{t}, \alpha) \in R^{\text{CSol}_{\text{A}}(S)}$, a set of tuples is guessed such that for each tuple $\bar{t}_0$ in the set:
  - $\bar{t}_0$ and $v(\bar{t})$ coincide on *closed* positions of α and
  - for each open attribute A of R annotated as *open* by α, $t_0(A) \in U_V \cup D_{I_0}$.

Finally the algorithm checks that $I' = V \cup E$ satisfies β. This proves the NP bound for $\bar{t} \notin \mathsf{certain}_{\Sigma_\alpha}(Q, S)$ and hence coNP membership for $\bar{t} \in \mathsf{certain}_{\Sigma_\alpha}(Q, S)$, and completes the proof of Proposition 5. □

5. Composing mappings

Composition and incomplete information

We now move to handling schema mappings themselves, and see how they behave under open, closed, or mixed open/closed annotations. We shall look in particular at composition of schema mappings, which is a key operation in schema evolution and model management in general [6, 7, 13, 27, 23].

We shall be dealing with schema mappings used in data exchange, i.e. triples (σ, τ, Σ). Since semantically a mapping is a binary relation consisting of pairs (S, T), where S and T are source and target instances satisfying Σ, [13] made a very natural proposal to use the composition of such relations to define the composition of mappings. One more condition, however, is required. Note that a pair (S, W) is in the composition of the binary relations given by the mappings (σ, τ, Σ) and (τ, ω, Δ) iff there is an instance T such that T is a solution for S (under Σ) and W is a solution for T (under Δ). But while solutions as defined in [11, 21] and here are instances over $\mathsf{Const} \cup \mathsf{Null}$, we do not have a definition of a solution for a source instance with nulls. Indeed, doing so would require evaluating universal constraints over instances with nulls, something that has long been known to be problematic [5, 17, 20].

So the definition of composition of [13] and others is restricted to instances over Const: a pair (S, W) of instances over Const belongs to the composition iff there is a solution T for S over Const, and W is a solution for T. But recall that, under the definition of a solution of [11], T that only uses elements of Const is a solution for S iff $T \in \mathit{Rep}(T')$ for some OWA-solution T' – equivalently, iff $T \in [\![S]\!]^{\Sigma}_{\text{OWA}}$. Thus, the semantics of a schema mapping used in [13] is

$$(\!|\Sigma|\!)^{\text{OWA}}_{\sigma,\tau} \;=\; \{(S, T) \;\mid\; T \in [\![S]\!]^{\Sigma}_{\text{OWA}}\},$$

where S and T range over σ- and τ-instances. Hence, their notion of composition is $(\!|\Sigma|\!)^{\text{OWA}}_{\sigma,\tau} \circ (\!|\Delta|\!)^{\text{OWA}}_{\tau,\omega}$. (We use slightly different brackets to denote the semantics of a schema mapping as opposed to the semantics of solutions.) So it is natural to ask how schema mappings and their compositions behave in settings where different assumptions about the semantics of incompleteness are made. In general, for an *annotated* mapping $(\sigma, \tau, \Sigma_\alpha)$, we define its semantics as

$$(\!|\Sigma_\alpha|\!)_{\sigma,\tau} \;=\; \{(S, T) \;\mid\; T \in [\![S]\!]^{\Sigma_\alpha}\},$$

and the composition of two annotated mappings $(\sigma, \tau, \Sigma_\alpha)$ and $(\tau, \omega, \Delta_{\alpha'})$ as

$$\Sigma_\alpha \circ \Delta_{\alpha'} \;\stackrel{\text{def}}{=}\; (\!|\Sigma_\alpha|\!)_{\sigma,\tau} \circ (\!|\Delta_{\alpha'}|\!)_{\tau,\omega}.$$

We omit the schemas from our notation $\Sigma_\alpha \circ \Delta_{\alpha'}$ as they will always be clear from the context.

Notice that if both α and α' are all-open annotations, then this is the definition of composition of [13].

We now deal with two basic problems related to schema mappings and their composition: their complexity, and syntactic representations (i.e., what is a class of constraints that captures $\Sigma_\alpha \circ \Delta_{\alpha'}$?).

For several results, the class of queries used as source formulae φ_σ in STDs will be important. In [12, 11, 13] only conjunctive queries are allowed in STDs; so far, as in [3, 21], we allowed arbitrary FO queries. We refer to STDs $\psi_\tau(\bar{x}, \bar{z}) \text{ :- } \varphi_\sigma(\bar{x}, \bar{y})$ as *CQ-STDs* or *monotone STDs* if φ_σ is a conjunctive (resp., monotone) query. Otherwise it is assumed to be an FO query, as before.

Complexity of composition

Let $(\sigma, \tau, \Sigma_\alpha)$ and $(\tau, \omega, \Delta_{\alpha'})$ be two annotated mappings. We consider the following *composition problem*:

PROBLEM	$\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$
INPUT:	a σ-database S, a ω-database W
QUESTION:	is (S, W) in $\Sigma_\alpha \circ \Delta_{\alpha'}$?

The all-open version $\textsc{Comp}(\Sigma_{op}, \Delta_{op})$ with CQ-STDs was shown to be NP-complete in [13]. We first extend this to more general annotations.

Lemma 3. *If Δ contains only monotone STDs, Σ is arbitrary, and α is any annotation of Σ, then*

$$\Sigma_\alpha \ \circ \ \Delta_{op} \ = \ \Sigma_{op} \ \circ \ \Delta_{op}.$$

Proof. Fix arbitrary instances S of schema σ and T of schema ω. Assume first that $(S, T) \in \Sigma_\alpha \circ \Delta_{op}$. Then there exists $J \in \llbracket S \rrbracket^{\Sigma_\alpha}$ such that $T \in \llbracket J \rrbracket^{\Delta_{op}}$. By Theorem 1, $\llbracket S \rrbracket^{\Sigma_\alpha} \subseteq \llbracket S \rrbracket^{\Sigma_{op}}$, and thus J is also in $\llbracket S \rrbracket^{\Sigma_{op}}$, implying that $(S, T) \in \Sigma_{op} \circ \Delta_{op}$.

Conversely assume $(S, T) \in \Sigma_{op} \circ \Delta_{op}$. Then there exists $J \in \llbracket S \rrbracket^{\Sigma_{op}}$ such that $T \in \llbracket J \rrbracket^{\Delta_{op}}$. Since J is in $\llbracket S \rrbracket^{\Sigma_{op}}$, it contains an instance $J_0 = v(\mathit{re\ell}(\textsc{CSol}_{\text{A}}^{\Sigma_{op}}(S)))$ for some valuation v. Moreover, since STDs in Δ are monotone and $(J, T) \models \Delta$, also $(J_0, T) \models \Delta$, that is $T \in \llbracket J_0 \rrbracket^{\Delta}_{\text{OWA}}$. By Theorem 1, $T \in \llbracket J_0 \rrbracket^{\Delta_{op}}$. Now note that J_0 is also in $\llbracket S \rrbracket^{\Sigma_\alpha}$: since

$$\mathit{re\ell}(\textsc{CSol}_{\text{A}}^{\Sigma_{op}}(S)) \ = \ \textsc{CSol}^{\Sigma}(S) \ = \ \mathit{re\ell}(\textsc{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$$

we see that $J_0 \in \mathit{Rep}_A(\textsc{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$. As a consequence, $T \in \Sigma_\alpha \circ \Delta_{op}$. □

Corollary 4. *If Δ contains only monotone STDs, then $\textsc{Comp}(\Sigma_\alpha, \Delta_{op})$ is in* NP *for every Σ_α. Moreover, there exist mappings with CQ-STDs Σ and Δ so that for an arbitrary annotation α of Σ, the problem $\textsc{Comp}(\Sigma_\alpha, \Delta_{op})$ is* NP*-complete.*

We now look at arbitrary FO-STDs. The complexity of the composition problem, as the complexity of query answering, is classified by the parameter $\#_{op}(\Sigma_\alpha)$, the maximum number of open positions per atom in an STD in Σ_α. The classification is another trichotomy. The complexity of $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ is:

- NP-complete if $\#_{op}(\Sigma_\alpha) = 0$;
- NEXPTIME-complete if $\#_{op}(\Sigma_\alpha) = 1$;
- undecidable if $\#_{op}(\Sigma_\alpha) > 1$.

Theorem 4. *If (σ, τ, Σ) and (τ, ω, Δ) are two schema mappings with arbitrary* FO*-STDs, and α and α' are annotations of Σ and Δ, then:*

- *If $\#_{op}(\Sigma_\alpha) = 0$ (i.e., α is the all-closed annotation), then* $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ *is in* NP. *Moreover, there exist Σ and Δ, using only CQ-STDs, so that* $\textsc{Comp}(\Sigma_{cl}, \Delta_{\alpha'})$ *is* NP*-hard for every annotation α'.*
- *If $\#_{op}(\Sigma_\alpha) = 1$, then* $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ *is in* NEXPTIME. *Moreover, there exist Σ_α with $\#_{op}(\Sigma_\alpha) = 1$ and Δ so that* $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ *is* NEXPTIME*-hard for every annotation α'.*
- *For each $k > 1$, there exist Σ_α with $\#_{op}(\Sigma_\alpha) = k$ and Δ so that* $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ *is undecidable for every annotation α'.*

Proof. Membership for $\#_{op}(\Sigma_\alpha) = 0$.

For arbitrary instances S and T of schemas σ and ω, respectively, the canonical solution $\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ can be computed in time polynomial in $\|S\|$. Then a solution J in $\mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S))$ can be guessed by simply guessing a valuation of nulls of $\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$. Finally $T \in [\![J]\!]^{\Delta_{\alpha'}}$ can be checked in non-deterministic polynomial time by Theorem 2.

Membership for $\#_{op}(\Sigma_\alpha) = 1$.

Let $\Delta = \{\psi_i(\bar{x}_i, \bar{z}_i) \coloneq \varphi_i(\bar{x}_i, \bar{y}_i) \mid i = 1, \ldots, k\}$, and let Q_i be the query over schema τ expressed by the first-order formula $\varphi_i(\bar{x}_i, \bar{y}_i)$. Assume, without loss of generality, that variables in $\bar{x}_i, \bar{y}_i$ are disjoint from variables in $\bar{x}_j, \bar{y}_j$, for $i \neq j$. Also let $R_1, \ldots, R_k$ be new distinct relation symbols, not occurring in $\sigma \cup \tau \cup \omega$, such that R_i has arity $|\bar{x}_i| + |\bar{y}_i|$. Define a set of STDs

$$\Gamma \;=\; \{\psi_i(\bar{x}_i, \bar{z}_i) \coloneq R_i(\bar{x}_i, \bar{y}_i) \mid i = 1, \ldots, k\},$$

and let $\Gamma_{\alpha'}$ the annotation of Γ with α'.

