On Equivalence and Cores for Incomplete Databases in Open and Closed Worlds

Henrik Forssell
University of Oslo, Norway
University of South East Norway, Drammen, Norway
jonf@ifi.uio.no

Evgeny Kharlamov
Bosch Center for Artificial Intelligence, Renningen, Germany
University of Oslo, Norway
evgeny.kharlamov@de.bosch.com

Evgenij Thorstensen
University of Oslo, Norway
evgenit@ifi.uio.no

Abstract

Data exchange heavily relies on the notion of incomplete database instances. Several semantics for such instances have been proposed and include open (OWA), closed (CWA), and open-closed (OCWA) world. For all these semantics important questions are: whether one incomplete instance semantically implies another; when two are semantically equivalent; and whether a smaller or smallest semantically equivalent instance exists. For OWA and CWA these questions are fully answered. For several variants of OCWA, however, they remain open. In this work we adress these questions for Closed Powerset semantics and the OCWA semantics of [24]. We define a new OCWA semantics, called OCWA*, in terms of homomorphic covers that subsumes both semantics, and characterize semantic implication and equivalence in terms of such covers. This characterization yields a guess-and-check algorithm to decide equivalence, and shows that the problem is NP-complete. For the minimization problem we show that for several common notions of minimality there is in general no unique minimal equivalent instance for Closed Powerset semantics, and consequently not for the more expressive OCWA* either. However, for Closed Powerset semantics we show that one can find, for any incomplete database, a unique finite set of its subinstances which are subinstances (up to renaming of nulls) of all instances semantically equivalent to the original incomplete one. We study properties of this set, and extend the analysis to OCWA*.

2012 ACM Subject Classification Theory of computation → Incomplete, inconsistent, and uncertain databases; Theory of computation

Keywords and phrases Incomplete Databases, Cores, Semantics, Open and Closed Worlds

Digital Object Identifier 10.4230/LIPIcs.ICDT.2020.10

Related Version A full version of the paper is available at `https://arxiv.org/abs/2001.04757`.

Funding This work was partially supported by the SIRIUS Centre, Norwegian Research Council project number 237898 and project number 230525.

1 Introduction

Data Exchange

Data exchange is the problem of translating information structured under a *source schema* into a *target schema*, given a source data set and a set of declarative *schema mappings* between the source and target schemata. This problem has originally been studied for traditional relational databases where a decade of intensive research brought up a number of foundational and system oriented work [2, 5, 7, 21, 26]. More recently research in data exchange changed its focus in various directions that include non-relational [4] and temporal data [13], knowledge bases [3], mapping discovery [27, 28], and probabilistic settings [19, 25].

In relational data exchange, a set of schema mappings M is defined as a set of source-to-target tuple generating dependences [1] of the form $\phi(\bar{x}, \bar{y}) \rightarrow \psi(\bar{x}, \bar{z})$, where $\phi(\bar{x}, \bar{y})$ (resp. $\psi(\bar{x}, \bar{z})$) is a query over the source (resp. target) schema with its variables. In general such mappings only partially specify how to populate attributes of the target schema with data from a given *source instance* [8], i.e., due to existential variables $\bar{z}$ in $\psi(\bar{x}, \bar{z})$. Therefore, data exchange can result in possibly multiple incomplete *target instances A*. Each such A represents a set of possible complete target instances and there are several options on how such correspondence, or *semantics of incomplete instances*, can be defined, including Open World (OWA) [8, 9], Closed World (CWA) [18], Open and Closed World (OCWA, with annotated instances) [24], and Powerset Closed World (PCWA) [12], which we discuss in detail in Section 2.

Problems for Data Exchange

In the context of data exchange the following questions have attracted considerable attention: given a semantics for incomplete database instances, decide:
- [**Semantic Implication:**] whether one incomplete instance semantically implies another;
- [**Equivalence:**] whether two incomplete instances are semantically equivalent; and
- [**Minimality, Core:**] whether a smaller or smallest (core) semantically equivalent incomplete instance exists.

These questions form a natural progression, in that a characterization of semantic implication leads to one for equivalence, which in turn allows the study of minimal equivalent instances. The latter is important since, e.g., in some cases one can use the smallest minimal instance for computing certain answers by naively' evaluating queries directly on this instance.

How These Problems Have Been Addressed So Far

These three questions are the focus of this paper since they have only partially been answered. Indeed, for OWA and CWA, these questions have been fully answered. For OWA, semantic implication corresponds to the existence of a database homomorphism from one instance into another, and a unique smallest equivalent instance (the core [9]) always exists, and is minimal for several natural notions of minimality. Likewise, for CWA semantic implication corresponds to the existence of a strongly surjective homomorphism from one instance to another [18]. This implies that equivalence corresponds to isomorphism, rendering the question of smallest equivalent instance moot. For PCWA, semantic implication corresponds to the existence of a *homomorphic cover* from one instance to another [12], while the question of smallest equivalent instance remains open. For OCWA with annotated instances, both questions are open, although preliminary results were previously presented by the authors [11]. Finally, we are not aware that the question of semantic implication between PCWA and OCWA with annotated instances has previously been considered.

Our Approach to Implication and Equivalence

Therefore, in this paper we address the questions of Semantic Implication, Equivalence, and Minimality for PCWA and OCWA semantics. To this end we introduce a novel open-and-closed-world semantics, OCWA*, based purely on the notion of homomorphic cover. We show how both PCWA and OCWA semantics with annotated instances can be defined as special cases of OCWA*. This subsumption property allows us to characterize semantic implication and equivalence for all three semantics using homomorphic covers, and thus also semantic implication and equivalence between PCWA and OCWA with annotated instances.

Our Approach to Minimality and Cores

We study several natural notions of minimality, and show for all of them that there is in general no unique minimal equivalent instance for PCWA nor, consequently, for the more expressive OCWA*. This raises the question: *How can one find a smaller or "better" equivalent instance?* Indeed, even if one can find all equivalent subinstances of a given incomplete instance A and compare them using the characterization of equivalence, one still does not know whether a there exists a smaller equivalent instance that is not a subinstance of A.

We address this challenge as follows. Focusing first on PCWA, we show that for all instances A there exists a finite set $\varpi(A)$ of "PCWA-cores" which serves to determine all minimal instances that are equivalent to A. More precisely, this set has the following properties:

1. each member of $\varpi(A)$ is minimal (for all notions of minimality that we consider in this paper) and a subinstance of A,
2. the union of the members of $\varpi(A)$ is equivalent to A,
3. A and B are equivalent if and only if $\varpi(A) = \varpi(B)$, up to renaming of nulls, and
4. any instance which is equivalent to A and which is minimal in the sense of having no equivalent subinstance must be an image of the union of the members of $\varpi(A)$. In particular, all such instances can be found, up to renaming of nulls, from (the union of) $\varpi(A)$.

We also apply the analysis of naïve evaluation of existential positive queries with Boolean universal guards from [12] and show that such queries can be evaluated on the smaller members in $\varpi(A)$ rather than on A itself. Finally, we extend the analysis to OCWA* and show that, by resolving a question of "redundant annotation", the function $\varpi(A)$ can be extended also to annotated instances, yielding similar properties for OCWA*. In summary, the contributions of this paper are:

- A new semantics OCWA* which properly extends PCWA and OCWA with annotated instances.
- Characterization and analysis of semantic implication and equivalence for PCWA, OCWA with annotated nulls, and OCWA*.
- Negative results for the existence of unique minimal instances in PCWA and OCWA*.
- A new concept of "PCWA-core" for PCWA; and in terms of it,
- a new "powerset canonical representative function" $\varpi(-)$ for PCWA and OCWA*, with the properties listed above.
- An analysis of "annotation redundancy" in OCWA*.

The paper is organised as follows. In Section 2 we give preliminaries and introduce known semantics for incomplete DBs. In Section 3 we present our OCWA* semantics and give its basic properties. In Section 4 we study semantic implication and equivalence for OCWA*. In Section 5 we show the non-existence of a subinstance minimal representative function for PCWA and, consequently, for OCWA*. In Section 6 we move to positive results for PCWA and then extend them in Section 7 for the general case of OCWA*.

2 Preliminaries

We use boldface for lists and tuples; thus $\mathbf{x}$ instead of $\bar{x}$ or $\overrightarrow{x}$. $\mathbb{N}^+$ is the set of positive (non-zero) natural numbers. $\mathcal{P}^+(A)$ is the set of non-empty subsets of A. $\mathcal{P}^{\text{fin}}(A)$ is the set of finite subsets of A. If S is a set of instances then $\overline{S}^{\cup}$ denotes the closure of S under binary unions.

2.1 Incomplete Databases

We assume that we are working with a fixed database schema. Let Const and Null be countable sets of constants and labeled nulls. For the sake of readability, we will use lower case letters late in the alphabet for nulls instead of the more common $\perp$. Lower case letters a, b, c, d will be used for constants. An (incomplete) *instance* A is a database instance whose (active) domain is a subset of $\mathsf{Const} \cup \mathsf{Null}$. A *complete instance* I is an instance without nulls. (This is also known as a *ground* instance.) We write $\mathcal{D}$ for the set of all instances and $\mathcal{C}$ for the set of all complete instances. We use upper case letters A, B, etc. from the beginning of the alphabet for instances in general, and upper case letters I, J, etc. from the middle of the alphabet for instances that are explicitly assumed to be complete.

Following [24] an *annotated instance* is an instance where each *occurrence* of a constant or null is annotated with either o, standing for *open*, or c, standing for *closed*. The added expressivity is used to define more fine-grained semantics for incomplete databases.

2.2 Homomorphisms and Disjoint Unions

We use the terms "homomorphism" and "isomorphism" to mean database homomorphism and database isomorphism, respectively, and we distinguish these from "structure" homomorphisms. Explicitly, if A and B are instances – whether incomplete or complete, annotated or not – a *structure homomorphism* $h : A \to B$ is a function from the active domain of A to the active domain of B such that for every relation symbol R, if a tuple $\mathbf{u}$ is in the relation R in A then the tuple $h(\mathbf{u})$ is in the relation R in B. We write $\mathsf{Str}(A, B)$ for the set of structure homomorphisms from A to B. A *structure isomorphism* is an invertible structure homomorphism.

