Semantic Bijectivity and the Uniqueness of Constant-Complement Updates in the Relational Context

Stephen J. Hegner
Umeå University
Department of Computing Science
SE-901 87 Umeå, Sweden
`hegner@cs.umu.se`
`http://www.cs.umu.se/~hegner`

Minor corrections: 08 November 2008

Abstract

Within the context of the relational model, a general technique for establishing that the translation of a view update defined by constant complement is independent of the choice of complement is presented. In contrast to previous results, the uniqueness is not limited to order-based updates (those constructed from insertions and deletions), nor is it limited to those well-behaved complements which define closed update strategies. Rather, the approach is based upon optimizing the change of information in the main schema which the view update entails. The only requirement is that the view and its complement together possess a property called *semantic bijectivity* relative to the information measure. It is furthermore established that a very wide range of views have this property. This results formalizes the intuition, long observed in examples, that it is difficult to find different complements which define distinct but reasonable update strategies.

1. Introduction

It has long been recognized that there is no ideal solution to the problem of supporting updates to views of database schemata. Rather, all solutions involve compromise of some sort. At the most conservative end of the spectrum lies the constant-complement strategy, introduced by Bancilhon and Spyratos more than a quarter-century ago [BS81]. It has recently seen renewed interest, both on the theoretical front [Lec03] [Heg04] and as a framework for applications [FGM*07], to no small extent because it is precisely the strategy which avoids all so-called update anomalies [Heg04, Sec. 1].

The idea behind the constant-complement strategy is simple. Let $\mathbf{D}$ be a database schema, with $\mathsf{LDB}(\mathbf{D})$ denoting its set of legal states. An update on $\mathbf{D}$ is just a pair $(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$ in which M_1 is the current state and M_2 is the new state after the update. A view of $\mathbf{D}$ is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is the view schema and $\gamma : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V})$ is the view mapping. Since a view is window on the main schema, its state must always be determined by that of the main schema

$\mathbf{D}$; hence, γ is always taken to be surjective. Let $M_1 \in \mathsf{LDB}(\mathbf{D})$ be the current state of the main schema $\mathbf{D}$, so that $\gamma(M_1)$ is the current state of the view Γ. A translation of the update request $(\gamma(M_1), N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V})$ (relative to M_1) is an update (M_1, M_2) on $\mathbf{D}$ with the property that $\gamma(M_2) = N_2$. In general, there are many such translations. However, now let $\Gamma' = (\mathbf{V}', \gamma')$ be a second view of $\mathbf{D}$. The decomposition mapping for $\{\Gamma, \Gamma'\}$ is $\gamma \coprod \gamma' : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}) \coprod \mathsf{LDB}(\mathbf{V}')$ given on elements by $M \mapsto (\gamma(M) \uplus \gamma'(M))$. Here $\uplus$ is the disjoint union operator; thus, the decomposition mapping retains the state of each constituent view. (See 3.4 for details.) The view Γ' is called a complement of Γ, and $\{\Gamma, \Gamma'\}$ is called a complementary pair, if this decomposition mapping is injective (or lossless), so that the state of $\mathbf{D}$ is recoverable from the combined states of the two views. A translation (M_1, M_2) of the proposed view update $(\gamma(M_1), N_2)$ to Γ has constant complement Γ' if $\gamma'(M_1) = \gamma'(M_2)$. If $\{\Gamma, \Gamma'\}$ forms a complementary pair, then it is easy to see that there can be at most one translation of $(\gamma(M_1), N_2)$ which has constant complement Γ', since there can be at most one $M_2 \in \mathsf{LDB}(\mathbf{D})$ with the property that $(\gamma \coprod \gamma')(M_2) = (N_2, \gamma'(M_1))$. Thus, upon fixing a complement Γ' to Γ, translations of view updates become unique.

There is a very desirable feature which limits the choice of complement Γ'. In general, the user of the view Γ will not know the precise state M_1 of the main schema; rather, only the image $\gamma(M_1)$ will be known within the context of Γ. Ideally, whether or not the view update $(\gamma(M_1), N_2)$ is allowed should depend only upon $\gamma(M_1)$ and N_2, and not upon M_1 itself. In other words, given $M_1' \in \mathsf{LDB}(\mathbf{D})$ with the property that $\gamma(M_1) = \gamma(M_1')$, the view update $(\gamma(M_1), N_2)$ should either be allowed for both M_1 and M_1', or else for neither. In [Heg04, 2.14], it is shown that this condition recaptured by requiring that the pair $\{\Gamma, \Gamma'\}$ be meet complementary, in the sense that the congruences on $\mathsf{LDB}(\mathbf{D})$ defined by γ and γ' commute.