Given an input (S, T) for the problem $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$, the following holds:

Claim 5. *A pair of instances (S, T) belongs to the composition $\Sigma_\alpha \circ \Delta_{\alpha'}$ if and only if there exists an instance K of schema $\{R_1, \ldots, R_k\}$ and an instance $J \in [\![S]\!]^{\Sigma_\alpha}$ such that:*

1. *$\|K\| = O(\|T\|^c)$, for a constant c that depends only on the mappings Σ and Δ;*
2. *$T \in \mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K))$;*
3. *$R_i^K \subseteq Q_i(J)$, for $i = 1, \ldots, k$;*
4. *$\pi_{\bar{x}_i}({R_i}^K) = \pi_{\bar{x}_i}(Q_i(J))$, for $i = 1, \ldots, k$.*

Note that by $\pi_{\bar{x}_i}$ we mean projection on attributes corresponding to the variables in $\bar{x}_i$.

Proof of Claim 5. If $(S, T) \in \Sigma_\alpha \circ \Delta_{\alpha'}$, then there exists an instance J such that $(S, J) \in (\!|\Sigma_\alpha|\!)$ and $(J, T) \in (\!|\Delta_{\alpha'}|\!)$. Let v be a valuation that witnesses $T \in \mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$. That is, v is a valuation of nulls of $\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$ so that T is in $\mathit{Rep}_A(v(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))$.

We define K as an instance of schema $\{R_1, \ldots, R_k\}$ obtained as follows:

- for each non-empty tuple $(\bar{t}_0, \alpha_0)$ in a relation R of $v(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$:
  - choose arbitrarily a tuple $(\bar{t}, \alpha_0)$ in relation R of $\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$ such that $\bar{t}_0 = v(\bar{t})$;
  - if $(\varphi_j, \psi_j, \bar{a}, \bar{b})$ is the justification for nulls of $\bar{t}$, put $(\bar{a}, \bar{b})$ in relation R_j^K (note that when $(\bar{a}, \bar{b})$ is put in R_j^K, we have $(\bar{a}, \bar{b}) \in Q_j(J)$);
- for each $i \in \{1, \ldots, k\}$ and for each tuple $\bar{a} \in \pi_{\bar{x}_i}(Q_i(J))$, add exactly one tuple $(\bar{a}, \bar{b})$ from $Q_i(J)$ to R_i^K.

Clearly $\|K\| \leq \|v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))\| + \|\pi_{\bar{x}_i}(Q_i(J))\|$. Observe that, for each i, the size of $\pi_{\bar{x}_i}(Q_i(J))$ is polynomial in $\|T\|$. In fact the active domain of $\pi_{\bar{x}_i}(Q_i(J))$ is contained in the active domain of $\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$, and thus in the active domain of T.

Moreover also $\|v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))\|$ is polynomial in $\|T\|$. Indeed $T \supseteq \mathit{rel}(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))$ therefore $\|\mathit{rel}(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))\| \leq \|T\|$. Although $v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$ may contain empty tuples and may replicate tuples of $\mathit{rel}(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))$ with different annotations, the number of empty tuples is bounded by the size of $\Delta_{\alpha'}$, and for each tuple in $\mathit{rel}(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))$ the number of its replications with different annotations is also bounded by the size of $\Delta_{\alpha'}$. Therefore $\|v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))\| \leq c_1 \cdot \|rel(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)))\| + c_2$ for some constants c_1 and c_2 that depend on $\Delta_{\alpha'}$. Then $\|v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))\| \leq c_1 \cdot \|T\| + c_2$.

We can conclude that the size of K is polynomial in $\|T\|$, with a polynomial dependent only on the mappings; this proves 1. Furthermore, 3 and 4 hold directly by construction of K.

We next show 2. We define a valuation v' on $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)$ and show that that T is in $\mathit{Rep}_A(v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)))$. We let E_Γ and E_Δ be the sub-instances containing all the empty tuples of $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)$ and $\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$, respectively. Notice that, since $R_i^K \subseteq Q_i(J)$ and left-hand sides of $\Delta_{\alpha'}$ and $\Gamma_{\alpha'}$ are the same, the following hold:

- $E_\Delta \subseteq E_\Gamma$. Indeed if an empty tuple $(\_, \alpha_0)$ is in some relation R of E_Δ, then there exists an STD $\psi_i :\!- \varphi_i$ in $\Delta_{\alpha'}$ such that $Q_i(J)$ is the empty set and ψ_i contains an atom $R(\bar{t})$ with annotation α_0. Therefore also R_i^K is the empty relation, and thus $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)$ contains $(\_, \alpha_0)$ in relation R.

- With a similar argument it can be easily verified that the restriction of $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)$ to non-empty tuples (that is, the instance $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K) - E_\Gamma$), is isomorphic to a sub-instance of $\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$: precisely the sub-instance $C_{\mathcal{J}}$ generated by the set of justifications $\mathcal{J} = \{(\varphi_i, \psi_i, \bar{a}, \bar{b}) \mid (\bar{a}, \bar{b}) \in R_i^K, i = 1, \ldots, k\}$ (where, for each relation symbol R, the sub-instance generated by the justification $(\varphi_i, \psi_i, \bar{a}, \bar{b})$ is precisely the set of annotated tuples occurring in the atoms of R contained in $\psi_i(\bar{a}, \bot_{(\varphi_i, \psi_i, \bar{a}, \bar{b})})$). This isomorphism simply maps nulls given by justifications $(\varphi_i, \psi_i, \bar{a}, \bar{b})$ to nulls given by justifications $(R_i, \psi_i, \bar{a}, \bar{b})$.

Now, given nulls $\bot$ in $C_{\mathcal{J}}$ and $\bot'$ in $\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K) - E_\Gamma$ mapped into each other by the isomorphism we just established, we define $v'(\bot')$ as $v(\bot)$, so that $v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K) - E_\Gamma) = v(C_{\mathcal{J}})$. We now prove that $v(C_{\mathcal{J}}) = v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta)$.

Since $C_{\mathcal{J}} \subseteq \mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta$, we have that $v(C_{\mathcal{J}}) \subseteq v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta)$. Moreover, by the construction of K, for each non-empty tuple $(\bar{t}_0, \alpha_0)$ in some relation R of $v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$, there exists a justification $(\varphi_j, \psi_j, \bar{a}, \bar{b}) \in \mathcal{J}$ and a tuple $(\bar{t}, \alpha_0)$ in the instance of R generated by $(\varphi_j, \psi_j, \bar{a}, \bar{b})$, such that $v(\bar{t}) = \bar{t}_0$. Since $(\bar{t}, \alpha_0)$ is generated by a justification of $\mathcal{J}$, then $(\bar{t}, \alpha_0)$ is a tuple of R in $C_{\mathcal{J}}$, therefore $v(C_{\mathcal{J}}) \supseteq v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta)$. This proves the reverse inclusion, thus $v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta) = v(C_{\mathcal{J}}) = v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K) - E_\Gamma)$. As a consequence

$$\mathit{rel}(v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))) = \mathit{rel}(v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K))) \tag{3}$$

Furthermore since $E_\Delta \subseteq E_\Gamma$, we have

$$v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)) \;=\; v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J) - E_\Delta) \cup E_\Delta \;\subseteq\; v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K) - E_\Gamma) \cup E_\Gamma \;=\; v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K))$$

and thus

$$v(\mathrm{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)) \subseteq v'(\mathrm{CSol}_{\mathrm{A}}^{\Gamma_{\alpha'}}(K)). \tag{4}$$

Now recall that v is the valuation witnessing $T \in \mathit{Rep}_A(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J))$. Therefore T is in $\mathit{Rep}_A(v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)))$. Then T contains $\mathit{re\ell}(v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)))$ and thus, by (3), the instance T contains $\mathit{re\ell}(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)))$. Moreover each tuple $\bar{t}$ in a relation R of T coincides with some tuple of $v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J))$ on closed positions; then, by (4), $\bar{t}$ coincides with some tuple of $v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K))$. Therefore T is in $\mathit{Rep}_A(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)))$, and then $T \in \mathit{Rep}_A(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K))$. This proves item 2 and completes the proof of one direction of the claim.

Conversely, assume that there exists an instance K of schema $\{R_1, \ldots, R_k\}$ and an instance $J \in [\![S]\!]^{\Sigma_\alpha}$ satisfying 1, 2, 3 and 4 of the claim. We prove that $(J, T) \in (\!|\Delta_{\alpha'}|\!)$, implying that $(S, T) \in (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$. From 2 we know that there exists a valuation v' of $\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)$ such that T is in $\mathit{Rep}_A(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)))$. We now define a valuation v on nulls of $\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)$ such that T is in $\mathit{Rep}_A(v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)))$.

Again we let E_Γ and E_Δ the sub-instances containing all the empty tuples of $\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)$ and $\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)$, respectively. Moreover, if we let $C_{(\varphi,\psi,\bar{a},\bar{b})}$ denote the instance generated by the justification $(\varphi, \psi, \bar{a}, \bar{b})$, then $\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)$ can be partitioned into the following disjoint sub-instances:

i. the sub-instance E_Δ containing all the empty tuples;

ii. the sub-instance $C_{\mathcal{J}}$ generated by the set of justifications $\mathcal{J} = \{(\varphi_i, \psi_i, \bar{a}, \bar{b}) \mid (\bar{a}, \bar{b}) \in R_i^K, i = 1, \ldots, k\}$;

iii. sub-instances $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$, for all justifications $(\varphi_i, \psi_i, \bar{a}, \bar{b})$ of nulls of $\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)$ such that $(\varphi_i, \psi_i, \bar{a}, \bar{b}) \notin \mathcal{J}$.

Notice that the valuation v can be defined independently on each of these sub-instances partitioning $\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)$, since their sets of nulls are pairwise disjoint.