If $P \subseteq \mathsf{Const} \cup \mathsf{Null}$ and h is a structure homomorphism we say that *h fixes P pointwise* if $h(p) = p$ for all $p \in P$ on which h is defined. We say that *h fixes P setwise* if it restricts to a bijection on the subset of P on which it is defined.

A *(database) homomorphism* from A to B is a structure homomorphism that fixes Const pointwise. We write $\mathsf{Hom}(A, B)$ for the set of homomorphisms from A to B. A *(database) isomorphism* is an invertible homomorphism.

A *subinstance* of A is an instance B with an inclusion homomorphism $B \hookrightarrow A$ – that is, with a homomorphism that fixes $\mathsf{Const} \cup \mathsf{Null}$ pointwise. B is a *proper* subinstance if $A \neq B$. We shall often be somewhat lax with the notion of a subinstance and regard B as a subinstance if it is so up to renaming of nulls, that is to say, up to (database) isomorphism. If we need to insist that the homomorphism $B \hookrightarrow A$ is an inclusion we say that B is a *strict* subinstance.

If $h : A \to B$ is a structure homomorphism then the *image* $h(A)$ of h is the subinstance of B defined by the condition that $\mathbf{v}$ is in the relation R in $h(A)$ if there exists $\mathbf{u}$ in R in A so that $h(\mathbf{u}) = \mathbf{v}$. If $h(A) = B$ we say that h is *strongly surjective* and write $h : A \twoheadrightarrow B$. If h is not a structure isomorphism we say that $h(A)$ is a *proper* image.

A *reflective subinstance* of A is an instance B with an inclusion homomorphism $m : B \hookrightarrow A$ and a strongly surjective homomorphism $q : A \twoheadrightarrow B$ such that $q \circ m$ is the identity on B. Again, we often say that B is a reflective subinstance if it is so up to renaming of nulls, and say that it is a strict reflective subinstance if we want to insist that m is an inclusion, rather than just an injective homomorphism.

If $H = \{h_i : A \to B \mid i \in S\}$ is a family of homomorphisms we say that H is a *covering family*, or simply a *cover*, if $B = \bigcup_{i \in S} h_i(A)$. We say that A *covers* B if $\mathsf{Hom}(A, B)$ is a cover. If $H = \{h_i : A_i \to B \mid i \in S\}$ is a family of homomorphisms with the same codomain we say that H *jointly covers* B if $B = \bigcup_{i \in S} h_i(A_i)$

If A is an incomplete instance, a *freeze* of A is, as usual, a complete instance $\overline{A}$ together with a structure isomorphism between A and $\overline{A}$ that fixes the constants in A. Whenever we take a freeze of an instance, we tacitly assume that it is "fresh", in the sense that the new constants in it do not occur in any other instances currently under consideration (that is, usually, that have been introduced so far in the proof).

We define the *null-disjoint* union $A \sqcup_{\mathsf{Null}} B$ of two instances A and B to be the instance obtained by renaming whatever nulls necessary to make sure that A and B have no nulls in common, and then taking the union of the result. As such, the null-disjoint union is only defined up to isomorphism. The key property of the null-disjoint union is the 1-1 correspondence $\mathsf{Hom}(A \sqcup_{\mathsf{Null}} B, C) \cong \mathsf{Hom}(A, C) \times \mathsf{Hom}(B, C)$ between homomorphisms from $A \sqcup_{\mathsf{Null}} B$ and pairs of homomorphisms from A and B.

The definition extends to n-ary and infinitary null-disjoint unions. (Infinitary null-disjoint unions are, strictly speaking, not database instances in so far as they are not finite, but they are an occasionally useful technical extrapolation, and we trust that they will cause no confusion in the places where we make use of them.) We shall mostly be considering the null-disjoint union of an instance with itself. For $n \in \mathbb{N}^+ \cup \{\infty\}$, we abuse notation and simply write A^n for the null-disjoint union of A with itself n times, with the property that $\mathsf{Hom}(A^n, C) \cong \prod_{i=1}^n \mathsf{Hom}(A, C)$. We denote by $\pi_m : A \to A^n$, for $m \in \mathbb{N}^+$ smaller or equal to n, the homomorphism that sends A to the mth copy of it in A^n. If $f : A^n \to C$ is a homomorphism we write $f = \langle f_1, \ldots, f_n \rangle$ where $f_i = f \circ \pi_i : A \to C$. We denote by $\nabla : A^n \to A$ the homomorphism that corresponds to the n-tuple of identity homomorphisms $A \to A$. That is to say, ∇ identifies all copies in A^n of a null in A with that null.

2.3 Semantics of Incomplete Databases

A *semantics* is a function $[\![-]\!] : \mathcal{D} \to \mathcal{P}^+(\mathcal{C})$ which assigns a non-empty set $[\![A]\!]$ of complete instances to every instance A. We say that A *represents* $[\![A]\!]$.

A semantics $[\![-]\!]$ induces a preordering on $\mathcal{D}$ by $A \leq B \Leftrightarrow [\![A]\!] \subseteq [\![B]\!]$[1]. We say that A and B are *semantically equivalent*, and write $A \equiv B$, if $A \leq B$ and $B \leq A$. Accordingly, $A \equiv B \Leftrightarrow [\![A]\!] = [\![B]\!]$. The semantic equivalence class of an instance is denoted using square brackets: $[A] := \{B \in \mathcal{D} \mid A \equiv B\}$.

A *representative function* (*cf. representative set, canonical function* in [12]) is a function $\chi : \mathcal{D} \to \mathcal{D}$ which picks a representative of each semantic equivalence class. We shall be content with $\chi(A)$ being defined up to isomorphism. A representative function χ is *subinstance minimal* if $\chi(A)$ is a subinstance of all members of $[A]$.

Next, we briefly recall the established semantics OWA, CWA, the Closed Powerset semantics of [12], and the Open and Closed World Assumption as defined by Libkin and Sirangelo [24].

[1] Note that this is the opposite of the standard order as defined in e.g. [12]

2.3.1 Open World Approach: OWA

Under OWA (Open World Assumption) an instance A represents the set of complete instances to which A has a (database) homomorphism; $[\![A]\!]_{\mathsf{OWA}} = \{I \in \mathcal{C} \mid \mathsf{Hom}(A, I) \neq \emptyset\}$.

Consequently, $[\![A]\!]_{\mathsf{OWA}}$ is closed under structure homomorphisms that fix the constants in A pointwise, in the sense that if $I \in [\![A]\!]_{\mathsf{OWA}}$ and $I \to J$ is a structure homomorphism that fixes the constants in A, then $J \in [\![A]\!]_{\mathsf{OWA}}$. It is well known (see e.g. [9]) that the function $\mathsf{Core}(-)$ that maps each instance to its core is a subinstance minimal representative function.

2.3.2 Closed World Approach: CWA

Under CWA (Closed World Assumption) an instance A represesents the set of its images; $[\![A]\!]_{\mathsf{CWA}} = \{I \in \mathcal{C} \mid \text{there exists } h : A \twoheadrightarrow I\}$

Note that $[\![A]\!]_{\mathsf{CWA}}$ is closed under strongly surjective structure homomorphisms that fix the constants in A pointwise. Clearly, the only possible representative function (up to isomorphism, as usual) is the identity.

2.3.3 Closed Powerset: PCWA

Under *Closed Powerset semantics* (PCWA) [12], A represents the set of its CWA-interpretations closed under union; $[\![A]\!]_{\mathsf{PCWA}} = \overline{[\![A]\!]_{\mathsf{CWA}}}^{\cup} = \{I_1 \cup \ldots \cup I_n | n \in \mathbb{N}^+,\ I_1, \ldots, I_n \in [\![A]\!]_{\mathsf{CWA}}\}$ Consequently, $[\![A]\!]_{\mathsf{PCWA}}$ is closed under unions and under strongly surjective homomorphisms that fix the constants in A pointwise. Note that in [12] this semantics is denoted $(|A|)_{\mathsf{CWA}}$ We recall the following from [12, Thm 10.1];

▶ **Proposition 1.** *$A \leq_{\mathsf{PCWA}} B$ iff there exists a cover from B to A.*

Thus, $A \equiv_{\mathsf{PCWA}} B$ iff there exists a cover both from B to A and from A to B. The existence of minimal representative functions for PCWA is the subject of Section 5 and 6.

▶ Remark 2. The semantics GCWA introduced in [17] defines $[\![A]\!]_{\mathsf{GCWA}}$ as the set of unions of *minimal* images of A. In [12] $[\![A]\!]_{\mathsf{GCWA}}$ is denoted by $(|A|)^{\min}_{\mathsf{CWA}}$. As with OWA, $\mathsf{Core}(-)$ is a minimal representative function for GCWA (see [12])

$[\![A]\!]_{\mathsf{GCWA}}$ is not in general closed under strong surjections preserving the constants in A (cf. [12, 9.1]). It therefore cannot be represented in the semantics introduced in Section 3 below.

2.3.4 Mixed Approach: OCWA$^{\mathrm{LS}}$

Let A be an annotated instance, i.e. such that each occurrence of a constant or null is annotated as open or closed. Under OCWA$^{\mathrm{LS}}$ (Open and Closed World Assumption - Libkin/Sirangelo) the set of complete instances represented by A is defined in two steps as follows [24]: for all complete instances I, $I \in [\![A]\!]_{\mathsf{OCWA}^{\mathrm{LS}}}$ if

(i) there exists a homomorphism $h : A \to I$; and

(ii) for every $R(\mathbf{t})$ in I there exists a $R(\mathbf{t}')$ in A such that $h(\mathbf{t}')$ and $\mathbf{t}$ agree on all positions annotated as closed in $\mathbf{t}'$.

OCWA^{LS} is subsumed by a more expressive semantics which we define next.