Even upon limiting attention to commuting congruences, there is a further complication surrounding the constant-complement approach; namely, the translation of a view update to the main schema is dependent upon the choice of complement, and except in the simplest of situations, there are many possible complements. In contrast to this theoretical result, in practice it is often clear that there is a natural translation of view updates to the main schema which is supported by the "obvious" complement, with other more contrived translations supported by equally contrived complements. In [Heg04], it is argued that there is a key additional property which "good" views have; namely, that they are monotonic with respect to the natural order on states. In the context of the relational algebra, this amounts to excluding negation. With this additional condition enforced, it has been shown that the translation of a view update to the main schema is independent of the choice of complement [Heg04, 4.3], with the limitation that the updates themselves are *order realizable*; that is, realizable as sequences of legal insertions and deletions.

In this paper, the issue of relaxing this limitation to order-based updates is addressed with the context of the classical relational model. Rather than using a syntactic notion of monotonicity based upon the order structure of database states, a semantic formulation, which characterizes the *information content* of a database state in terms of the set of sentences in a particular family which it satisfies, is employed. The decomposition mapping is then required to be not only bijective in the ordinary sense but also a *semantic bijection*, in the sense that it defines a bijection between (semantic equivalence classes of) the sentences which define the information content. Under this requirement, it is shown that constant-complement translations of all view updates, order based or not, are always information optimal, and hence unique and independent of the choice of complement. It is furthermore shown that when the main schema is constrained by a wide range of classical database dependencies, and the view mappings are SPJ-mappings, that is, conjunctive queries, these conditions are always satisfied. Interestingly, this result is not limited to meet-complementary pairs in the sense of [Heg04, 2.12]; that is, pairs of view which define well-behaved families of updates. Rather, it applies to any pair of complementary views whose decomposition mapping is a semantic bijection, and any update defined via constant-complement within that context.

This work is based upon the notions of information content and optimal translations which were introduced in [Heg08]. In that work, it is argued that view updates should be *information optimal*, in that the correct reflection should entail the least information change over all possibilities. However, the information measure in that case allows for the equivalence of updates which are isomorphic in a certain sense. This work applies that philosophy under the additional constraint that the update strategy should be defined by constant complement, with the reflected updates to the main schema truly unique, and not just isomorphic.

Although the results are vastly different, some of the background material of this paper is similar or identical to that of [Heg08]. Specifically, much of Section 2, as well as some content of 3.2, 4.2, 4.3, 4.17 and 4.19 is adapted from [Heg08], although numerous changes in detail and often simplifications have been made to accommodate the specific needs of this work.

2. The Relational Model

The results of this paper are formulated within the relational model, and familiarity with its standard notions, as presented in references such as [PDGV89] and [AHV95], is assumed. Nevertheless, there are aspects which must be formulated with particular care. Most important are the need to take all relational schemata over the same domain, with the same constant symbols, and the need to express databases themselves as sets of ground atoms. For this reason, the key features which are unique to this formulation are presented in this section.

2.1 Relational contexts and constant interpretations A relational context contains the logical information which is shared amongst the schemata and database mappings. Formally, a relational context $\mathcal{D}$ consists of a finite nonempty set $\mathcal{A}_{\mathcal{D}}$ of attribute names, a countable set $\mathsf{Vars}(\mathcal{D})$ of variables, and for each $A \in \mathcal{A}_{\mathcal{D}}$, an at-most-countable set $\mathsf{Const}_{\mathcal{D}}(A)$ of constant symbols, with $\mathsf{Const}(\mathcal{D}) = \bigcup\{\mathsf{Const}_{\mathcal{D}}(A) \mid A \in \mathcal{A}_{\mathcal{D}}\}$. The variables in $\mathsf{Vars}(\mathcal{D})$ are further partitioned into two disjoint sets; a countable set $\mathsf{GenVars}(\mathcal{D}) = \{x_0, x_1, x_2, \ldots\}$ of *general variables*, and special $\mathcal{A}_{\mathcal{D}}$-indexed set $\mathsf{AttrVars}(\mathcal{D}) = \{x_A \mid A \in \mathcal{A}_{\mathcal{D}}\}$ of *attribute variables*. The latter is used in the definition of interpretation mappings; see 2.6 for details.