Definition of v on $C_{\mathcal{J}}$. As observed already, by 3 and the fact that left-hand sides of $\Delta_{\alpha'}$ and $\Gamma_{\alpha'}$ coincide, $\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K) - E_\Gamma$ is isomorphic to $C_{\mathcal{J}}$. So for each pair of nulls $\perp'$ in $\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K) - E_\Gamma$, and $\perp$ in $C_{\mathcal{J}}$ related by this isomorphism, we define $v(\perp)$ as $v'(\perp')$. Hence $v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K) - E_\Gamma) = v(C_{\mathcal{J}})$, then

$$v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)) = v(C_{\mathcal{J}}) \cup E_\Gamma \tag{5}$$

Definition of v on instances $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$. For each such sub-instance $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$ defined as in iii, we know that $(\bar{a}, \bar{b}) \notin R_i^K$. However, $(\bar{a}, \bar{b}) \in Q_i(J)$ and, by 4, we can find a tuple $(\bar{a}, \bar{c})$ in R_i^K, and then a justification $(\varphi_i, \psi_i, \bar{a}, \bar{c})$ in $\mathcal{J}$. A key observation is that $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$ is isomorphic to $C_{(\varphi_i,\psi_i,\bar{a},\bar{c})} \subseteq C_{\mathcal{J}}$ (since they consist of the tuples occurring in $\psi_i(\bar{a}, \bar{\perp}_{(\varphi_i,\psi_i,\bar{a},\bar{b})})$ and $\psi_i(\bar{a}, \bar{\perp}_{(\varphi_i,\psi_i,\bar{a},\bar{c})})$, respectively). Then, for each pair of nulls $\perp$ in $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$ and $\perp'$ in $C_{(\varphi_i,\psi_i,\bar{a},\bar{c})}$ mapped into each other by this isomorphism, we define $v(\perp) = v(\perp')$. Observe that, since $C_{(\varphi_i,\psi_i,\bar{a},\bar{c})} \subseteq C_{\mathcal{J}}$, the valuation v has already been defined on its nulls. As a consequence $v(C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}) = v(C_{(\varphi_i,\psi_i,\bar{a},\bar{c})}) \subseteq v(C_{\mathcal{J}})$. Since this holds for all sub-instances $C_{(\varphi_i,\psi_i,\bar{a},\bar{b})}$ defined in iii, we conclude that $v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J) - E_\Delta) = v(C_{\mathcal{J}})$ that is

$$v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)) = v(C_{\mathcal{J}}) \cup E_\Delta \tag{6}$$

Now recall that T is in $\mathit{Rep}_A(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)))$, and then $T \supseteq \mathit{re\ell}(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K)))$. By comparing (5) and (6) one derives $\mathit{re\ell}(v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K))) = \mathit{re\ell}(v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)))$, therefore $T \supseteq \mathit{re\ell}(v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J)))$. Moreover each tuple $\bar{t}$ in a relation R of T coincides with some tuple $\bar{t}'$ of relation R in $v'(\textsc{CSol}_{\textsc{a}}^{\Gamma_{\alpha'}}(K))$ on positions annotated as closed. Then there are two cases.

- *Case 1.* If $\bar{t}'$ is in $v(C_{\mathcal{J}})$, then $\bar{t}'$ is also in $v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J))$.

- *Case 2.* If $\bar{t}'$ is in E_Γ, then $\bar{t}'$ is an empty tuple and must be annotated all-open. We now show that there exists some all-open annotated tuple also in $v(\textsc{CSol}_{\textsc{a}}^{\Delta_{\alpha'}}(J))$.

Indeed $\bar{t}'$ is also a tuple of R in $\text{CSol}_{\text{A}}^{\Gamma_{\alpha'}}(K)$, therefore there exists an STD ψ_i :- R_i of $\Gamma_{\alpha'}$ such that ψ_i contains an atom of relation R with annotation all-open. Consequently the STD ψ_i :- φ_i is in $\Delta_{\alpha'}$, and thus $\text{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J)$ contains some (possibly empty) tuple of R annotated all-open. Thus also $v(\text{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J))$, contains a tuple annotated all-open in relation R.

In both cases there exists some tuple of relation R in $v(\text{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J))$ coinciding with $\bar{t}$ on closed positions. Since this holds for all tuples $\bar{t}$ in all relations R of T, we conclude that T is in $Rep_A(v(\text{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J)))$.

This proves that $T \in Rep_A(\text{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J))$, that is $(J,T) \in \llparenthesis \Delta_{\alpha'} \rrparenthesis$, and concludes the proof of Claim 5. □

Now consider the query Q over schema τ expressed by the first order formula

$$\beta(\bar{x}_1, \bar{y}_1 \ldots, \bar{x}_k, \bar{y}_k) \;=\; \varphi_1(\bar{x}_1, \bar{y}_1) \wedge \cdots \wedge \varphi_k(\bar{x}_k, \bar{y}_k).$$

Clearly, for each instance K of schema $\{R_1, \ldots R_k\}$, and each instance J of schema τ, we have

$$R_1^K \times \cdots \times R_k^K \subseteq Q(J) \quad \Leftrightarrow \quad R_i^K \subseteq Q_i(J) \text{ for each } i \in \{1 \ldots k\}$$

and

$$\pi_{\bar{x}_1,\ldots,\bar{x}_k}(Q(J)) \;=\; \pi_{\bar{x}_1,\ldots,\bar{x}_k}(R_1^K \times \cdots \times R_k^K) \quad \Leftrightarrow \quad \pi_{\bar{x}_i}(R_i^K) = \pi_{\bar{x}_i}(Q_i(J)) \text{ for each } i \in \{1 \ldots k\}.$$

A non-deterministic exponential time algorithm for the composition problem can be obtained as follows:

- an instance K of schema $\{R_1, \ldots R_k\}$ and of size polynomial in $\|T\|$ is guessed;
- it is checked that $T \in Rep_A(\text{CSol}_{\text{A}}^{\Gamma_{\alpha'}}(K))$ in non-deterministic polynomial time, as stated in Theorem 2;
- finally it is checked in non-deterministic exponential time, as stated in Lemma 2, that there exists $J \in Rep_A(\text{CSol}_{\text{A}}^{\Sigma_\alpha}(S))$ such that $R_1^K \times \cdots \times R_k^K \subseteq Q(J)$ and $\pi_{\bar{x}_1,\ldots,\bar{x}_k}(Q(J)) = \pi_{\bar{x}_1,\ldots,\bar{x}_k}(R_1^K \times \cdots \times R_k^K)$.

By Claim 5, this verifies precisely whether $(S,T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis$, thus proving that $\text{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ is in NEXPTIME.

Hardness and undecidability.

We now prove the hardness results for $\#_{op}\Sigma_\alpha = 0, 1$, and the undecidability of the composition problem for $\#_{op}\Sigma_\alpha > 1$, by means of a reduction from the complement of the query answering problem. Let $(\sigma, \tau, \Gamma_\gamma)$ be an annotated schema mapping and Q an arbitrary FO query over the target represented by an FO formula $\varphi(\bar{x})$. We reduce the complement of the problem $\text{DEQA}(\Gamma_\gamma, Q)$ to the problem $\text{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$, where Σ_α and $\Delta_{\alpha'}$ are constructed from Γ_γ and φ as follows:

- A source schema for Σ is $\sigma \cup \{R'\}$, a target schema is $\tau \cup \{R\}$, where R and R' are relation symbols distinct from all symbols of $\sigma \cup \tau$ and have the same arity as Q, and:

  $$\Sigma_\alpha = \Gamma_\gamma \cup \{R(\bar{x}^{cl}) \text{ :- } R'(\bar{x})\}$$

  where $\bar{x}$ is a tuple of distinct variables.

- A source schema for Δ is $\tau \cup \{R\}$, its target schema is $\{C\}$, with C of the same arity as Q, and the only STD is

$$C(\bar{x}) :- \mathit{adom}(\bar{x}) \wedge \varphi(\bar{x}) \wedge R(\bar{x})$$

  where $\bar{x}$ is a tuple of distinct variables, and $\mathit{adom}(\bar{x})$ is a first-order formula of vocabulary τ checking that each element of $\bar{x}$ either belongs the the active domain of τ-relations or is a constant occurring in $\varphi(\bar{x})$.

- Let finally $\Delta_{\alpha'}$ be an arbitrary annotation of Δ.

Given an instance I of schema σ and a tuple $\bar{t}$ with $|\bar{t}| = |\bar{x}|$, we construct an input (S, T) for $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ as follows. The source instance S is a copy of I on relations of σ, and the interpretation of R' in S is $\{\bar{t}\}$. The target T is the empty instance of relation C.

We now prove that $(S, T) \in \Sigma_\alpha \circ \Delta_{\alpha'}$ if and only if $\bar{t} \notin \mathsf{certain}_{\Gamma_\gamma}(Q, I)$. Assume first that $\bar{t} \notin \mathsf{certain}_{\Gamma_\gamma}(Q, I)$. Then there exists an instance $J_\tau \in \mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Gamma_\gamma}(I))$ such that $\bar{t} \notin Q(J_\tau)$, that is $J_\tau \not\models \mathit{adom}(\bar{t}) \wedge \varphi(\bar{t})$. Let J the instance of schema $\tau \cup \{R\}$ obtained by extending J_τ with instance $\{\bar{t}\}$ for R. The instance J is in $\mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S))$ (that is, $(S, J) \in (\!|\Sigma_\alpha|\!)$), since $\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S)$ coincides with $\textsc{CSol}_{\mathrm{A}}^{\Gamma_\gamma}(I)$ on relations of τ, and interprets R as $\{\bar{t}^{cl}\}$.

The only possible satisfying assignment for the right-hand side of $\Delta_{\alpha'}$ in J is $\bar{x} = \bar{t}$, since $\bar{t}$ is the only tuple in R^J. Indeed, since $J_\tau \not\models \mathit{adom}(\bar{t}) \wedge \varphi(\bar{t})$, we also have $J \not\models \mathit{adom}(\bar{t}) \wedge \varphi(\bar{t})$. Therefore there is no satisfying assignment for the right-hand side of $\Delta_{\alpha'}$ in J, and thus $\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J)$ interprets C as the empty tuple $\{(\_, \alpha')\}$. Then $\mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$ (i.e. $[\![J]\!]^{\Delta_{\alpha'}}$) contains the empty instance T. Consequently $(J, T) \in (\!|\Delta_{\alpha'}|\!)$ and thus $(S, T) \in \Sigma_\alpha \circ \Delta_{\alpha'}$.