3 Our Semantics: OCWA*

In this section we propose the semantics OCWA* for annotated instances as a properly more expressive version of both OCWA$^{\mathrm{LS}}$and PCWA. The semantics OCWA* presupposes that instances are annotated according to certain conditions, which we define first:

▶ **Definition 3.** *We say that an annotated instance A is in* normal form *if:*

1. *all occurrences of constants in A are annotated as closed; and*
2. *all occurrences in A of a null agree on the annotation of that null.*

The following then allows us to restrict attention to instances in normal form without loss of generality with respect to OCWA^{LS}.

▶ **Proposition 4.** *Let A be an annotated instance. Then there exists an annotated instance A' in normal form such that $[\![A]\!]_{\mathsf{OCWA^{LS}}} = [\![A']\!]_{\mathsf{OCWA^{LS}}}$.*

Proof. For any atoms that contain open constants or open nulls annotated as closed elsewhere, change the annotation to "closed" and add a copy of the atom where those terms are replaced by fresh open nulls. ◀

▶ **Definition 5.** *If A is a normal form annotated instance and B is an instance, an* RCN-cover *$H : A \overset{\mathrm{RCN}}{\rightrightarrows} B$ is a set $H \subseteq \mathsf{Hom}(A, B)$ such that the homomorphisms in H are jointly strongly surjective and agree on the closed nulls of A.*

▶ **Definition 6** (OCWA*)**.** *Let A be a annotated instance in normal form. Then A represents those complete instances under OCWA* that it RCN-covers; $[\![A]\!]_{\mathsf{OCWA^*}} = \{I \in \mathcal{C} \,|\, \exists H : A \overset{\mathrm{RCN}}{\rightrightarrows} I\}$.*

▶ Remark 7. The definition of $[\![A]\!]_{\mathsf{OCWA^*}}$ could equivalently be given as the set of finite unions $h_1(A) \cup \ldots \cup h_n(A)$ of complete images of A such that the homomorphisms $h_1, \ldots, h_n$ agree on the closed nulls of A. Thus OCWA* lies within what [12] call *Powerset semantics*; that is, semantics that are defined in terms of a relation from instances to sets of complete instances (certain finite sets of valuations, in this case) and a relation from sets of complete instances to complete instances (unions, in this case).

OCWA* properly extends OCWA$^{\mathrm{LS}}$in the following sense:

▶ **Theorem 8.** **1.** *For every normal form annotated instance A one can compute in time linear in $|A|$ a normal form annotated instance A' such that $[\![A]\!]_{\mathsf{OCWA^{LS}}} = [\![A']\!]_{\mathsf{OCWA^{LS}}} = [\![A']\!]_{\mathsf{OCWA^*}}$.*

2. *There is a normal form annotated instance A such that for every A' it holds that $[\![A]\!]_{\mathsf{OCWA^*}} \neq [\![A']\!]_{\mathsf{OCWA^{LS}}}$.*

Proof. (1) Given A, extend it to a new instance A' by: for each atom $R(\mathbf{t})$ in A add an atom $R(\mathbf{t}')$ where $\mathbf{t}'$ has every *occurrence* of an open null in $\mathbf{t}$ replaced by a fresh open null. It is then straightforwardly verified that $[\![A]\!]_{\mathsf{OCWA^{LS}}} = [\![A']\!]_{\mathsf{OCWA^{LS}}} = [\![A']\!]_{\mathsf{OCWA^*}}$.

(2) Consider the annotated instance $A = \{R(a^c, x^o, x^o)\}$. The instances in $[\![A]\!]_{\mathsf{OCWA^*}}$ contain only tuples where the second and third coordinate are equal. However, the definition of OCWA^{LS} requires only that one tuple in each instance from $[\![A]\!]_{\mathsf{OCWA^{LS}}}$ respects this equality. Since there is no bound on the size of instances in $[\![A]\!]_{\mathsf{OCWA^*}}$, there is no A' such that $[\![A']\!]_{\mathsf{OCWA^{LS}}} = [\![A]\!]_{\mathsf{OCWA^*}}$. ◀

Regarding PCWA, if A is a normal form annotated instance without any closed nulls, then an RCN-cover $A \overset{\text{RCN}}{\rightrightarrows} C$ is simply a cover, since there are no closed nulls to agree upon. Thus PCWA is OCWA* restricted to instances without closed nulls. Explicitly, let A be an un-annotated instance, and let its *canonical annotation* be that which annotates each constant as closed and each null as open. Then we have:

▶ **Proposition 9.** *Let A be an (un-annotated) instance and let $A[]$ be that instance with canonical annotation. Then $[\![A[]]\!]_{\mathsf{OCWA}^*} = [\![A]\!]_{\mathsf{PCWA}}$.*

For the rest of this paper we assume that all annotated instances are in normal form. This allows us to introduce some notational conventions that simplify the study of RCN-covers on such instances. We also switch to annotating nulls by using lower and upper case instead of superscripts, since this allows us to more clearly emphasize the distinguished status of the closed nulls. We introduce the following conventions:

– Open nulls are written in lower case, x, y, z. Closed nulls are written in upper case, X, Y, Z. (All instances are in normal form, so no null may occur both in lower and upper case in an instance.)

– We display the closed nulls of an instance together with the instance; so that $A[\mathbf{X}]$ is an annotated instance where $\mathbf{X}$ is a listing of the closed nulls of the instance. Thus $\mathbf{X}$ can be the empty list. We allow ourselves to treat $\mathbf{X}$ as the set of closed nulls of A when convenient. It is a list for purposes of substitution. In particular:

– If $\mathbf{t}$ is a list of constants or nulls, $A[\mathbf{t}/\mathbf{X}]$ is the instance obtained by replacing $\mathbf{X}$ with $\mathbf{t}$. If clear from context, we use $A[\mathbf{t}]$ as shorthand.

– Let $n \in \mathbb{N}^+ \cup \{\infty\}$. Recall from Section 2.2 that we, for an un-annotated instance A, write A^n as a shorthand for the n-ary null-disjoint union of A with itself. For an annotated instance $A[\mathbf{X}]$ with closed nulls $\mathbf{X}$, we extend this notation and write $A^n[\mathbf{X}]$ for the n-ary *open-null-disjoint* union; that is, the result of taking the union of n copies of $A[\mathbf{X}]$ where the *open* nulls have been renamed so that no two copies have any open nulls in common. Accordingly, a homomorphism $A^n[\mathbf{X}] \to C$ corresponds to an n-tuple of homomorphisms $A[\mathbf{X}] \to C$ that agree on the closed nulls $\mathbf{X}$.

We close this section by displaying some equivalent definitions of $[\![A[\mathbf{X}]]\!]_{\mathsf{OCWA}^*}$, including in terms of CWA and PCWA, which will be made use of in the sequel. Note that for $n \in \mathbb{N}^+ \cup \{\infty\}$, the family $\{\pi_m : A[\mathbf{X}] \to A^n[\mathbf{X}] \mid m \leq n,\ m \in \mathbb{N}^+\}$ forms a RCN cover from $A[\mathbf{X}]$ to $A^n[\mathbf{X}]$.

▶ **Theorem 10.** *Let $A[\mathbf{X}]$ be an annotated instance and I a complete instance. The following are equivalent:*

1. *$I \in [\![A[\mathbf{X}]]\!]_{\mathsf{OCWA}^*}$, i.e there exist an RCN-cover $A[\mathbf{X}] \overset{\text{RCN}}{\rightrightarrows} I$;*
2. *$I \in \bigcup_{n \in \mathbb{N}^+} [\![A^n[\mathbf{X}]]\!]_{\mathsf{CWA}}$;*
3. *$I \in [\![A^\infty[\mathbf{X}]]\!]_{\mathsf{CWA}}$; and*
4. *$I \in \bigcup_{\mathbf{d} \in \mathsf{Const}^k} [\![A[\mathbf{d}/\mathbf{X}]]\!]_{\mathsf{PCWA}}$ where k is the length of $\mathbf{X}$.*

▶ **Corollary 11.** *$[\![A[\mathbf{X}]]\!]_{\mathsf{OCWA}^*}$ is closed under strongly surjective structure homomorphisms that fix the constants in $A[\mathbf{X}]$ pointwise.*

We now proceed to the study of implication and equivalence OCWA*.

4 OCWA*: Implication, Equivalence

Since RCN-covers are closed under left composition with strong surjections, we have (by Theorem 10) that $[\![A[\mathbf{X}]]\!] \subseteq [\![B[\mathbf{Y}]]\!]$ iff there is an RCN-cover from $B[\mathbf{Y}]$ to $A^n[\mathbf{X}]$, for all $n \in \mathbb{N}^+$, or, equivalently, that there is an RCN-cover from $B[\mathbf{Y}]$ to $A^\infty[\mathbf{X}]$. We display this and show that n can be bounded by a number depending on B, or indeed that n can be bounded by 2 if one considers RCN-covers of a particular form. Note that the following theorem can also be applied to OCWA^{LS} via the translations of Proposition 4 and Theorem 8.