Databases are represented as ground atoms, as elaborated in 2.2 below. Therefore, it is necessary that each domain element be bound to a unique constant symbol. Formally, a *constant interpretation* for the relational context $\mathcal{D}$ is a pair $I = (\mathsf{Dom}_I, \mathsf{IntFn}_I)$ in which Dom_I is a countably infinite set, called the *domain* of I, and $\mathsf{IntFn}_I : \mathsf{Const}(\mathcal{D}) \to \mathsf{Dom}_I$ is a bijective function, called the *interpretation function* of I. This effectively stipulates the following two well-known conditions [GN87, p. 120]:

Domain closure: $(\forall x)(\bigvee_{a \in \mathsf{Const}(\mathcal{D})} x = a)$ $\qquad (\mathsf{DCA}(\mathcal{D}))$

Unique naming: $(\neg(a = b))$ for distinct $a, b \in \mathsf{Const}(\mathcal{D})$ $\qquad (\mathsf{UNA}(\mathcal{D}))$

Since there are countably many constant symbols, the domain-closure axiom is not a finite disjunction. This is not a problem however, since it is never used in a context in which a first-order constraint is necessary. Because the assignment of domain values to constants is fixed, it is not necessary to verify independently that it holds.

As a notational convention, from this point on, unless stated otherwise, fix a relational context $\mathcal{D}$ and a constant interpretation $I = (\mathsf{Dom}_I, \mathsf{IntFn}_I)$ for it.

2.2 Tuples and databases An *unconstrained relational schema* over $(\mathcal{D}, I)$ is a pair $\mathbf{D} = (\mathsf{Rels}(\mathbf{D}), \mathsf{Ar}_{\mathbf{D}})$ in which $\mathsf{Rels}(\mathbf{D})$ is finite set of relational symbols and $\mathsf{Ar}_{\mathbf{D}} : \mathsf{Rels}(\mathbf{D}) \to \mathbf{2}^{\mathcal{A}_{\mathcal{D}}}$ a function which assigns an *arity*, a set of distinct attributes from $\mathcal{A}_{\mathcal{D}}$, to each $R \in \mathsf{Rels}(\mathbf{D})$.

A *ground R-atom* is a function $t : \mathsf{Ar}_{\mathbf{D}}(R) \to \mathsf{Const}(\mathcal{D})$ with the property that $t[A] \in \mathsf{Const}_{\mathcal{D}}(A)$. The set of all ground R-atoms is denoted $\mathsf{GrAtoms}(R, \mathbf{D})$. A *ground* $\mathbf{D}$*-atom* is a ground R-atom for some $R \in \mathsf{Rels}(\mathbf{D})$, with the set of all ground $\mathbf{D}$-atoms denoted $\mathsf{GrAtoms}(\mathbf{D})$. An *atom database for* $\mathbf{D}$ is a finite subset of $\mathsf{GrAtoms}(\mathbf{D})$, with the set of all atom databases for $\mathbf{D}$ denoted $\mathsf{DB}(\mathbf{D})$.

An *R-atom t* is defined similarly, except that $t[A]$ is not required to be a constant; rather, $t[A] \in \mathsf{Const}_{\mathcal{D}}(A) \cup \mathsf{GenVars}(\mathcal{D}) \cup \{x_A\}$. The $\mathbf{D}$*-atoms* and the set $\mathsf{Atoms}(\mathbf{D})$ are defined in the obvious way.

It is convenient to be able to recover the associated relation name from a tuple, and so *tagging* is employed, in which tuples are marked with the relation name. Formally, this is accomplished by introducing a new attribute $\mathsf{RName} \notin \mathcal{A}_{\mathcal{D}}$, and then regarding a ground R-atom not as a function t just on $\mathsf{Ar}_{\mathbf{D}}(R)$ but rather as one on $\{\mathsf{RName}\} \cup \mathsf{Ar}_{\mathbf{D}}(R)$ with the property that $t[\mathsf{RName}] = R$. Tagging of R-atoms will be used from this point on throughout the paper. Nevertheless, in writing such atoms, the more conventional notation $R(a_1, a_2, \ldots, a_n)$ will be used in lieu of the technically more correct $(R, a_1, a_2, \ldots, a_n)$, although tags will be used in formal constructions.