Conversely, assume that $\bar{t} \in \mathsf{certain}_{\Gamma_\gamma}(Q, I)$. For an arbitrary instance $J \in \mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Sigma_\alpha}(S))$, let J_τ the restriction of J to vocabulary τ. Clearly $J_\tau \in \mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Gamma_\gamma}(I))$, and thus $\bar{t} \in Q(J_\tau)$, that is $J_\tau \models \mathit{adom}(\bar{t}) \wedge \varphi(\bar{t})$. Then also $J \models \mathit{adom}(\bar{t}) \wedge \varphi(\bar{t})$, and so $\bar{t}$ is a satisfying assignment for the right-hand side of $\Delta_{\alpha'}$ in J. As there are no other satisfying assignments, $\mathit{rel}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$ interprets C as $\{\bar{t}\}$. This implies that, for each instance T' in $\mathit{Rep}_A(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$ (regardless of the annotation α'), the tuple $\bar{t}$ is in $C^{T'}$. As a consequence, the empty instance T does not belong to $[\![J]\!]^{\Delta_{\alpha'}}$ and, because this holds for all $J \in [\![S]\!]^{\Sigma_\alpha}$, we have that $(S, T) \notin \Sigma_\alpha \circ \Delta_{\alpha'}$.

This completes the reduction. Now observe that $\#_{op}(\Sigma_\alpha) = \#_{op}(\Gamma_\gamma)$, therefore from Theorem 3 it follows that:

- there exist mappings Σ_α and Δ, with $\#_{op}(\Sigma_\alpha) = 0$, such that $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ is NP-complete for each annotation α' on Δ;

- there exist mappings Σ_α and Δ with $\#_{op}(\Sigma_\alpha) = 1$ such that $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ is NEXPTIME-complete for each annotation α' on Δ;

- for each $k > 1$ there exist mappings Σ_α and Δ with $\#_{op}(\Sigma_\alpha) = k$ such that $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ is undecidable for each annotation α' on Δ.

This proves the complexity bounds, but there is still one statement claimed above that is not yet proved. Namely, for the case of $\#_{op}(\Sigma_\alpha) = 0$ (i.e., under the CWA) we claimed that Σ and Δ can be found that only used CQ-STDs such that $\textsc{Comp}(\Sigma_{cl}, \Delta_{\alpha'})$ is NP-complete for every annotation α'. We now give a direct proof of this special case, adapting the proof of NP-hardness of the composition problem under OWA from [13].

We reduce 3-*colorability* to the composition problem $\textsc{Comp}(\Sigma_{cl}, \Delta_{\alpha'})$ where Σ is the following set of STDs over source schema $\sigma = \{V, E, D\}$ and target schema $\tau = \{C, E', D'\}$:

$$\Sigma : \begin{array}{rcl} C(x,z) & :- & V(x) \\ E'(x,y) & :- & E(x,y) \\ D'(x,y) & :- & D(x,y) \end{array}$$

Δ is the following set of STDs over source schema τ and target schema $\omega = \{\bar{D}\}$:

$$\Delta : \begin{array}{rcl} \bar{D}(u,v) & :- & E'(x,y) \wedge C(x,u) \wedge C(y,v) \\ \bar{D}(u,v) & :- & D'(u,v) \end{array}$$

Finally, α' is an arbitrary fixed annotation on Δ.

An input graph G for the 3-colorability problem can be translated into the following instances S and T of schema σ and ω, respectively:

- V^S is interpreted as the set of nodes of G;
- E^S as the set of edges of G;
- D^S and $\bar{D}^T$ are interpreted as the relation $\neq$ between colors $\{r, g, b\}$.

We now prove that $(S,T) \in (\!|\Sigma_{cl}|\!) \circ (\!|\Delta_{\alpha'}|\!)$ if and only if G is 3-colorable. Let $\{\nu_1, \ldots, \nu_n\}$ be the set of vertices of G, observe that $\textsc{CSol}_{\mathrm{A}}^{\Sigma_{cl}}(S)$ interprets relations E' and D' as (all-closed annotated) copies of E^S and D^S respectively, and relation C as $\{(\nu_i^{cl}, \bot_i^{cl}) \mid i = 1, \ldots, n\}$. For each valuation v of nulls $\bot_i$, $i = 1, \ldots, n$, let $J_v = v(rel(\textsc{CSol}_{\mathrm{A}}^{\Sigma_{cl}}(S)))$ and consider J_v as an input to $\Delta_{\alpha'}$. Observe that $\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J_v)$ is an annotated instance without nulls, and therefore $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J_v))$ is in $[\![J_v]\!]^{\Delta_{\alpha'}}$, no matter what the annotation α' is. Moreover, $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J_v))$ interprets relation $\bar{D}$ as

$$D^S \cup \{(v(\bot_i), v(\bot_j)) \mid (\nu_i, \nu_j) \text{ is an edge of } G\}. \tag{7}$$

Now assume that G is 3-colorable, and let c_i be the color associated to ν_i in an $\{r, g, b\}$-coloring of G. Define a valuation $\bar{v}$ on $\textsc{CSol}_{\mathrm{A}}^{\Sigma_{cl}}(S)$ as $\bar{v}(\bot_i) := c_i$. Then, for each edge (ν_i, ν_j) of G, the pair $(\bar{v}(\bot_i), \bar{v}(\bot_j))$ is in D^S. This implies that $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J_{\bar{v}}))$ interprets relation $\bar{D}$ as D^S. That is, $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J_{\bar{v}})) = T$, and therefore $T \in [\![J_{\bar{v}}]\!]^{\Delta_{\alpha'}}$. Clearly $J_{\bar{v}} \in [\![S]\!]^{\Sigma_{cl}}$, and therefore $(S,T) \in (\!|\Sigma_{cl}|\!) \circ (\!|\Delta_{\alpha'}|\!)$.

Conversely, assume $(S,T) \in (\!|\Sigma_{cl}|\!) \circ (\!|\Delta_{\alpha'}|\!)$. Then there exists $J \in [\![S]\!]^{\Sigma_{cl}}$ such that $T \in [\![J]\!]^{\Delta_{\alpha'}}$. Let $\tilde{v}$ be the valuation of $\textsc{CSol}_{\mathrm{A}}^{\Sigma_{cl}}(S)$ such that $J = J_{\tilde{v}}$. As $T \in [\![J]\!]^{\Delta_{\alpha'}}$, we see that T contains $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$, no matter what the annotation α' is. Moreover, as mentioned already, $\mathit{re\ell}(\textsc{CSol}_{\mathrm{A}}^{\Delta_{\alpha'}}(J))$ interprets relation $\bar{D}$ as in (7), while T interprets $\bar{D}$ as D^S. This implies that, for each (ν_i, ν_j) edge of G, the tuple $(\tilde{v}(\bot_i), \tilde{v}(\bot_j))$ is in D^S. Consequently the mapping which assigns to each vertex ν_i of G the color $\tilde{v}(\bot_i)$ is an $\{r, g, b\}$-coloring of G. This concludes the proof of Theorem 4. □

By Corollary 4 and Theorem 4, if both Σ and Δ contain only CQ-STDs, the composition problem is in NP if Σ has an all-closed annotation, or Δ has an all-open annotation. Moreover, composing Σ_{cl} with any $\Delta_{\alpha'}$ (even if both have FO-STDs) matches the complexity of OWA-composition achieved only for CQ-STDs. For more open nulls, our results suggest that one needs to restrict STDs to monotone to keep the complexity reasonable.

The results on the complexity of the composition problem $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$ presented in this section are summarized in the table in Figure 1.

$\Delta_{\alpha'}$ Σ_α	arbitrary	$\alpha' = op$ and monotone STDs
$\#_{op} = 0$	NP-complete	NP-complete
$\#_{op} = 1$	NEXPTIME-complete	
$\#_{op} > 1$	undecidable	

Figure 1: Complexity of $\textsc{Comp}(\Sigma_\alpha, \Delta_{\alpha'})$

Syntactic descriptions of composition

As was noticed in [13], under the OWA, schema mapping composition cannot be captured syntactically without increasing the class of STDs: there exist Σ and Δ (with CQ-STDs only) such that one cannot find Γ with FO-STDs satisfying $\llparenthesis\Gamma\rrparenthesis^{\text{OWA}} = \llparenthesis\Sigma\rrparenthesis^{\text{OWA}} \circ \llparenthesis\Delta\rrparenthesis^{\text{OWA}}$. One can see this by a complexity gap argument: OWA-schema mappings have low (AC^0) complexity, but their composition could be NP-hard [13]. Arbitrarily annotated mappings could be of higher complexity, but we can still show the following strong failure of closure under composition without any additional assumptions.

Proposition 6. *There exist schema mappings with CQ-STDs Σ and Δ such that, given their arbitrary annotations α and α', there is no annotated mapping $\Gamma_{\alpha''}$ with FO-STDs that satisfies $\llparenthesis\Gamma_{\alpha''}\rrparenthesis = \Sigma_\alpha \circ \Delta_{\alpha'}$.*

Proof. Consider the following set of STDs Σ with a source schema $\sigma = \{R, P\}$ and a target schema $\tau = \{N, C\}$, where all relations are unary:

$$\Sigma : \begin{cases} N(y) & \text{:-} \quad R(x) \\ C(x) & \text{:-} \quad P(x) \end{cases}$$

and a set of STDs Δ, whose source schema is τ and a target schema ω, containing a single binary relation D:

$$\Delta : \;\; D(x,y) \quad \text{:-} \quad C(x) \wedge N(y)$$

Fix an arbitrary annotation α on Σ, and α' on Δ, and assume by contradiction, that there exists a set of annotated STDs Γ_γ, over the source schema σ and the target schema ω, satisfying $\llparenthesis\Gamma_\gamma\rrparenthesis = \llparenthesis\Sigma_\alpha\rrparenthesis \circ \llparenthesis\Delta_{\alpha'}\rrparenthesis$.

Let k be the maximum number of atoms in the left-hand side of any dependency in Γ_γ; then choose $n > k$ and let S_0 be an instance of schema σ having $R = \{0\}$ and $P = \{1, \ldots, n\}$. The following claim describes properties of instances related to S_0 in $\llparenthesis\Sigma_\alpha\rrparenthesis \circ \llparenthesis\Delta_{\alpha'}\rrparenthesis$.