▶ **Theorem 12.** *Let $A[\mathbf{X}]$ and $B[\mathbf{Y}]$ be annotated instances. The following are equivalent:*
- **(i)** $[\![A[\mathbf{X}]]\!]_{\mathsf{OCWA}^*} \subseteq [\![B[\mathbf{Y}]]\!]_{\mathsf{OCWA}^*}$.
- **(ii)** *There is an RCN-cover from $B[\mathbf{Y}]$ to $A^n[\mathbf{X}]$, for all $n \in \mathbb{N}^+$.*
- **(iii)** *There is an RCN-cover from $B[\mathbf{Y}]$ to $A^\infty[\mathbf{X}]$*
- **(iv)** *There is a strongly surjective homomorphism from $B^\infty[\mathbf{Y}]$ to $A^\infty[\mathbf{X}]$.*
- **(v)** *There is an RCN-cover from $B[\mathbf{Y}]$ to $A^{n+1}[\mathbf{X}]$ where n is the number of closed nulls in $B[\mathbf{Y}]$, i.e. the length of $\mathbf{Y}$.*
- **(vi)** *There exists a RCN-cover H from $B[\mathbf{Y}]$ to $A^2[\mathbf{X}]$ such that H contains at least one homomorphism h which factors through $\pi_1 : A[\mathbf{X}] \to A^2[\mathbf{X}]$.*

Proof. v⇒vi : Let n be the length of $\mathbf{Y}$, and let H be an RCN-cover from $B[\mathbf{Y}]$ to $A^{n+1}[\mathbf{X}]$. Choose an h in H. There are more copies of $A[\mathbf{X}]$ in $A^{n+1}[\mathbf{X}]$ than there are $\mathbf{Y}$s, so we can assume that for all closed nulls Y_i in $B[\mathbf{Y}]$, if $h(Y_i)$ is in the $n+1$th copy, then $h(Y_i)$ is either a closed null or a constant. Then the composite $h' = \langle \pi_1, \dots, \pi_n, \pi_n \rangle \circ h : B[\mathbf{Y}] \to A^{n+1}[\mathbf{X}] \to A^{n+1}[\mathbf{X}]$ agrees with H on all $\mathbf{Y}$, so $H' = H \cup \{h'\}$ is an RCN-cover. Now, if we compose H' with the strong surjection $\langle \pi_1, \dots, \pi_1, \pi_2 \rangle : A^{n+1}[\mathbf{X}] \to A^2[\mathbf{X}]$ which sends the n first copies of $A^{n+1}[\mathbf{X}]$ to the first copy in $A^2[\mathbf{X}]$ and the $n+1$th copy of $A^{n+1}[\mathbf{X}]$ to the second in $A^2[\mathbf{X}]$, we obtain an RCN-cover of $A[\mathbf{X}] \to A^2[\mathbf{X}]$ in which the map $\langle \pi_1, \dots, \pi_1, \pi_2 \rangle \circ h'$ factors through $\pi_1 : A[\mathbf{X}] \to A^2[\mathbf{X}]$.

vi⇒ii: Let n be given, and let H be an RCN-cover from $B[\mathbf{Y}]$ to $A^2[\mathbf{X}]$ such that $h \in H$ factors through $\pi_1 : A[\mathbf{X}] \to A^2[\mathbf{X}]$. For $1 \le i \le n$, $\pi_1 : A[\mathbf{X}] \to A^n[\mathbf{X}]$ and $\pi_i : A \to A^n[\mathbf{X}]$ is a pair of homomorphisms that agree on closed nulls, so correspond to a homomorphism $\langle \pi_1, \pi_i \rangle : A^2[\mathbf{X}] \to A^n[\mathbf{X}]$. The family $\{\langle \pi_1, \pi_i \rangle \mid 1 \le i \le n\}$ of such homomorphisms is an RCN-cover from $A^2[\mathbf{X}]$ to $A^n[\mathbf{X}]$. The composite of this cover with H is RCN, since for any closed null Y_i in $B[\mathbf{Y}]$, $h' \in H$, $1 \le i \le n$, we have that $\langle \pi_1, \pi_i \rangle (h'(Y_i)) = \langle \pi_1, \pi_i \rangle (h(Y_i)) = \pi_1(h(Y_i))$.

The remaining implications are straightforward. ◀

From Theorem 12 we can derive two guess-and-check algorithms to decide containment between annotated instances. On the one hand, we may construct $A^{n+1}[\mathbf{X}]$, where n is the length of $\mathbf{Y}$, guess a set of homomorphisms from $B[\mathbf{Y}]$ to this instance, and check that it is an RCN-cover. Alternatively, we may avoid this blowup of $A[\mathbf{X}]$ by constructing $A^2[\mathbf{X}]$, guessing a homomorphism h from $B[\mathbf{Y}]$ to $A[\mathbf{X}]$ as well as a set of homomorphisms H from $B[\mathbf{Y}]$ to $A^2[\mathbf{X}]$, and checking that $\{h\} \cup H$ is an RCN-cover.

Complexity analysis

Since the instance $A^{n+1}[\mathbf{X}]$ has size at most $|A[\mathbf{X}]| \times (|\mathbf{Y}| + 1)$, and the number of homomorphisms in any non-redundant cover is bounded by the number of tuples in the target instance, the complexity of this problem stays in NP. For NP-hardness, we adapt the reduction of 3-colourability for graphs to the problem of deciding whether a given graph has a homomorphism into K_3, the complete graph on three vertices. It is easy to see that any homomorphism from a graph with at least one edge into K_3 extends to a cover of K_3. Therefore, the problem of deciding $\leq_{\mathsf{PCWA}}$, and consequently $\leq_{\mathsf{OCWA*}}$, is likewise NP-complete. It follows that the problem of deciding, given two instances A and B, whether A is a minimal equivalent instance for B given a partial order among instances, belongs to the class DP, as it involves checking the non-existence of a smaller instance. In other words, deciding semantic implication and equivalence for annotated instances has the same complexity as the homomorphism problem.

5 Issues with Minimality in OCWA*

In this section and the next we study the notion of OCWA* semantic equivalence and the question of whether, or to what extent, there exists a unique "best" annotated instance to choose among those that are semantically equivalent. For motivation and illustration, we first recall the situation in OWA in some more detail. It is well known that $A \equiv_{\mathsf{OWA}} B$ if and only if A and B are "homomorphically equivalent", that is, if there exists a homomorphism both from A to B and from B to A. Furthermore, there is, up to isomorphism, a least subinstance of A to which it is homomorphically equivalent, known as the *core* of A. Instances A and B are homomorphically equivalent if and only if their cores are isomorphic. Moreover, as a consequence of being the least homomorphically equivalent subinstance of A, the core of A is also the least reflective subinstance of A, and the least homomorphically equivalent image of A. Thus there are three quite natural notions of minimality to which the core is the answer in OWA. We say that an instance is *a core* if it is its own core, i.e. if it has no homomorphically equivalent subinstances. Cores can be characterized as those instances C with the property that any homomorphism $C \to C$ must be an isomorphism. (See [9, 10, 16] for more about cores.)

We show now that for OCWA* there does not in general exist least semantically equivalent instances in any of the three senses above. We then turn to the question of whether a "good" representative function can nevertheless be found, first for PCWA and then for OCWA* in general. We begin by fixing some terminology.

▶ **Definition 13.** *Let A and B be instances. In the context of a given semantics, we say that:*

1. *B is* sub-minimal *(subinstance minimal) if there are no proper semantically equivalent subinstances of B;*
2. *B is* rfl-minimal *(reflective subinstance minimal) if there are no proper semantically equivalent reflective subinstances of B;*
3. *B is a* least *semantically equivalent (reflective) subinstance of A if $B \equiv A$ and B is a (reflective) subinstance of all semantically equivalent (reflective) subinstances of A;*
4. *B is* img-minimal *(image minimal) if there are no proper semantically equivalent images of B, and finally;*
5. *B is a* least *semantically equivalent image of A if $B \equiv A$ and for all semantically equivalent images C of A, B is an image of C.*

We show by the examples that follow that in PCWA, and hence in OCWA*, least semantically equivalent subinstances, reflective subinstances, and images do not in general exist, and that when they do, they need not coincide. In the examples all instances consist of nulls only.

▶ **Example 14.** B_1 and C_1 are non-isomorphic PCWA-equivalent reflective subinstances of A_1. Both B_1 and C_1 are sub-minimal and rfl-minimal.

A_1	R		
	x	x	y
	x	x	x
	v	w	w
	v	v	v
	z	z	r
	z	s	s
	z	z	z

B_1	R		
	x	x	y
	x	x	x
	v	w	w
	v	v	v

C_1	R		
	z	z	r
	z	s	s
	z	z	z

▶ **Example 15.** The instances B_2 and C_2 are non-isomorphic PCWA-equivalent images of the instance A_2. Both B_2 and C_2 are img-minimal.

A_2	R				
	x	x	u	y	z
	x	x	x	x	z
	x	x	x	y	x
	x	x	x	x	x
	v	p	p	r	s
	p	p	p	p	s
	p	p	p	r	p
	p	p	p	p	p

B_2	R				
	p	p	u	r	z
	p	p	p	p	z
	v	p	p	r	s
	p	p	p	p	s
	p	p	p	r	p
	p	p	p	p	p

C_2	R				
	p	p	u	y	s
	p	p	p	y	p
	v	p	p	r	s
	p	p	p	p	s
	p	p	p	r	p
	p	p	p	p	p

▶ **Example 16.** The instance A_3 has a least PCWA-equivalent reflective subinstance, a least PCWA-equivalent subinstance, and a least PCWA-equivalent image, consisting of the non-isomorphic instances A_3, B_3, and C_3, respectively:

A_3	R				
	x	x'	y	y	z
	v'	v	s	t	s
	x	v	u	u	u
	x	x	x	x	x
	v	v	v	v	v

B_3	R				
	x	x'	y	y	z
	v'	v	s	t	s
	x	x	x	x	x
	v	v	v	v	v

C_3	R				
	w	x'	y	y	z
	v'	w	s	t	s
	w	w	w	w	w

We summarize:

▶ **Theorem 17.** *In PCWA (OCWA*),*

1. *there exists an (annotated) instance A for which there exists two non-isomorphic semantically equivalent sub-minimal subinstances;*
2. *there exists an (annotated) instance A for which there exists two non-isomorphic semantically equivalent img-minimal images; and*
3. *there exists an (annotated) instance A for which there exists two non-isomorphic semantically equivalent rfl-minimal reflective subinstances.*

6 Minimality in PCWA

Recall from Section 2.3 that a representative function for a given semantics is a function $\chi : \mathcal{D} \to \mathcal{D}$ which chooses a representative for each equivalence class. That is to say, $A \equiv B \Leftrightarrow \chi(A) = \chi(B)$, for all $A, B \in \mathcal{D}$, and $\chi(A) \equiv A$, for all $A \in \mathcal{D}$. Again, we only require that $\chi(A)$ is defined up to isomorphism, i.e. up to renaming of nulls. Recall further that a representative function is subinstance minimal if $\chi(A)$ is a subinstance of A (up to isomorphism) for all $A \in \mathcal{D}$. Similarly, we say that a representative function is *image minimal* if $\chi(A)$ is an image of A, and that it is *reflective subinstance minimal* if $\chi(A)$ is a reflective subinstance of A. The canonical example is the Core function, which is a minimal representative function for OWA in all of these three senses.