2.3 Formulas and constraint classes The first-order language associated with the relational schema $\mathbf{D}$ is defined in the natural way; however, it is useful to introduce some notation which identifies particular sets of formulas. Define $\mathsf{WFF}(\mathbf{D})$ to be the set of all well-formed first-order formulas with equality in the language whose set of relational symbols is $\mathsf{Rels}(\mathbf{D})$, whose set of constant symbols is $\mathsf{Const}(\mathcal{D})$, and which contains no non-nullary function symbols. The variables are those of $\mathcal{D}$; these formulas are typed only to the extent that for $A \in \mathcal{A}_{\mathcal{D}}D$, the variable x_A may only occur in a $\mathbf{D}$-atom in a position associated with attribute A. In other words, the conditions for $\mathbf{D}$-atoms identified in 2.2 are enforced. $\mathsf{WFS}(\mathbf{D})$ denotes the subset of $\mathsf{WFF}(\mathbf{D})$ consisting of sentences; that is, formulas with no free variables.

A *constraint class* $\mathcal{C}$ identifies a subset of $\mathsf{WFF}(\mathbf{D})$, denoted $\mathsf{WFF}(\mathbf{D}, \mathcal{C})$. Of particular interest in this work are $\exists{\neq}$, $\exists{+}$, $\exists{\wedge}{+}$, Atoms, and $\mathbf{1}$, defined as follows.

- $\mathsf{WFF}(\mathbf{D}, \exists{\neq})$ is the subset of $\mathsf{WFF}(\mathbf{D})$ consisting of those formulas in which only existential quantification is allowed, and in which negation (explicit or implicit) occurs only at the level of equality atoms. More precisely, negation may only occur in the form $\neg(\tau_1 = \tau_2)$, with τ_1 and τ_2 terms. Negation of other atoms, such as in $(\exists x_1)(\exists x_2)(R(x) \wedge (\neg S(x))$, is prohibited.
- $\mathsf{WFF}(\mathbf{D}, \exists{+})$ is the subset of $\mathsf{WFF}(\mathbf{D}, \exists{\neq})$ in which no negation at all is allowed.
- $\mathsf{WFF}(\mathbf{D}, \exists{\wedge}{+})$ is the subset of $\mathsf{WFF}(\mathbf{D}, \exists{+})$ in which disjunction is also disallowed, so that the only logical connective which is allowed is conjunction. These formulas define the so-called *conjunctive queries* [CGT90, Sec. 4.2].
- $\mathsf{WFF}(\mathbf{D}, \mathsf{Atoms})$ is just $\mathsf{Atoms}(\mathbf{D})$.
- $\mathsf{WFF}(\mathbf{D}, \mathbf{1})$ is shorthand for $\mathsf{WFF}(\mathbf{D})$.

In each case, the corresponding set $\mathsf{WFS}(\mathbf{D}, \mathcal{C})$ of sentences is defined in the obvious way. In particular, note that $\mathsf{WFS}(\mathbf{D}, \mathsf{Atoms}) = \mathsf{GrAtoms}(\mathbf{D})$.

2.4 Atomic models Even though databases are represented as sets of ground atoms, and not as interpretations in the usual logical sense, it is still essential to have an appropriate notion of model for a given sentence. This is relatively straightforward; a model for a sentence φ is a database which is consistent with both φ and the unique-naming axioms. There is one complication, however. In representing a database as a set of $\mathbf{D}$-atoms, the closed-world assumption is implicit. On the other hand, to express what it means for such a representation to satisfy an arbitrary sentence in $\mathsf{WFS}(\mathbf{D})$, it is necessary to state explicitly which atoms are not true as well. Formally, for $M \in \mathsf{DB}(\mathbf{D})$, define the *diagram* of M to be $\mathsf{Diagram}_{\mathbf{D}}(M) = M \cup \{\neg t \mid t \in \mathsf{GrAtoms}(\mathbf{D}) \setminus M\}$. Now, say that $M \in \mathsf{DB}(\mathbf{D})$ is an *atomic I-model* of $\varphi \in \mathsf{WFS}(\mathbf{D})$ if $\mathsf{Diagram}_{\mathbf{D}}(M) \cup \{\varphi\} \cup \mathsf{UNA}(\mathcal{D})$ is consistent. $\mathsf{AtMod}_I(\varphi)$ denotes the set of all atomic I-models of φ, with $\mathsf{AtMod}_I(\Phi) = \bigcap\{\mathsf{AtMod}_I(\varphi) \mid \varphi \in \Phi\}$ for $\Phi \subseteq \mathsf{WFS}(\mathbf{D})$.