Claim 6. *Let T_0 be an instance $\{(i, \bot), i = 1, \ldots, n\}$ of relation D. Then*

1. *For each valuation v on T_0, we have $(S_0, v(T_0)) \in \llparenthesis\Sigma_\alpha\rrparenthesis \circ \llparenthesis\Delta_{\alpha'}\rrparenthesis$; and*
2. *For each instance T such that $(S_0, T) \in \llparenthesis\Sigma_\alpha\rrparenthesis \circ \llparenthesis\Delta_{\alpha'}\rrparenthesis$, we have $T \supseteq v(T_0)$, for some valuation v.*

Proof of Claim 6. Observe that $\mathit{rel}(\textsc{CSol}_{\text{A}}^{\Sigma_\alpha}(S_0))$ interprets relation N as a singleton $\{\bot_0\}$, and C as $\{1, \ldots, n\}$. Thus, regardless of the annotation α, for each solution J such that $(S_0, J) \in \llparenthesis\Sigma_\alpha\rrparenthesis$, the relation N^J contains $\{v_0(\bot_0)\}$, for some valuation v_0, and C^J contains $\{1, \ldots, n\}$. Therefore $\mathit{rel}(\textsc{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J))$ contains $\{1, \ldots, n\} \times \{v_0(\bot_0)\}$, which belongs to $\mathit{Rep}(T_0)$. Clearly, also each instance in $\mathit{Rep}_A(\textsc{CSol}_{\text{A}}^{\Delta_{\alpha'}}(J))$ (that is, each instance T such that $(J, T) \in \llparenthesis\Delta_{\alpha'}\rrparenthesis$) contains $\{1, \ldots, n\} \times$

$\{v_0(\bot_0)\}$, regardless of α'. As a consequence, each instance T of schema ω such that $(S_0, T) \in (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$ contains some valuation of T_0. This proves 2).

Moreover, for each valuation v of $\bot$, if we take the instance J_0 of schema τ such that $N^{J_0} = \{v(\bot)\}$ and $C^{J_0} = \{1, \dots, n\}$, then $(S_0, J_0) \in (\!|\Sigma_\alpha|\!)$ and $(J_0, v(T_0)) \in (\!|\Delta_{\alpha'}|\!)$. This proves 1) and concludes the proof of the claim. □

Now we distinguish two cases.

- *Case 1.* There is a tuple (a, c), with $c \in \mathsf{Const}$, in $\mathit{rel}(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$. Then any instance of $\mathit{Rep}_A(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$ contains a tuple (b, c), for some b. Now take any valuation v on T_0 such that $v(\bot) \neq c$, then each tuple of $v(T_0)$ is of the form (i, d) with $d \neq c$, thus $v(T_0) \notin \mathit{Rep}_A(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$. In other words $(S_0, v(T_0)) \notin (\!|\Gamma_\gamma|\!)$ and thus $(S_0, v(T_0)) \notin (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$. But this contradicts Claim 6.
- *Case 2.* Only nulls occur in the second column of $\mathit{rel}(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$. By construction of the canonical solution, k is the maximum number of tuples in $\mathit{rel}(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$ that can have a null in common. Therefore, by assigning distinct constants to distinct nulls, one obtains a valuation $\bar{T}$ of $\mathit{rel}(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$ where the same constant occurs at most k times in the second column. On the other hand, no matter what the annotation γ is, $\bar{T}$ is in $\mathit{Rep}_A(\text{CSOL}_{\text{A}}^{\Gamma_\gamma}(S_0))$. Thus $(S_0, \bar{T}) \in (\!|\Gamma_\gamma|\!)$ and then $(S_0, \bar{T}) \in (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$. This implies, by Claim 6, that $\bar{T}$ contains a valuation of T_0, and therefore it contains $n > k$ occurrences of the same constant in the second column, which contradicts the previous conclusion.

In both cases contradiction is reached, thus there exists no annotated mapping Γ_γ such that $(\!|\Gamma_\gamma|\!) = (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$. This completes the proof of Proposition 6. □

So we need to extend the class of mappings to make it closed under composition. We say that a class of mappings $\mathcal{C}$ with a semantics $(\!|\cdot|\!)^{\mathcal{C}}$ is *closed under composition* if for every two mappings $(\sigma, \tau, M_{\sigma\tau})$ and $(\tau, \omega, M_{\tau\omega})$ from $\mathcal{C}$, there exists another mapping $(\sigma, \omega, M_{\sigma\omega})$ from $\mathcal{C}$ so that

$$(\!|M_{\sigma\omega}|\!)^{\mathcal{C}} \;=\; (\!|M_{\sigma\tau}|\!)^{\mathcal{C}} \;\circ\; (\!|M_{\tau\omega}|\!)^{\mathcal{C}}.$$

In [13], such a class was found under the OWA: it was based on Skolemized CQ-STDs[1]. We now define such Skolemized STDs in an annotated setting, and prove a composition lemma for them that gives us two classes of annotated mappings closed under composition: the class of [13] and its closed-world analog.

Assume that we have a countable collection $\mathcal{F}$ of function symbols. Given two schemas σ and τ, an *annotated Skolemized STD*, or an annotated *SkSTD*, over them is an expression of the form:

$$\psi_\tau(u_1, \dots, u_k) \;:\!-\; \varphi_\sigma(x_1, \dots, x_n),$$

together with an annotation α of ψ_τ where

- φ_σ is an FO formula over $\sigma \cup \mathcal{F}$ whose atomic subformulae are either $R(\bar{z})$, where $\bar{z}$ are variables, or $y = f(\bar{z})$, where y is a variable;
- ψ_τ is a conjunction of atomic τ formulae; and
- each u_i is either one of the x_j's, or $f(\bar{z})$, for some $f \in \mathcal{F}$ and $|\mathit{arity}(f)|$ variables $\bar{z}$ among $\bar{x}$.

[1] Such STDs were called second-order in [13] because their semantics was defined by existentially quantifying over the Skolem functions. We prefer to call them Skolemized STDs, and use CQ, FO, etc in STDs to restrict the class of formulae φ in $\psi :\!- \varphi$.

Annotations are defined as before, i.e. by assigning *op* or *cl* to each position in each atom in ψ_τ.

For example, if our source has tuples (*em,proj*) of employee names and projects and we want to create a target with tuples (*empl_id,em,phone*) that invents ids and phones of employees, we can capture this by an annotated SkSTD

$$T(f(em)^{cl},\ em^{cl},\ g(em,proj)^{op})\ \ :\!-\ \ S(em,proj) \tag{8}$$

indicating that one id is created for each name, with f being the function from names to ids. Using a null instead of $f(em)$ would have generated a new null for each (*em,proj*) pair, rather than just the name. The *phone* attribute is open, allowing employees to have multiple phones.

Next, we extend the definition of the semantics to annotated mappings $(\sigma, \tau, \Sigma_\alpha)$ with SkSTDs. Let S be a source instance. Let $F = \{f_1, \ldots, f_r\}$ be the set of function symbols used in Σ_α, and for each m-ary f_i, let f'_i be a function from Const^m to Const. For this set $F' = \{f'_1, \ldots, f'_r\}$, we construct a solution $\textsc{Sol}_{F'}^{\Sigma_\alpha}(S)$ as follows: compute the result of φ_σ in S, with functions interpreted as F', and for each tuple $\bar{a}$ in it, put annotated tuples in the target to satisfy $\psi_\tau(\bar{u}')$, where, if $u_i = x_j$, then $u'_i = a_j$, and if $u_i = f(\bar{x})$ then $u'_i = f'(\bar{a})$. The annotation is the same as in Σ_α. If φ_σ evaluates to the empty set, then, as before, empty annotated tuples are added.

In our example (8), if $S = \{(John, P1)\}$ and $f'(John) = 001$ and $g'(John, P1) = 1234$, then $\textsc{Sol}_{\{f',g'\}}(S)$ has one tuple $(001^{cl}, John^{cl}, 1234^{op})$.

For Σ_α with SkSTDs, the semantics of S is given by

$$[\![S]\!]^{\Sigma_\alpha} = \bigcup_{F'} Rep_A\big(\textsc{Sol}_{F'}(S)\big),$$

as F' ranges over functions from Const to Const that match the arity of functions in F. Note that as $\textsc{Sol}_{F'}(S)$ has no nulls, the only effect of applying Rep_A to it is adding tuples that coincide with some tuple t in $\textsc{Sol}_{F'}(S)$ on all attributes annotated *cl.* For example, $[\![S]\!]^{\Sigma_\alpha}$ for the SkSTD (8) and the above source will contain a relation $\{(001, John, 1234), (001, John, 5678)\}$.

Then, finally, we define

$$(\!|\Sigma_\alpha|\!) = \{(S,T)\ \mid\ T \in [\![S]\!]^{\Sigma_\alpha}\}.$$

First observe that if Σ is a set of un-annotated SkSTDs, then $(\!|\Sigma_{op}|\!)$ is precisely the semantics of [13] — that is, [13] used the open-world semantics. A formal proof is given below.

Proposition 7. *If Σ is a set of unannotated SkSTDs that use function symbols $F = \{f_1, \ldots, f_r\}$, and Ψ_Σ stands for the corresponding second-order sentence*

$$\Psi_\Sigma = \exists f_1 \ldots \exists f_r \bigwedge_{\psi :- \varphi(\bar{x})\ \in\ \Sigma} \forall \bar{x}\ \big(\varphi(\bar{x}) \rightarrow \psi\big)$$

then $(\!|\Sigma_{op}|\!)$ is precisely the set of pairs (S,T) such that $(S,T) \models \Psi_\Sigma$.

Note that the set of pairs (S,T) such that $(S,T) \models \Psi_\Sigma$ represents the semantics of the mapping Σ according to [13].

Proof. Let γ_Σ the first order sentence

$$\bigwedge_{\psi :- \varphi(\bar{x})\ \in\ \Sigma} \forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi).$$

It follows from the definition of Rep_A for all-open annotated instances that, for given actual functions F', an instance T is in $Rep_A(\textsc{Sol}_{F'}^{\Sigma_{op}}(S))$ if and only if $r\ell(\textsc{Sol}_{F'}^{\Sigma_{op}}(S)) \subseteq T$.