Theorem 17 showed that there can be no minimal representative function for PCWA, for any of these three senses of "minimal". However, we show that there is a function $\varpi(-) : \mathcal{D} \to \mathcal{P}^{\text{fin}}(\mathcal{D})$ that assigns a finite set $\{E_1, \ldots, E_n\}$ to each instance A that is representative in the sense that $A \equiv_{\mathsf{PCWA}} B \Leftrightarrow \varpi(A) = \varpi(B)$, for all $A, B \in \mathcal{D}$, and $\bigcup_{E \in \varpi(A)} E \equiv_{\mathsf{PCWA}} A$, for all $A \in \mathcal{A}$; and "minimal" in the sense that

- E is a reflective subinstance of A, for all $A \in \mathcal{D}$ and all $E \in \varpi(A)$, and
- E is semantically minimal in the strong sense that if $C \equiv_{\mathsf{PCWA}} E$ then E is a reflective subinstance of C, for all $E \in \varpi(A)$.

Thus the members of $\varpi(A)$ are both sub-, img-, and rfl-minimal, in the sense of Definition 13. Furthermore, if $\varpi(A) = \{E_1, \ldots, E_n\}$ then

$$[\![A]\!]_{\mathsf{PCWA}} = \overline{[\![E_1]\!]_{\mathsf{CWA}} \cup \ldots \cup [\![E_n]\!]_{\mathsf{CWA}}}^{\cup}. \tag{1}$$

We propose $\varpi(-)$ as a form of "power core" or "multi-core" function for PCWA; giving for each A a finite set of PCWA-minimal instances which jointly embody the PCWA-relevant structure of A, analogously to the role that the single instance $\mathsf{Core}(A)$ plays in OWA. In addition to the properties just listed, we show the following as an instance of the usefulness of $\varpi(-)$. For any given instance A, the set of sub-minimal subinstances of A is of course finite. But this set may have no overlap with the set of sub-minimal subinstances of B, even if A and B are semantically equivalent. Thus it is, on the face of it, not obvious that the set $\text{Min}([A]_{\mathsf{PCWA}})$ of sub-minimal members of the whole equivalence class $[A]_{\mathsf{PCWA}}$ must be finite (up to renaming of nulls). However, we show that any sub-minimal member of $[A]_{\mathsf{PCWA}}$ must be an image of $\bigcup_{E \in \varpi(A)} E$, establishing thereby that $\varpi(A)$ both yields a finite bound on the size of $\text{Min}([A]_{\mathsf{PCWA}})$, and a way to compute it.

Moreover, we show in Section 6.3 that for the class of queries known as *existential positive with Boolean universal guards*, the so-called *certain answers* can in fact be computed directly from the elements in $\varpi(A)$, rather than from the larger A.

In the rest of Section 6 we fix the semantics to be PCWA, and thus leave the subscripts implicit.

6.1 PCWA-cores

Recall that A is a core if and only if every homomorphism $A \to A$ is an isomorphism. In analogy, we introduce the notion of PCWA-core as follows.

▶ **Definition 18.** *We say that an instance A is a* PCWA-core *if every self-cover $H \subseteq$ $\mathsf{Hom}(A, A)$ contains an isomorphism.*

▶ **Example 19.** $D = \{R(z, z, r), R(z, z, z)\}$ is a PCWA-core, as the only endomorphism hitting $R(z, z, r)$ is the identity. The core of D is $\{R(z, z, z)\}$.

Accordingly, every core is a PCWA-core. It is also evident that cores have the property that if C is a core and A is any instance, then A and C are OWA semantically equivalent if and only if C is a reflective subinstance of A. For PCWA-cores we have the following:

▶ **Proposition 20.** *Let $A \equiv B$ and assume that A is a PCWA-core. Then A is a reflective subinstance of B.*

Proof. $\mathsf{Hom}(B, A) \circ \mathsf{Hom}(A, B)$ is a cover so it contains an isomorphism. ◀

Consequently, if two PCWA-cores are semantically equivalent, they are isomorphic.

Section 5 introduced three different notions of minimality with respect to semantic equivalence. We relate these to each other and to the property of being a PCWA-core.

▶ **Proposition 21.** *Let A be an instance. The following implications hold and are strict.*

1. *If A is a PCWA-core then A is sub-minimal and img-minimal.*
2. *If A is sub-minimal or img-minimal then it is rfl-minimal.*

Proof. 1) follows from Proposition 20 and 2) is immediate.That the implications are strict is shown in Examples 14 and 16. Specifically, C_1 of Example 14 is both sub-minimal and img-minimal, but it is not a PCWA-core. And A_3 of Example 16 is rfl-minimal, but neither sub- nor img-minimal. ◀

In what follows it is convenient to fix a more compact notation for atoms $R(\mathbf{t})$ that occur in an instance A. We primarily use the variable k for atoms, and write $k : A$ for "k is an atom of A". If $f : A \to B$ is a homomorphism and $k = R(\mathbf{t}) : A$ then $f(k) = R(f(\mathbf{t}))$.

We recall the notion of "core with respect to a tuple":

▶ **Definition 22.** *Let $k : A$. The* core of A with respect to k*, denoted C_k^A, is the least strict reflective subinstance of A containing k.*

The instance C_k^A can be regarded as the "core of A with k frozen", and thus is unique, up to isomorphism. As a reflective subinstance, it comes with an injective homomorphism to A and a strong surjection from A, which we write $m_k : C_k^A \to A$ and $q_k : A \to C_k^A$, respectively. When the instance A is clear from context, we leave the superscript implicit and just write C_k. We display the following for emphasis.

▶ **Lemma 23.** *Any homomorphism $h : C_k^A \to C_k^A$ that fixes k must be an isomorphism.*

▶ **Definition 24.** *We say that two atoms $k, k' : A$ are* endomorphism-equivalent*, and write $k \sim_A k'$, if there exist $f, g \in \mathsf{Hom}(A, A)$ such that $f(k) = k'$ and $g(k') = k$. We say that $k : A$ is* (endomorphism-)maximal *if "only equivalent atoms map to it". That is, for all $k' : A$ and $f \in \mathsf{Hom}(A, A)$, $f(k') = k$ implies that $k \sim_A k'$. If k is maximal we write $\mathrm{Max}_A(k)$.*

▶ **Lemma 25.** *Let A be an instance, and $k, k' : A$. If $k \sim_A k'$ then $C_k \cong C_{k'}$.*

Proof. Suppose $f, g \in \mathsf{Hom}(A, A)$ such that $f(k) = k'$ and $g(k') = k$ and consider the diagram

$$\begin{tikzcd}
& A \arrow[loop above, "{f,g}"] \arrow[dl, bend right, "q_{k'}"'] \arrow[dr, bend left, "q_k"] & \\
C_{k'} \arrow[ur, bend left, "m_{k'}"] \arrow[rr, bend left, "q_k \circ f \circ m_{k'}"] & & C_k \arrow[ul, bend right, "m_k"'] \arrow[ll, bend left, "q_{k'} \circ g \circ m_k"]
\end{tikzcd}$$

The homomorphism $h := (q \circ f \circ m') \circ (q' \circ g \circ m : C_k \to C_k)$ fixes k. So h must be an isomorphism. By symmetry, we obtain that $C_k \cong C_{k'}$. ◀

▶ **Lemma 26.** *If $k : A$ is maximal, then C_k is a PCWA-core.*

Proof. First note that for any instance B and any set of homomorphisms $H \subseteq \mathsf{Hom}(B, B)$, if $\overline{H}$ is the closure of H under composition, then: 1) H is a cover if and only if $\overline{H}$ is a cover; and 2) H contains an isomorphism if and only if $\overline{H}$ contains an isomorphism. Let $H \subseteq \mathsf{Hom}(C_k, C_k)$ be a cover, and assume without loss of generality that it is closed under composition. Then we can find $k' : C_k$ and $f \in H$ such that $f(k') = k$. Since k is maximal in A it is maximal in C_k, so there is a homomorphism $h : C_k \to C_k$ such that $h(k) = k'$. But then $f \circ h$ is an isomorphism, so f must be an isomorphism as well. ◀

Thus the maximal atoms of an instance determine a set of reflective subinstances which are PCWA-cores. We show that these are invariant under semantic equivalence.

▶ **Theorem 27.** *Let $H \subseteq \mathsf{Hom}(A, B)$ and $G \subseteq \mathsf{Hom}(B, A)$ be covers. Let $k_B : B$ be maximal. Then there exist $h \in H$ and $k_A : A$ such that k_A is maximal and $h(k_A) = k_B$. Moreover, the homomorphism $q_{k_B} \circ h \circ m_{k_A} : C^A_{k_A} \to C^B_{k_B}$ is an isomorphism.*

Proof. First, we show that, more generally, whenever $A \equiv B$, it is the case that for all $f : A \to B$ and all $k : A$, if $\mathrm{Max}_B(f(k))$ then $\mathrm{Max}_A(k)$.

For suppose $g : A \to A$ and $k' : A$ is such that $g(k') = k$. Choose $f' : B \to A$ and $k'' : B$ such that $f'(k'') = k'$. Then $f \circ g \circ f'(k'') = f(k)$ so there is $f'' : B \to B$ such that $f''(f(k)) = k''$, whence $g \circ f' \circ f'' \circ f(k) = k'$. This establishes the first claim of the theorem.

Next, let $\mathrm{Max}_A(k_A)$, $\mathrm{Max}_B(k_B)$, and $h \in H$ such that $h(k_A) = k_B$. Chose $k' : B$ and $g \in G$ such that $g(k') = k_A$. Then $f \circ g(k') = k_B$, so there exists $f : B \to B$ such that $f(k_B) = k'$.

$$\begin{tikzcd}
A \arrow[rr, bend left, "h"] \arrow[d, bend left, "q_{k_A}"] & & B \arrow[ll, bend left, "g"] \arrow[loop above, "f"] \arrow[d, bend left, "q_{k_B}"] \\
C^A_{k_A} \arrow[u, hook, bend left, "m_{k_A}"] \arrow[rr, bend left, "q_{k_B} \circ h \circ m_{k_A}"] & & C^B_{k_B} \arrow[u, hook, bend left, "m_{k_B}"] \arrow[ll, bend left, "q_{k_A} \circ g \circ f \circ m_{k_B}"]
\end{tikzcd}$$

Then $q_{k_B} \circ h \circ m_{k_A}(k_A) = k_B$ and $q_{k_A} \circ g \circ f \circ m_{k_B}(k_B) = k_A$, whence their composites are isomorphisms. So they must themselves be isomorphisms. ◀

Finally, we note the following property of PCWA-cores which will be used in the next section.

▶ **Lemma 28.** *An instance A is a PCWA-core if and only if there exists $k : A$ with the property that for all $f \in \mathsf{Hom}(A, A)$, if k is in the image of f then f is an isomorphism.*

Proof. Suppose A is a PCWA core. For each maximal k, let $f_k : A \to A$ be the composition of $q_k : A \to C_k$ and $m_k : C_k \to A$. Then $\mathsf{Hom}(A, A) \circ \{f_k \mid \mathrm{Max}_A(k)\}$ is covering, so one of its homomorphisms, and hence one of the f_ks, must be an isomorphism. The converse is immediate. ◀

6.2 PCWA Multicores

Consider the family $\{C_k|\mathrm{Max}_A(k)\}$ of (strict) reflective subinstances of A. From the definition of maximality we have that for any atom $t : A$ there exists a maximal atom $k : A$ and an endomorphism $h : A \to A$ such that $f(k) = t$. Thus the family $\{C_k \mid \mathrm{Max}_A(k)\}$ jointly covers A. Clearly, if we successively remove any member of $\{C_k \mid \mathrm{Max}_A(k)\}$ that is a reflective subinstance of another member, we will retain a subset that still jointly covers A. Thus we can summarize what we have so far with the following.

▶ **Theorem 29.**
1. *For each $A \in \mathcal{D}$ there exists a finite set $\varpi(A) \subseteq \mathcal{D}$ such that:*
   a. *for all $E \in \varpi(A)$, $E \cong C_k^A$ for some maximal $k : A$;*
   b. *for all maximal $k : A$, there exists $E \in \varpi(A)$ such that C_k^A is a reflective subinstance of E; and*
   c. *for all $E, E' \in \varpi(A)$ if E is a reflective subinstance of E' then $E = E'$.*
2. *for given $A \in \mathcal{D}$ the set $\varpi(A)$ is unique with properties 1.(a)–1.(c), up to isomorphisms of its members. That is to say, if X is another set satisfying properties 1.(a)–1.(c), then there exists a bijection $f : \varpi(A) \to X$ such that $f(E) \cong E$.*
3. *$A \equiv B$ if and only if $\varpi(A) = \varpi(B)$, up to isomorphism of the members.*

We refer to $\varpi(A)$ as the *multicore* of A. The multicore of an instance A is only defined up to isomorphisms of its members, so we can assume without loss whenever it is convenient that no nulls are shared between those members; i.e. that for all $E, E' \in \varpi(A)$ we have $\mathrm{dom}(E) \cap \mathrm{dom}(E') \subseteq \mathsf{Cons}$. We also regard multicores as equal when their members are isomorphic.

Before proceeding, we characterize when a set of instances is (up to isomorphism) $\varpi(A)$ for some A. We need the following lemma.

▶ **Lemma 30.** *Let A be an instance and B a reflective subinstance of A. Let $k : A$ and suppose that k is maximal. Then C_k is a reflective subinstance of B if and only if there exists a homomorphism $f : B \to C_k$ and $k' : B$ such that $f(k') = k$.*

Proof. The left-to-right is immediate. Assume that there exists a homomorphism $f : B \to C_k$ and $k' : B$ such that $f(k') = k$. Since k is maximal in A there exists a homomorphism $g : C_k \to B$ such that $g(k) = k'$. But then $f \circ g$ fixes k, so it is an isomorphism. ◀

▶ **Theorem 31.** *Let $\mathfrak{F} = \{C_1, \cdots, C_n\}$ be a family of instances (with no nulls in common). The following are equivalent:*
1. *There exists an instance A such that $\mathfrak{F} = \varpi(A)$ (up to isomorphism of the members).*
2. a. *C_i and C_j have the same core (up to isomorphism) for all $i, j \leq n$, and*
   b. *there exists a selection of atoms $k_i : C_i$, $1 \leq i \leq n$, satisfying the condition that if there exists $h : C_j \to C_i$ such that k_i is in the image of h, then $i = j$ and h is an isomorphism.*

Proof. Assume $\mathfrak{F} = \varpi(A)$. Then we can regard $\varpi(A)$ as $\{C_k \mid k \in I\}$ for a set I of maximal $k : A$. Firstly, the core of A is the core of C_k for all $k \in I$. Secondly, by Lemma 30, if $h : C_j \to C_k$ such that k is in the image of h then C_k is a reflective subinstance of C_j, whence by the definition of $\varpi(A)$ we have that $j = k$ and h is an isomorphism.

Assume conditions in a) and b) are satisfied. b) ensures, together with Lemma 28, that C_i is a PCWA core for all $i \in I$. Let $A := \bigcup_{k \in I} C_k$ (relying on the assumption that the members of $\mathfrak{F}$ have no nulls in common). Since the C_is share the same core, C_i is a reflective subinstance of A for all $i \in I$. Specifically, $m_i : C_i \to A$ is the inclusion and $q_i : A \to C_i$ is the homomorphism induced by $\{f_{j,i} : C_j \to C_i \mid j \in I\}$ where $f_{j,i}$ sends C_j to the core if $j \neq i$, and $f_{i,i}$ is the identity. Next, to show that c_i is maximal in A for all $i \in I$: suppose there exists a homomorphism $h : A \to A$ and a $t : A$ such that $h(t) = c_i$. Then t is contained in some C_j. By composing

$$C_j \xrightarrow{m_j} A \xrightarrow{h} A \xrightarrow{q_i} C_i$$

and by Lemma 30, we see that $i = j$ and $(q_i \circ h \circ m_i)$ is an isomorphism on C_i. Hence $m_i \circ (q_i \circ h \circ m_i)^{-1} \circ q_i(c_i) = t$. Finally, if $k : A$ is maximal, then $k : C_i$ for some $i \in I$, whence C_k is a reflective subinstance of C_i. ◀

Let $\chi_p(A)$ be the union of all members of the multicore, where these are chosen so as to have no nulls in common, $\chi_p(A) := \bigcup_{E \in \varpi(A)} E$. It is now easy to see that $\chi_p(A) \equiv A$, so that $\chi_p(-)$ is a representative function, in the sense of Section 2.3. $\chi_p(A)$ need not be minimal either in terms of subinstances or images. However, an instance is sub-minimal only if it is an image of $\chi_p(A)$, as we show by way of the following lemma.

▶ **Lemma 32.** *Let $A \in \mathcal{D}$. There exists a homomorphism $m : \chi_p(A) \to A$ such that $\chi_p(A) \equiv m(\chi_p(A)) \equiv A$.*

Proof. We can regard $\varpi(A)$ as a set $\{C_k^A \mid k \in S\}$ for a suitable set S. For each $k \in S$ we have an inclusion $i_k : C_k \to A$, and a strong surjection $s_k : A \to C_k$. The family $\{i_k : C_k \to A \mid k \in S\}$ determines a homomorphism $m : \chi_p(A) \to A$. For each k the inclusion $i_k : C_k \to A$ factors through $m(\chi_p(A))$ and the composite $C_k \xrightarrow{i_k} m(\chi_p(A)) \subseteq A \xrightarrow{s_k} C_k$ is the identity. Thus $\chi_p(A) \equiv m(\chi_p(A)) \equiv A$. ◀

▶ **Theorem 33.** *An instance A is sub-minimal only if there exists a strongly surjective homomorphism $m : \chi_p(A) \to A$.*

▶ **Corollary 34.** *Identifying isomorphic instances, the number of sub-minimal instances that are semantically equivalent to A is bounded by the number of (semantically equivalent) images of $\chi_p(A)$.*

▶ Remark 35. We note that there will usually be proper semantically equivalent images of $\chi_p(A)$. In particular, this always exists if the core of A has a null in it and $\varpi(A)$ has more than one member. The reason is that members of $\varpi(A)$ can be "glued" along common reflective subinstances; such subinstances induce a filter which yields a semantically equivalent image of $\chi_p(A)$. Observe that if $\varpi(A)$ has a *single* member, then that member is equivalent to A, and thus $[A]$ has a least element both in terms of subinstances, reflective subinstances, and images.

▶ **Example 36.** Consider Example 15. $\varpi(A_2)$ consists of the two PCWA-cores C_{k_1} and C_{k_5}. In addition to the core, C_{k_1} and C_{k_5} have the reflective subinstances V and W in common.

C_{k_1}	R				
k_1	x	x	u	y	z
k_2	x	x	x	x	z
k_3	x	x	x	y	x
k_4	x	x	x	x	x

C_{k_5}	R				
k_5	v	p	p	r	s
k_6	p	p	p	p	s
k_7	p	p	p	r	p
k_8	p	p	p	p	p

V	R				
	x	x	x	y	x
	x	x	x	x	x

W	R				
	p	p	p	p	s
	p	p	p	p	p

The filter induced on $C_{k_1} \cup C_{k_5} = A_2$ by V identifies x with p and y with r. If we write out the resulting image by overwriting x with p and y with r, we obtain B_2 of Example 15. It follows that B_2 is a semantically equivalent image of A_2. Similarly, from W we see that we can produce a semantically equivalent image of A_2 by overwriting x with p and z with s. This results in C_2.

6.3 Naïve Evaluation of Queries

Before proceeding to the study of minimality for OCWA* in general, we make an example remark on the use of $\varpi(-)$ in the evaluation of queries. The motivation is, briefly, that it may be significantly cheaper to evaluate a query separately on the smaller instances in $\varpi(A)$ than on all of A.