On the other hand, a tuple $\bar{t}$ is in a relation R of $\mathit{re\ell}(\text{Sol}_{F'}^{\Sigma_{op}}(S))$ if and only if there exists an SkSTD ψ :- $\varphi(\bar{x})$ in Σ and a constant tuple $\bar{a}$ such that $\varphi(\bar{a})$ holds in S, with F interpreted as F', and $R(\bar{w})$ is an atom of ψ with $\bar{w}[\bar{x} \to \bar{a}, \bar{F} \to \bar{F}'] = \bar{t}$.

Consequently, T is in $\mathit{Rep}_A(\text{Sol}_{F'}^{\Sigma_{op}}(S))$ if and only if whenever for some SkSTD $\psi(\bar{u})$:- $\varphi(\bar{x})$ of Σ the formula $\varphi(\bar{a})$ holds in S with F interpreted as F', and $R(\bar{w})$ is an atom of $\psi(\bar{u})$, then we have that the tuple $\bar{w}[\bar{x} \to \bar{a}, \bar{F} \to \bar{F}']$ is in R^T. That is, T is in $\mathit{Rep}_A(\text{Sol}_{F'}^{\Sigma_{op}}(S))$ if and only if T satisfies γ_Σ with F interpreted as F'.

It follows that $(S,T) \in (\!|\Sigma_{op}|\!)$ (that is, there exist actual functions F' such that T is in $\mathit{Rep}_A(\text{Sol}_{F'}^{\Sigma_{op}}(S)))$ if and only if $(S,T) \models \Psi_\Sigma$, as claimed. □

Even though mappings with SkSTDs do not explicitly allow null values, semantically they extend the usual STD-based mappings:

Lemma 4. *For every annotated mapping $(\sigma, \tau, \Sigma_\alpha)$ based on STDs there exists an equivalent mapping $(\sigma, \tau, \Gamma_\alpha)$ with the same annotations based on SkSTDs, i.e. $(\!|\Sigma_\alpha|\!) = (\!|\Gamma_\alpha|\!)$. Furthermore, the right-hand sides of STDs in Σ and in Γ are the same.*

Proof. For each STD $\psi(\bar{x}, \bar{z})$:- $\varphi(\bar{x}, \bar{y})$ in Σ, and for each variable z in $\bar{z}$, we create a fresh function symbol $f_{(\varphi,\psi,z)}$ of arity $|\bar{x}| + |\bar{y}|$, then we replace each occurrence of z in ψ with $f_{(\varphi,\psi,z)}(\bar{x}, \bar{y})$. The same annotation α is maintained on atoms of ψ. Γ_α is the set of SkSTDs thus obtained from Σ_α.

It was already observed in [13] that Γ and Σ are logically equivalent, therefore they are equivalent under the semantics of [13]. This implies that, if α is all-open, $(\!|\Sigma_\alpha|\!) = (\!|\Gamma_\alpha|\!)$; we now prove that the equality holds also for an arbitrary α.

In what follows we omit the superscripts Σ_α and Γ_α, as they will be always understood. Fix an arbitrary source instance S; recall that in $\text{CSol}_\text{A}(S)$ there is exactly one distinct null for each tuple $(\varphi, \psi, \bar{a}, \bar{b}, z)$ where φ and ψ identify a STD in Σ_α, $(\bar{a}, \bar{b})$ is a satisfying assignment for $\varphi(\bar{x}, \bar{y})$ in S and z is a variable among $\bar{z}$ in $\psi(\bar{x}, \bar{z})$.

Given actual functions F' interpreting function symbols of Γ_α such that $f'_{(\varphi,\psi,z)}$ interprets $f_{(\varphi,\psi,z)}$, and given a valuation v of nulls of $\text{CSol}_\text{A}(S)$, we write $F' \sim v$ if for each null $\bot_{(\varphi,\psi,\bar{a},\bar{b},z)}$ of $\text{CSol}_\text{A}(S)$, we have $v(\bot_{(\varphi,\psi,\bar{a},\bar{b},z)}) = f'_{(\varphi,\psi,z)}(\bar{a}, \bar{b})$. Note that if $F' \sim v$ then $v(\text{CSol}_\text{A}(S)) = \text{Sol}_{F'}(S)$.

Now assume that a target instance T is in $[\![S]\!]^{\Sigma_\alpha}$. Then there exists a valuation v on $\text{CSol}_\text{A}(S)$ such that T is in $\mathit{Rep}_A(v(\text{CSol}_\text{A}(S)))$. From this v one can always construct actual functions F' for Γ_α such that $v \sim F'$. Therefore $\text{Sol}_{F'}(S) = v(\text{CSol}_\text{A}(S))$, and thus T, being in $\mathit{Rep}_A(\text{Sol}_{F'}(S))$, is in $[\![S]\!]^{\Gamma_\alpha}$.

Conversely assume that $T \in [\![S]\!]^{\Gamma_\alpha}$. Then T is in $\mathit{Rep}_A(\text{Sol}_{F'}(S))$, for some actual functions F'. Similarly one can construct a valuation v such that $v \sim F'$, and then $T \in \mathit{Rep}_A(v(\text{CSol}_\text{A}(S)))$ and thus, $T \in [\![S]\!]^{\Sigma_\alpha}$. This concludes the proof of Lemma 4. □

We now state the main technical lemma which shows when two annotated mappings can be composed.

Lemma 5. *Let Σ_α and $\Delta_{\alpha'}$ be two schema mappings with annotated SkSTDs such that either*

- *the annotation of $\Delta_{\alpha'}$ is all-open, and it only has monotone queries in its SkSTDs; or*
- *the annotation of Σ_α is all-closed.*

Then one can construct a composition mapping $\Gamma_{\alpha'}$ (i.e. $(\!|\Gamma_{\alpha'}|\!) = (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$) with annotated SkSTDs such that

- *the left-hand sides and annotations of SkSTDs in $\Gamma_{\alpha'}$ and $\Delta_{\alpha'}$ are the same;*
- *the right-hand sides of SkSTDs in $\Gamma_{\alpha'}$ are CQs if the same is true for Σ_α and $\Delta_{\alpha'}$.*

Proof. We construct a mapping $\Gamma_{\alpha'}$, with annotated SkSTDs, from Σ_α and $\Delta_{\alpha'}$ using an adaptation of the composition algorithm defined in [13].

We proceed according to the following steps:

1. Variables and function symbols are renamed, so that no variable, as well as no function symbol, occurs both in Σ_α and $\Delta_{\alpha'}$.
2. Constraints in Σ_α are put in the following normal form. Each SkSTD $R_1(\bar{u}_1) \wedge \cdots \wedge R_m(\bar{u}_m) \coloneq \varphi(\bar{x})$ in Σ is replaced with the set of dependencies $R_j(\bar{u}_j) \coloneq \varphi(\bar{x})$, for $j = 1, \ldots, m$, and annotations are inherited from α. Observe that the transformation preserves $(\!|\Sigma_\alpha|\!)$.
3. For each SkSTD $\psi \coloneq \eta$ with annotation α_ψ in $\Delta_{\alpha'}$ we put a SkSTD $\psi \coloneq \eta'$ with annotation α_ψ in $\Gamma_{\alpha'}$. The formula η' is obtained from η by replacing each relational atom $R(\bar{y})$ occurring in η with the sub-formula $\beta_{R(\bar{y})}$ constructed as follows. Let

$$\begin{array}{lcl} R(\bar{u}_1) & \coloneq & \varphi_1(\bar{z}_1) \\ \vdots & & \\ R(\bar{u}_k) & \coloneq & \varphi_k(\bar{z}_k) \end{array}$$

be precisely the set of SkSTDs in the normal form of Σ having an atom of relation R in the left-hand side. Then $\beta_{R(\bar{y})}$ is the following formula over the vocabulary that includes σ and function symbols used in SkSTDs in Σ:

$$\beta_{R(\bar{y})} \overset{\text{def}}{=} \bigvee_{j=1}^{k} \exists \bar{z}_j (\varphi_j(\bar{z}_j) \wedge \bar{y} = \bar{u}_j)$$

It can be easily verified that $\Gamma_{\alpha'}$ is a set of SkSTDS over source schema σ and target schema ω. Furthermore, by construction, each SkSTD in $\Gamma_{\alpha'}$ is obtained from a SkSTD in $\Delta_{\alpha'}$ by preserving left-hand side and annotation, therefore $\Gamma_{\alpha'}$ and $\Delta_{\alpha'}$ have the same annotated left-hand sides.

In the case that Σ_α and $\Delta_{\alpha'}$ contain only CQ-SkSTDs, we now show that $\Gamma_{\alpha'}$ is equivalent (in terms of $(\!|\cdot|\!)$) to another mapping with only CQ-SkSTDs, which still preserves left-hand sides of $\Delta_{\alpha'}$. Let $\psi \coloneq \eta$ be an arbitrary SkSTD in Δ, and let $\eta = \theta \wedge R_1(\bar{x}_1) \wedge \cdots \wedge R_n(\bar{x}_n)$, where θ is a conjunction of equality atoms. We know that the corresponding SkSTD in $\Gamma_{\alpha'}$ is $\psi \coloneq \eta'$ where η' is of the form $\eta' = \theta \wedge \bigwedge_{i=1}^{n} \bigvee_{j=1}^{k_i} \gamma_{ij}$, with $\gamma_{ij} = \exists \bar{z}_{ij} \delta_{ij}$, and δ_{ij} a conjunction of atomic sub-formulae of free variables $\bar{x}_i, \bar{z}_{ij}$. Clearly η' can be rewritten as

$$\bigvee_{j_1=1}^{k_1} \cdots \bigvee_{j_n=1}^{k_n} \bigwedge_{i=1}^{n} (\gamma_{ij_i} \wedge \theta),$$

and thus the SkSTD $\psi \coloneq \eta'$ is equivalent to the collection of SkSTDs $\{\psi \coloneq \bigwedge_{i=1}^{n} (\gamma_{ij_i} \wedge \theta),\ j_i \in 1, \ldots, k_i \text{ for } i = 1, \ldots, n\}$.

Now assume without loss of generality that variables $\bar{z}_{ij}$ are all distinct from variables $\bar{z}_{hk}$ for $i \neq h$ or $j \neq k$. Then $\psi \coloneq \eta'$ is equivalent to the collection of SkSTDs

$$\Pi = \{\psi \coloneq \exists \bar{z}_{1j_1} \ldots \bar{z}_{nj_n} \bigwedge_{i=1}^{n} (\delta_{ij_i} \wedge \theta), \quad \text{for } j_i \in 1, \ldots, k_i \text{ for } i = 1, \ldots, n\},$$

with annotations the same as in α'.

In each SkSTD of Π, existential quantifiers can be dropped without changing the semantics $(\!|\Pi|\!)$. Indeed terms of ψ are not based on variables z_{ij_i}, therefore for each source instance S and actual function H', the computation of $\textsc{Sol}^{\Pi}_{H'}(S)$ is not affected by the presence of the quantifiers. (Note that it is important that we work here with SkSTDs, and new values invented in targets are applications of function terms; hence, variables that are not used in those terms or among target variables can be quantified out without changing the semantics.)

To sum up, if $\Delta_{\alpha'}$ and Σ_α contain only CQ-SkSTD, each SkSTD ψ :- η' in $\Gamma_{\alpha'}$ can be replaced with the set of CQ-SkSTDs $\{\psi :- \bigwedge_{i=1}^{n}(\delta_{ij_i} \wedge \theta), j_i \in 1, \ldots, k_i \text{ for } i = 1, \ldots, n\}$, without affecting $(\!|\Gamma_{\alpha'}|\!)$. Hence $\Gamma_{\alpha'}$ is equivalent to a mapping with only CQ-SkSTDs which has the same left-hand sides of $\Delta_{\alpha'}$.

We now show that, if α is the all-closed annotation, then $(\!|\Gamma_{\alpha'}|\!) = (\!|\Sigma_\alpha|\!) \circ (\!|\Delta_{\alpha'}|\!)$. Let F and G and H the sets of function symbols used in Σ_α, $\Delta_{\alpha'}$ and $\Gamma_{\alpha'}$, respectively. Clearly H is a subset of $F \cup G$.

For each sub-formula φ occurring in the right-hand side of an SkSTD of $\Delta_{\alpha'}$ (i.e., a formula over τ and G), we let φ' the formula obtained from φ by replacing each relational atom $R(\bar{y})$ with $\beta_{R(\bar{y})}$, as defined above. Note that φ' is a formula over σ and function symbols in H.

We start by proving the following claim.

Claim 7. *If S is an instance of schema σ, and F' and G' are functions interpreting function symbols F and G respectively, and H' is the set of functions interpreting each function symbol of $H \cap F$ as in F' and each function symbol of $H \cap G$ as in G', then the following hold:*

a) For each SkSTD ψ :- η of Δ, each sub-formula $\varphi(\bar{x})$ of η, and each tuple of constants $\bar{a}$, the formula $\varphi(\bar{a})$ holds in rel$(\text{SOL}_{F'}^{\Sigma_\alpha}(S))$, with functions G interpreted as G', if and only if $\varphi'(\bar{a})$ holds in S, with functions from H interpreted as H'.

b) $\text{SOL}_{H'}^{\Gamma_{\alpha'}}(S) = \text{SOL}_{G'}^{\Delta_{\alpha'}} \circ \mathit{rel} \circ \text{SOL}_{F'}^{\Sigma_\alpha}(S)$.

Proof of Claim 7. Throughout the proof of the claim we let J stand for $\mathit{rel}(\text{SOL}_{F'}^{\Sigma_\alpha}(S))$.

Proof of a). The proof is by induction on the structure of φ:

- If $\varphi(\bar{x})$ consists of a relational atom $R(\bar{z})$ (where $\bar{z}$ is a tuple of variables and $\bar{x}$ represents the distinct variables occurring in $\bar{z}$) then $\varphi'(\bar{x}) = \beta_{R(\bar{z})}$. In particular, φ' can only mention functions from F.

  Therefore $\varphi(\bar{a})$ holds in J under interpretation of functions G':

  iff the tuple $\bar{z}[\bar{x} \to \bar{a}]$ is in R^J; that is

  iff (by definition of J as $\text{SOL}_{F'}^{\Sigma_\alpha}(S)$) there exists a SkSTD $R(\bar{u}_j)$:- $\varphi_j(\bar{z}_j)$ in Σ (in normal form), and a tuple $\bar{b}$ over Const such that $\varphi_j(\bar{b})$ holds in S, under interpretation F' of F, and $\bar{u}_j[\bar{z}_j \to \bar{b}, F \to F'] = \bar{z}[\bar{x} \to \bar{a}]$;

  iff (by construction of $\beta_{R(\bar{z})}$) there exists a disjunct $\gamma(\bar{x})$ in $\beta_{R(\bar{z})}$ ($\gamma(\bar{x}) = \exists\bar{z}_j(\varphi_j(\bar{z}_j) \wedge \bar{z} = \bar{u}_j)$) such that $\gamma(\bar{a})$ holds in S under interpretation F' of F;

  iff $\varphi'(\bar{a})$ holds in S under interpretation F' of F.

  iff $\varphi'(\bar{a})$ holds in S under interpretation H' of H (since H' interprets functions of F as in F', and only functions of F occur in $\varphi'(\bar{x})$).

- If $\varphi(\bar{x})$ consists of an equality atom $y = g(\bar{y})$ where y and $\bar{y}$ are variables in $\bar{x}$ and $g \in G$, then $\varphi'(\bar{x}) = \varphi(\bar{x})$. Then the statement holds trivially since G' and H' interpret G with the same actual functions;

- if $\varphi(\bar{x}) = \gamma(\varphi_1(\bar{x}_1), \ldots, \varphi_k(\bar{x}_k))$, where γ is an arbitrary boolean combination of formulas $\varphi_1(\bar{x}_1), \ldots, \varphi_k(\bar{x}_k)$, then $\varphi'(\bar{x}) = \gamma(\varphi'_1(\bar{x}_1), \ldots, \varphi'_k(\bar{x}_k))$. Similarly if $\varphi(\bar{x}) = Q\bar{y}\varphi_1(\bar{x}, \bar{y})$, where Q is either $\forall$ or $\exists$, then $\varphi'(\bar{x}) = Q\bar{y}\varphi'_1(\bar{x}, \bar{y})$. Therefore, as we know from the induction hypothesis that a) holds for sub-formulas $\varphi_1, \ldots, \varphi_k$, we see that it holds for φ as well. This proves a).

Proof of b). By construction of $\Gamma_{\alpha'}$, a non empty annotated tuple $(\bar{t}_0, \alpha_0)$ is in relation R of $\text{SOL}_{H'}^{\Gamma_{\alpha'}}(S)$ if and only if there exists an SkSTD ψ :- $\eta(\bar{x})$ in $\Delta_{\alpha'}$, and a tuple $\bar{a}$ over Const such that:

1. $\eta'(\bar{a})$ holds in S under the interpretation H' of H (i.e., under the interpretation F' of symbols in F),
2. ψ contains an atom $R(\bar{u})$ with annotation α_0 and
3. $\bar{t}_0 = \bar{u}[\bar{x} \to \bar{a}, H \to H']$.

By item a) proved above, in 1), we have that $\eta'(\bar{a})$ holds in S under interpretation H' of H if and only if $\eta(\bar{a})$ holds in J under interpretation G' of G. Moreover, in 3), terms of $\bar{u}$ contain only function symbols of G, therefore $\bar{u}[\bar{x} \to \bar{a}, H \to H'] = \bar{u}[\bar{x} \to \bar{a}, G \to G']$. We can conclude that the non empty annotated tuple $(\bar{t}_0, \alpha_0)$ is in relation R of $\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S)$, if and only if $(\bar{t}_0, \alpha_0)$ is a tuple of R in $\textsc{Sol}_{G'}^{\Delta_{\alpha'}}(J)$.

Similarly, an empty annotated tuple $(\_, \alpha_0)$ is in relation R of $\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S)$ if and only if there exists a SkSTD $\psi :- \eta(\bar{x})$ in $\Delta_{\alpha'}$, and a tuple $\bar{a}$ over Const such that:

1. $\eta'(\bar{a})$ evaluates to the empty set over S under interpretation H' of H and
2. ψ contains an atom $R(\bar{u})$ with annotation α_0.

Again by a), the formula $\eta'(\bar{x})$ evaluates to the empty set over S under interpretation H' of H if and only if $\eta(\bar{a})$ evaluates to the empty set over J with G interpreted as G'. As a consequence, $(\_, \alpha_0)$ is a tuple of relation R in $\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S)$, if and only if $(\_, \alpha_0)$ is a tuple of R in $\textsc{Sol}_{G'}^{\Delta_{\alpha'}}(J)$.

This proves $\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S) = \textsc{Sol}_{G'}^{\Delta_{\alpha'}}(\mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S)))$ and concludes the proof of Claim 7. □

Now fix arbitrary instances S of schema σ and T of schema ω. We have that $(S, T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis$ if and only if there exist actual functions F' and G' interpreting F and G, respectively, and an instance J in $\mathit{Rep}_A(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S))$ such that T is in $\mathit{Rep}_A(\textsc{Sol}_{G'}^{\Delta_{\alpha'}}(J))$. Since annotation on Σ_α is all-closed, J belongs to $\mathit{Rep}_A(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S))$ if and only if $J = \mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S))$. Therefore $(S, T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis$ if and only if there exist functions F' and G' such that T is in $\mathit{Rep}_A(\textsc{Sol}_{G'}^{\Delta_{\alpha'}} \circ \mathit{rel} \circ \textsc{Sol}_{F'}^{\Sigma_\alpha}(S))$, which by Claim 7 coincides with $\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S)$, since H' is the restriction of $F' \cup G'$ to function symbols of H.

This shows that $(S, T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis$ if and only if $T \in \mathit{Rep}_A(\textsc{Sol}_{H'}^{\Gamma_{\alpha'}}(S))$ for some H'; that is if and only if $(S, T) \in \llparenthesis \Gamma_{\alpha'} \rrparenthesis$. The conclusion is that $\Gamma_{\alpha'}$ captures the composition of Σ_α and $\Delta_{\alpha'}$ in the case that annotation on Σ_α is all-closed.

In the case that annotation on $\Delta_{\alpha'}$ is all-open and $\Delta_{\alpha'}$ contains only monotone SkSTDs the following claim shows that $\llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis = \llparenthesis \Sigma_{cl} \rrparenthesis \circ \llparenthesis \Delta_{\alpha'} \rrparenthesis$. Therefore the mapping $\Gamma_{\alpha'}$ constructed as shown above from $\Delta_{\alpha'}$ and Σ_{cl} represents also a composition mapping for $\Delta_{\alpha'}$ and Σ_α.