Recall from e.g. [12] that that the *certain answers* of a query Q on an instance A under a semantics $[\![-]\!]$ is the intersection of the answers obtained on $[\![A]\!]$: $\mathsf{certain}(Q, A) := \bigcap\{Q(I) \mid I \in [\![A]\!]\}$. The *naïve evaluation* of Q on A is the result of removing all tuples with nulls from $Q(A)$. Naïve evaluation is said to *work* for Q if it produces precisely the certain answers.

It is shown in [12] that naïve evaluation works for the class $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$ of existential positive queries with Boolean universal guards with respect to PCWA. Here $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$ is the least class of formulas containing all atomic formulas, including equality statements, and closed under conjunction; disjunction; existential quantification; and the following rule: if α is an atomic formula, ϕ a formula in $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$, and $\mathbf{x}$ a list of distinct variables containing all free variables in both α and ϕ, then $\forall\mathbf{x}(\alpha \to \phi)$ is a formula in $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$.

▶ **Theorem 37.** *Let Q be a query of arity n in $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$. Then* $\mathsf{certain}(Q, A) = \bigcap\{Q(E) \mid E \in \varpi(A)\} \cap \mathsf{Const}^n$.

Proof. The inclusion from left to right follows from the fact that naïve evaluation works for Q, that Q is preserved under strong surjections, and that each $E \in \varpi(A)$ is an image of A. For the inclusion from right to left, it is sufficient to show that if $\mathbf{a}$ is a tuple of constants in $\bigcap\{Q(E) \mid E \in \varpi(A)\}$ then $\mathbf{a} \in Q(I)$ for all $I \in \overline{\cup_{E\in\varpi(A)}[\![E]\!]_{\mathsf{CWA}}}^{\cup}$. But this is a straightforward modification of the proof that formulas in $\exists\mathbf{Pos} + \forall\mathbf{G}^{\text{bool}}$ are preserved under unions of strong surjections (Lemma 10.12) in [12]. ◀

7 Minimality in OCWA*

We return now to OCWA* in general and apply our results from the special case of the previous section. Recall from Section 4 that in order to determine whether $A[\mathbf{X}] \leq_{\mathsf{OCWA*}} B[\mathbf{Y}]$ we have to look for an RCN-cover from $B[\mathbf{Y}]$ to $A^{\text{length}(\mathbf{Y})+1}[\mathbf{X}]$. The reason is that RCN-covers do not compose; it is insufficient just to know that we have an RCN-cover from $B[\mathbf{Y}]$ to $A[\mathbf{X}]$. This fact complicates the study of minimality for OCWA*. However, note that if there exists an RCN-cover from $B[\mathbf{Y}]$ to $A[\mathbf{X}]$ which sends the closed nulls $\mathbf{Y}$ to closed terms in $A[\mathbf{X}]$ – i.e. either to $\mathbf{X}$ or to constants – then, because such covers do compose, we have $A[\mathbf{X}] \leq_{\mathsf{OCWA*}} B[\mathbf{Y}]$. But, on the face of it, we cannot restrict to such "closedness-preserving" covers. Consider the following example.

▶ **Example 38.** The following annotated instances are all semantically equivalent.

$$A[V,W] = \{R(x,y), R(V,W)\} \quad A[V,w] = \{R(x,y), R(V,w)\} \quad A[v,w] = \{R(x,y), R(v,w)\}$$
$$A[v,W] = \{R(x,y), R(v,W)\} \quad B[] = \{R(x,y)\}$$

Although $A[V, W]$ and $B[]$ are equivalent, there is no RCN-cover from $A[V, W]$ to $B[]$ that satisfies the restriction that closed nulls should be sent to closed terms, since everything in $B[]$ is open. It is the (ordinary) RCN-cover to $B^3[]$ that witnesses that $B[] \leq_{\mathsf{OCWA*}} A[V, W]$. Nevertheless, Example 38 hints at a solution to this; the problem with $A[V, W]$, it can be said, is that it has closed nulls that could equivalently have been annotated as open. Once we re-annotate to $A[v, w]$, the equivalence with $B[]$ *is* witnessed by a cover to $B[]$. We show in this section that if we restrict to annotated instances where no closed null can be equivalently replaced by an open null, then any semantic equivalence is witnessed by RCN-covers which preserve closed nulls. These closed nulls can then, essentially, be treated as constants, so that the results of Section 6 can be applied.

We note, first, when an annotated instance is semantically equivalent to an instance without closed nulls. Our fixed semantics in this section is OCWA*.

▶ **Proposition 39.** *Let $A[\mathbf{X}]$ be an annotated instance. The following are equivalent:*
1. *$A[\mathbf{X}]$ is semantically equivalent to an instance $B[]$ in which all nulls are open.*
2. *$[\![A[\mathbf{X}]]\!]$ is closed under unions.*
3. *$A[\mathbf{X}]$ is semantically equivalent to the instance $A[\mathbf{x}]$ obtained by changing the annotation of $A[\mathbf{X}]$ so that all nulls are open.*

Proof. For 2. ⇒ 3., note that $[\![A[\mathbf{X}]]\!] \subseteq [\![A[\mathbf{x}]]\!]$ is clear; and it is also clear that $[\![A[\mathbf{x}]]\!]_{\mathsf{CWA}} \subseteq [\![A[\mathbf{X}]]\!]$. But then, since $[\![A[\mathbf{X}]]\!]$ is closed under unions, $[\![A[\mathbf{x}]]\!] \subseteq [\![A[\mathbf{X}]]\!]$. ◀

▶ **Definition 40.** *Let $A[\mathbf{X}, \mathbf{Y}]$ be an annotated instance. We say that $\mathbf{X}$ is* annotation redundant *(relative to $\mathbf{Y}$) if $A[\mathbf{X}, \mathbf{Y}] \equiv A[\mathbf{x}, \mathbf{Y}]$ i.e. if changing the annotation of $\mathbf{X}$ to "open" yields an equivalent instance. We say that an annotated incomplete instance is* annotation minimal *is no subset of its closed nulls are annotation redundant (with respect to the rest).*

▶ **Lemma 41.** *Let $A[\mathbf{X}, \mathbf{Y}]$ be an annotated incomplete instance and $\mathbf{c}$ a list of the same length as $\mathbf{Y}$ of distinct constants not occurring in A . Then $\mathbf{X}$ is redundant with respect to $\mathbf{Y}$if and only if for all finite lists of instances $I_1, \ldots, I_k \in [\![A[\mathbf{X}, \mathbf{c}]]\!]$ it is the case that $I_1 \cup \ldots \cup I_k \in [\![A[\mathbf{X}, \mathbf{Y}]]\!]$.*

Proof. *If*: We must show that $A[\mathbf{X}, \mathbf{Y}] \equiv A[\mathbf{x}, \mathbf{Y}]$. Let $m \geq 1$ be given, and consider $A^m[\mathbf{x}, \mathbf{Y}]$. Let J be a freeze of $A^m[\mathbf{x}, \mathbf{Y}]$ where $\mathbf{Y}$ is replaced by $\mathbf{c}$ and the other nulls by fresh constants. Then $J = I_1 \cup \ldots \cup I_m$ where $I_1, \ldots, I_m \in [\![A[\mathbf{X}, \mathbf{c}]]\!]$. So $J \in A[\mathbf{X}, \mathbf{Y}]$, by assumption, and then $A[\mathbf{X}, \mathbf{Y}] \equiv A[\mathbf{x}, \mathbf{Y}]$ by Theorem 12.

Only if: If $I_1, \ldots, I_k \in [\![A[\mathbf{X}, \mathbf{c}]]\!]$ then $I_1, \ldots, I_k \in [\![A[\mathbf{x}, \mathbf{c}]]\!]$, and then, since the latter is closed under unions, $I_1 \cup \ldots \cup I_k \in [\![A[\mathbf{x}, \mathbf{c}]\!] \subseteq [\![A[\mathbf{x}, \mathbf{Y}]]\!] = [\![A[\mathbf{X}, \mathbf{Y}]]\!]$. ◀

The following theorem displays the main property of annotation-minimal instances. The proof is rather long and is omitted for reasons of space.

▶ **Theorem 42.** *Let $A[\mathbf{X}]$ and $B[\mathbf{Y}]$ be two annotation-minimal instances such that $A[\mathbf{X}] \equiv B[\mathbf{Y}]$. Then for all strong surjections $f : A^\infty[\mathbf{X}] \twoheadrightarrow B^\infty[\mathbf{Y}]$ it is the case that f restricts to a bijection $f\restriction_{\mathbf{X}}: \mathbf{X} \to \mathbf{Y}$ on the sets of closed nulls.*

▶ **Corollary 43.** *Let $A[\mathbf{X}]$ and $B[\mathbf{Y}]$ be two annotation-minimal instances. Then $A[\mathbf{X}] \equiv B[\mathbf{Y}]$ if and only if there exists RCN-covers $\{f_i : A[\mathbf{X}] \to B[\mathbf{Y}] | 1 \leq i \leq n\}$ and $\{g_j : B[\mathbf{Y}] \to A[\mathbf{X}] | 1 \leq j \leq m\}$ such that f_i restricts to a bijection $f_i\restriction_{\mathbf{X}}: \mathbf{X} \to \mathbf{Y}$ and g_j to a bijection $g_j\restriction_{\mathbf{Y}}: \mathbf{Y} \to \mathbf{X}$.*

▶ **Corollary 44.** *Let $A[\mathbf{X}]$ and $B[\mathbf{Y}]$ be two annotation-minimal instances. Then $A[\mathbf{X}] \equiv B[\mathbf{Y}]$ if and only if $\mathbf{X}$ is of the same length as $\mathbf{Y}$ and there exists injective functions $f : \mathbf{X} \to \mathsf{Const}$ and $g : \mathbf{Y} \to \mathsf{Const}$ such that $A[f(\mathbf{X})/\mathbf{X}] \equiv_{\mathsf{PCWA}} B[g(\mathbf{Y})/\mathbf{Y}]$, where $f(\mathbf{X})$ and $g(\mathbf{Y})$ are disjoint from the constants in $A[\mathbf{X}]$ and $B[\mathbf{Y}]$.*

That is to say, $A[\mathbf{X}]$ and $B[\mathbf{Y}]$ are equivalent if there is a way to "freeze" the closed nulls so that they become PCWA-equivalent.