Claim 8. *If Δ is a mapping containing only monotone SkSTDs, Σ is an arbitrary mapping with SkSTDs, and α is an arbitrary annotation on Σ, then*

$$\llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{op} \rrparenthesis = \llparenthesis \Sigma_{cl} \rrparenthesis \circ \llparenthesis \Delta_{op} \rrparenthesis$$

Proof of Claim 8. Fix arbitrary instances S of schema σ and T of schema ω. Assume first that $(S, T) \in \llparenthesis \Sigma_{cl} \rrparenthesis \circ \llparenthesis \Delta_{op} \rrparenthesis$. Then there exists an instance $J \in \llbracket S \rrbracket^{\Sigma_{cl}}$ such that $T \in \llbracket J \rrbracket^{\Delta_{op}}$. Since $J \in \llbracket S \rrbracket^{\Sigma_{cl}}$, we have $J = \mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_{cl}})$, for some actual functions F' interpreting function symbols of Σ. But since $\mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_{cl}}) = \mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_\alpha})$, we conclude that J is also in $\llbracket S \rrbracket^{\Sigma_\alpha}$. Consequently $(S, T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{op} \rrparenthesis$.

Conversely assume $(S, T) \in \llparenthesis \Sigma_\alpha \rrparenthesis \circ \llparenthesis \Delta_{op} \rrparenthesis$. Then there exists an instance $J \in \llbracket S \rrbracket^{\Sigma_\alpha}$ such that $T \in \llbracket J \rrbracket^{\Delta_{op}}$. Since $J \in \llbracket S \rrbracket^{\Sigma_\alpha}$, it contains an subinstance $J_0 = \mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S))$, for some actual functions F' interpreting functions in Σ. Since $\mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_\alpha}(S)) = \mathit{rel}(\textsc{Sol}_{F'}^{\Sigma_{cl}}(S))$, we have $J_0 \in \llbracket S \rrbracket^{\Sigma_{cl}}$.

We now prove that T is also in $\llbracket J_0 \rrbracket^{\Delta_{op}}$. We know T is in $\mathit{Rep}_A(\textsc{Sol}_{G'}^{\Delta_{op}}(J))$ for some actual functions G' interpreting functions in Δ. Furthermore, due to the monotonicity of the SkSTDs of Δ, we have $re\ell(\text{SOL}_{G'}^{\Delta_{op}}(J)) \supseteq re\ell(\text{SOL}_{G'}^{\Delta_{op}}(J_0))$. Therefore, since T contains $re\ell(\text{SOL}_{G'}^{\Delta_{op}}(J))$, we see that T also contains $re\ell(\text{SOL}_{G'}^{\Delta_{op}}(J_0))$. Moreover, all tuples of $\text{SOL}_{G'}^{\Delta_{op}}(J_0)$ have open annotations; hence T belongs to $Rep_A(\text{SOL}_{G'}^{\Delta_{op}}(J_0))$ and thus $T \in [\![J_0]\!]^{\Delta_{op}}$. Together with the fact that $J_0 \in [\![S]\!]^{\Sigma_{cl}}$, this proves $(S,T) \in (\!|\Sigma_{cl}|\!) \circ (\!|\Delta_{op}|\!)$, and concludes the proof of Claim 8 and Lemma 5. □

As a corollary of Lemma 5, we have our main composition result:

Theorem 5. *The following two classes of schema mappings given by annotated SkSTDs are closed under composition:*

1. *mappings with all-open annotations in which only conjunctive queries are used in SkSTDs; and*
2. *mappings with all-closed annotations in which arbitrary* FO *queries are used in SkSTDs.*

Proof. Given two schema mappings Σ_α and $\Delta_{\alpha'}$ with all-open annotated CQ-SkSTDs, by Lemma 5, there exists a composition mapping $\Gamma_{\alpha'}$, whose SkSTDs have the same annotation as $\Delta_{\alpha'}$, and whose right-hand sides are CQ. Therefore $\Gamma_{\alpha'}$ contains only all-open annotated CQ-SkSTDs.

On the other hand if annotations of both Σ_α and $\Delta_{\alpha'}$ are all-closed, again by Lemma 5, there exists a composition mapping $\Gamma_{\alpha'}$ whose SkSTDs have the same annotation as $\Delta_{\alpha'}$, therefore annotation on $\Gamma_{\alpha'}$ is all-closed. □

In Theorem 5, the first case of course is that of [13]. Theorem 5 says that we can also achieve compositionality under the CWA with more general queries used in mappings.

6. Conclusions

Two previous approaches to data exchange have been based either on the OWA, or on the CWA, and both had their limitations. We have shown that, using an old idea of allowing both open and closed null values, we obtain mappings that can mix OWA and CWA in an arbitrary manner. We looked at query evaluation and composition of mappings, proved two classification results for their complexity, established criteria for schema compositionality, and showed particularly nice behaviour of positive queries in mixed contexts.

Several extensions of our results can be obtained. We mention three. The first trichotomy theorem is true for any query language of PTIME data complexity that contains FO. Second, if we allow 1-to-m relationships in place of 1-to-many relationships and define such limited open nulls (i.e. each such null can be replicated at most m times), then all the complexity results about CWA mappings apply to this case. Third, if a mapping Δ has only existential queries, then every composition $\Sigma_\alpha \circ \Delta_{\alpha'}$ is in NP, regardless of annotations.

A few open problems remain. First of all we would like to see whether Theorem 5 can be extended to other classes of annotated mappings. The next step is extending results to cover mappings with target constraints, as was done in [16]. It is likely that adding weakly acyclic constraints [11, 10] would lead to a terminating chase as in both open-world [11] and closed-world [16] cases. We also would like to see if the mixed open/closed mappings are applicable in more general frameworks that try to unify data exchange, integration, and peer-to-peer scenarios, such as in [9].

Acknowledgments The authors were supported by the EPSRC grants E005039 and F028288 and by the FET-Open grant agreement FOX, number FP7-ICT-233599; the first author also by the EU grant MEXC-CT-2005-024502.

References

[1] S. Abiteboul, O. Duschka. Complexity of answering queries using materialized views. In *PODS 1998*, pages 254–263.

[2] S. Abiteboul, P. Kanellakis, G. Grahne. On the representation and querying of sets of possible worlds. *TCS* 78 (1991), 158–187.

[3] M. Arenas, P. Barceló, R. Fagin, L. Libkin. Locally consistent transformations and query answering in data exchange. In *PODS 2004*, pages 229–240.

[4] M. Arenas, P. Barceló, J. Reutter. Query Languages for Data Exchange: Beyond Unions of Conjunctive Queries. In *ICDT 2009*, pages 73–83.

[5] P. Atzeni, N. Morfuni. Functional dependencies and constraints on null values in database relations. *Information and Control* 70(1): 1–31 (1986).

[6] P. Bernstein, T.Green, S. Melnik, A. Nash. Implementing mapping composition. *VLDB'06*, pages 55–66.

[7] P. Bernstein, S. Melnik. Model management 2.0: manipulating richer mappings. *SIGMOD'07*, pages 1–12.

[8] L. Chiticariu, W.-C. Tan. Debugging schema mappings with routes. In *VLDB'06*, pages 79–90.

[9] G. De Giacomo, D. Lembo, M. Lenzerini, R. Rosati. On reconciling data exchange, data integration, and peer data management. In *PODS'07*, pages 133–142.

[10] A. Deutsch, V. Tannen. Reformulation of XML queries and constraints. In *ICDT'03*, pages 225–241.

[11] R. Fagin, Ph. Kolaitis, R. Miller, L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.* 336(1): 89–124 (2005).

[12] R. Fagin, Ph. Kolaitis, L. Popa. Data exchange: getting to the core. *ACM TODS* 30(1): 174–210 (2005).

[13] R. Fagin, Ph. Kolaitis, L. Popa, W.C. Tan. Composing schema mappings: second-order dependencies to the rescue. *ACM TODS* 30(4) 994–1055 (2005).

[14] G. Gottlob, R. Zicari. Closed world databases opened through null values. In *VLDB'88*, pages 50–61.

[15] G. Grahne. *The Problem of Incomplete Information in Relational Databases*. Springer, 1991.

[16] A. Hernich, N. Schweikardt. CWA-solutions for data exchange settings with target dependencies. In *PODS'07*, pages 113–122.

[17] T. Imielinski, W. Lipski. Incomplete information in relational databases. *J. ACM* 31 (1984), 761–791.

[18] Ph. Kolaitis. Schema mappings, data exchange, and metadata management. In *PODS 2005*.

[19] M. Lenzerini. Data integration: a theoretical perspective. In *PODS'02*, pages 233–246.

[20] M. Levene, G. Loizou. Axiomatisation of functional dependencies in incomplete relations. *Theoretical Computer Science* 206 (1998), 283–300.

[21] L. Libkin. Data exchange and incomplete information. In *PODS'06*, pages 60–69.

[22] W. Lipski. On semantic issues connected with incomplete information in databases. *ACM Trans. Database Systems* 4 (1979), 262–296.

[23] J. Madhavan, A. Halevy. Composing mappings among data sources. In *VLDB'03*, pages 572–583.

[24] A. Madry. Data exchange: on the complexity of answering queries with inequalities. *IPL* 94 (2005) 253–257.

[25] J. Makowsky and Y. Pnueli. Arity and alternation in second-order logic. *Annals of Pure and Applied Logic*, 78 (1996), 189–202.

[26] R. Miller, M. Hernandez, L. Haas, L. Yan, C. Ho, R. Fagin, L. Popa. The Clio project: managing heterogeneity. *SIGMOD Record* 30 (2001), 78–83.

[27] A. Nash, P. Bernstein, S. Melnik. Composition of mappings given by embedded dependencies. *ACM TODS* 32(1): 4 (2007).

[28] L. Popa, Y. Velegrakis, R. Miller, M. Hernández, R. Fagin. Translating web data. In *VLDB 2002*, pages 598–609.

[29] R. Reiter. On closed world databases. In *"Logic and Databases"*, H. Gallaire and J. Minker eds, Plenum Press, 1978, pages 55–76.