Now, let $A[\mathbf{X}]$ be an annotation-minimal instance. Let $\mathbf{c}$ be a list of fresh constants, of the same length as $\mathbf{X}$. Then we can compute $\varpi(A[\mathbf{c}]) = \{E_1, \ldots, E_n\}$ and $\chi_p(A[\mathbf{c}]) = E_1 \cup \ldots \cup E_n$, as in Section 6, and then substitute $\mathbf{X}$ back in for $\mathbf{c}$. This yields a set $\varpi(A[\mathbf{X}]) := \{E_1[\mathbf{X}], \ldots, E_n[\mathbf{X}]\}$ of annotated instances and an annotated instance $\chi_p(A)[\mathbf{X}] := E_1[\mathbf{X}] \cup \ldots \cup E_n[\mathbf{X}]$. Since $\chi_p(A)[\mathbf{X}]$ is semantically equivalent to $A[\mathbf{X}]$ and has the same (number of) closed nulls, $\chi_p(A)[\mathbf{X}]$ is annotation-minimal. Thus we have a function $\varpi(-)$ from annotation minimal annotated instances to finite sets of annotated instances, and $\chi_p(-)$ from annotation minimal instances to annotation minimal instances. As in Section 6, we identify $\varpi(A[\mathbf{X}])$ and $\varpi(B[\mathbf{Y}])$ if they "are the same up to renaming of nulls", but in the presence of closed nulls we have to add a condition to what this means: we say that $\varpi(A[\mathbf{X}]) = \varpi(B[\mathbf{Y}])$ if there is a bijection of sets F between them; an isomorphism $f_E : E \to F(E)$ for each $E \in \varpi(A[\mathbf{X}])$; and for all $E, E' \in \varpi(A[\mathbf{X}])$, the homomorphisms f_E and $f_{E'}$ restrict to one and the same bijection of sets $\mathbf{X} \to \mathbf{Y}$. We now have:

▶ **Theorem 45.** *Let $A[\mathbf{X}]$, $B[\mathbf{Y}]$ be an annotation-minimal instances. Then:*

1. *$A[\mathbf{X}] \equiv B[\mathbf{Y}]$ if and only if $\varpi(A[\mathbf{X}]) = \varpi(B[\mathbf{Y}])$;*
2. *$A[\mathbf{X}] \equiv \chi_p(A[\mathbf{X}])$, and $\chi_p(A[\mathbf{X}])$ is annotation-minimal;*
3. *if $E[\mathbf{X}] \in \varpi(A[\mathbf{X}])$ and $A[\mathbf{X}] \equiv B[\mathbf{Y}]$ then $E[\mathbf{X}]$ is a reflective subinstance of $B[\mathbf{Y}]$ (up to annotation-preserving isomorphism); and*
4. *if $A[\mathbf{X}] \equiv B[\mathbf{Y}]$ then $B[\mathbf{Y}]$ is sub-minimal only if there is a strongly surjective homomorphism*
   $\chi_p(A)[\mathbf{X}] \twoheadrightarrow B[\mathbf{Y}]$ restricting to a bijection $\mathbf{X} \to \mathbf{Y}$.

Accordingly, $\varpi(-)$ is representative in the sense of (1) and (2) and minimal in the sense of (3). $\chi_p(-)$ bounds the number of sub-minimal equivalent instances by (4). $\varpi(-)$ (and $\chi_p(-)$) can be extended to all annotated instances by first choosing an equivalent annotation minimal instance and then applying $\varpi(-)$, and (1) ensures that the result does not depend on the choice.

8 Discussion and Conclusion

In this work we study the problems of implication, equivalence, and minimality (and consequently cores) in mixed open and closed worlds. These problems have particular importance in the context of date exchange and remain open for several variants of mixed worlds. In particular, we adress these problems for the Closed Powerset semantics and the OCWA semantics. To this end, we define a novel semantics for mixed worlds that we called OCWA* and subsumes both Closed Powerset and OCWA. Our semantics is introduced with the help of homomorphic covers and it is characterised in terms of such covers. For the minimization problem we presented negative results for several common notions of minimality. Then, we showed that one can find cores using a different notion of minimality.

Observe that homomorphic covers have been already used in several related contexts. In [15], Grahne et al. uses homomorphic covers in the context of source instance recovery in data exchange. In [6], Chaudhuri and Vardi give the existence of a cover as a sufficient condition for conjunctive query containment under bag semantics. In [22], Kostylev et al. use various notions of cover to study annotated query containment. On the other hand, Knauer and Ueckerdt [20] apply this notion to coverage relations between graphs.

In our opinion several more data management scenarios can benefit from the concept of homomorphic cover and the machinery that we have developed for it. For instance, two conjunctive queries whose relational structures cover each other retrieve the same tuples from every relation of any database instance, a fact of potential relevance in e.g. data privacy settings. In the field of constraint programming, this property is closely connected to the notion of a minimal constraint network [14], and may have applications there. For another example, treating one conjunctive query as a view, it can be used to completely rewrite another if there exists a cover from the view (cf. [23]). Thus in this setting, cover-equivalence corresponds to mutual complete rewritability.

References

1. Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
2. Marcelo Arenas, Pablo Barceló, Leonid Libkin, and Filip Murlak. *Foundations of Data Exchange*. Cambridge University Press, 2014.
3. Marcelo Arenas, Elena Botoeva, Diego Calvanese, and Vladislav Ryzhikov. Knowledge base exchange: The case of OWL 2 QL. *Artif. Intell.*, 238:11–62, 2016.
4. Marcelo Arenas and Leonid Libkin. XML data exchange: Consistency and query answering. *J. ACM*, 55(2):7:1–7:72, 2008.
5. Philip A. Bernstein and Sergey Melnik. Model management 2.0: manipulating richer mappings. In *SIGMOD*, pages 1–12, 2007.
6. Surajit Chaudhuri and Moshe Y. Vardi. Optimization of Real Conjunctive Queries. In *PODS*, 1993.
7. Ronald Fagin, Laura M. Haas, Mauricio A. Hernández, Renée J. Miller, Lucian Popa, and Yannis Velegrakis. Clio: Schema Mapping Creation and Data Exchange. In *Conceptual Modeling: Foundations and Applications*, pages 198–236, 2009.
8. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.
9. Ronald Fagin, Phokion G. Kolaitis, and Lucian Popa. Data exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.
10. Jan Foniok. *Homomorphisms and Structural Properties of Relational Systems*. PhD thesis, Charles University in Prague, 2007.
11. Henrik Forssell, Evgeny Kharlamov, and Evgenij Thorstensen. Towards Characterising Data Exchange Solutions in Open and Closed Words (Extended Abstract). In *AMW*, 2017.
12. Amélie Gheerbrant, Leonid Libkin, and Cristina Sirangelo. NaïVe Evaluation of Queries over Incomplete Databases. *ACM Trans. Database Syst.*, 39(4):31:1–31:42, December 2014.
13. Ladan Golshanara and Jan Chomicki. Temporal data exchange. *Inf. Syst.*, 87, 2020.
14. Georg Gottlob. On minimal constraint networks. *Artif. Intell.*, 191-192:42–60, 2012.
15. Gösta Grahne, Ali Moallemi, and Adrian Onet. Recovering Exchanged Data. In *PODS*, pages 105–116, 2015.
16. Pavol Hell and Jaroslav Nešetřil. The core of a graph. *Discrete Mathematics*, 109(1):117–126, 1992.
17. André Hernich. Answering Non-monotonic Queries in Relational Data Exchange. In *ICDT*, pages 143–154, 2010.
18. André Hernich, Leonid Libkin, and Nicole Schweikardt. Closed world data exchange. *ACM Trans. Database Syst.*, 36(2):14:1–14:40, 2011.

19 Angelika Kimmig, Alex Memory, Renée J. Miller, and Lise Getoor. A Collective, Probabilistic Approach to Schema Mapping Using Diverse Noisy Evidence. *IEEE Trans. Knowl. Data Eng.*, 31(8):1426–1439, 2019.

20 Kolja Knauer and Torsten Ueckerdt. Three ways to cover a graph. *Discrete Mathematics*, 339(2):745–758, 2016.

21 Phokion G. Kolaitis. Schema mappings, data exchange, and metadata management. In *PODS*, 2005.

22 Egor V. Kostylev, Juan L. Reutter, and András Z. Salamon. Classification of annotation semirings over containment of conjunctive queries. *ACM Trans. Database Syst.*, 39(1):1, 2014.

23 Alon Y. Levy, Alberto O. Mendelzon, Yehoshua Sagiv, and Divesh Srivastava. Answering Queries Using Views. In *PODS*, pages 95–104, 1995.

24 Leonid Libkin and Cristina Sirangelo. Data exchange and schema mappings in open and closed worlds. *J. Comput. Syst. Sci.*, 77(3):542–571, 2011.

25 Thomas Lukasiewicz, Maria Vanina Martinez, Livia Predoiu, and Gerardo I. Simari. Basic Probabilistic Ontological Data Exchange with Existential Rules. In *AAAI*, pages 1023–1029, 2016.

26 Lucian Popa, Yannis Velegrakis, Renée J. Miller, Mauricio A. Hernández, and Ronald Fagin. Translating Web Data. In *VLDB*, 2002.

27 Balder ten Cate, Phokion G. Kolaitis, Kun Qian, and Wang-Chiew Tan. Approximation Algorithms for Schema-Mapping Discovery from Data Examples. *ACM Trans. Database Syst.*, 42(2):12:1–12:41, 2017.

28 Balder ten Cate, Phokion G. Kolaitis, Kun Qian, and Wang-Chiew Tan. Active Learning of GAV Schema Mappings. In *PODS*, pages 355–368, 2018.