2.5 Schemata with constraints and constrained databases To obtain full relational schemata, constraints are added to the unconstrained schemata of 2.2. Formally, a *relational schema* over $(\mathcal{D}, I)$ is a triple $\mathbf{D} = (\mathsf{Rels}(\mathbf{D}), \mathsf{Ar}_{\mathbf{D}}, \mathsf{Constr}(\mathbf{D}))$ in which $(\mathsf{Rels}(\mathbf{D}), \mathsf{Ar}_{\mathbf{D}})$ is an unconstrained relational schema over $(\mathcal{D}, I)$ and $\mathsf{Constr}(\mathbf{D}) \subseteq \mathsf{WFS}(\mathbf{D})$ is the set of *dependencies* or *constraints* of $\mathbf{D}$.

Define the *legal* (or *constrained*) *databases* $\mathsf{LDB}(\mathbf{D})$ of $\mathbf{D}$ to be $\mathsf{AtMod}_I(\mathsf{Constr}(\mathbf{D}))$.

Define the equivalence relation $\equiv_{\mathbf{D}}$ on $\mathsf{WFS}(\mathbf{D})$ by $\varphi_1 \equiv_{\mathbf{D}} \varphi_2$ iff $\mathsf{AtMod}_I(\varphi_1) \cap \mathsf{LDB}(\mathbf{D}) = \mathsf{AtMod}_I(\varphi_2) \cap \mathsf{LDB}(\mathbf{D})$ or, equivalently, $\mathsf{AtMod}_I(\{\varphi_1\} \cup \mathsf{Constr}(\mathbf{D})) = \mathsf{AtMod}_I(\{\varphi_2\} \cup \mathsf{Constr}(\mathbf{D}))$. Thus, $\equiv_{\mathbf{D}}$ identifies sentences which have identical truth values on all $M \in \mathsf{LDB}(\mathbf{D})$. The equivalence class of φ_1 under $\equiv_{\mathbf{D}}$ is denoted $[\varphi_1]_{\equiv_{\mathbf{D}}}$.

2.6 Database morphisms and views Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be relational schemata over $(\mathcal{D}, I)$. There are two fundamental ways to represent a database morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ in the relational context. On the one hand, such a morphism may be represented as a function $f : \mathsf{DB}(\mathbf{D}_1) \to \mathsf{DB}(\mathbf{D}_2)$, using expressions from the relational algebra. On the other hand, by providing an interpretation formula $f^R \in \mathsf{WFF}(\mathbf{D}_1)$ for each $R \in \mathsf{Rels}(\mathbf{D}_2)$, the morphism may be represented using the relational calculus [JAK82]. The equivalence of these two representations is one of the classical results of relational database theory [PDGV89, Sec. 2.4-2.6]. The interpretation formulation is taken as the basic one in this work. Formally, given $R \in \mathsf{Rels}(\mathbf{D}_2)$, an *interpretation* for R into $\mathbf{D}_1$ is a $\varphi \in \mathsf{WFF}(\mathbf{D})$ in which precisely the variables $\{x_A \mid A \in \mathsf{Ar}_{\mathbf{D}}(R)\}$ are free, with x_A is used to mark the position in the formula which is bound to attribute A. The set of all interpretations of R into $\mathbf{D}_1$ is denoted $\mathsf{Interp}(R, \mathbf{D}_1)$. A *syntactic morphism* $f : \mathbf{D}_1 \to \mathbf{D}_2$ is a family $f = \{f^R \mid R \in \mathsf{Rels}(\mathbf{D}_2)$ and $f^R \in \mathsf{Interp}(R, \mathbf{D}_1)\}$.

Let $t \in \mathsf{Atoms}(R, \mathbf{D}_2)$. The *substitution* of t into f, denoted $\mathsf{Subst}\langle f, t\rangle$, is the formula in $\mathsf{WFF}(\mathbf{D}_1)$ obtained by substituting $t[A]$ for x_A, for each $A \in \mathsf{Ar}_{\mathbf{D}}(R)$. Note that If t is a ground atom, then $\mathsf{Subst}\langle f, t\rangle \in \mathsf{WFS}(\mathbf{D}_1)$.

For $M \in \mathsf{DB}(\mathbf{D}_1)$, define $f(M) = \{t \in \mathsf{GrAtoms}(\mathbf{D}_2) \mid M \in \mathsf{AtMod}_I(\mathsf{Subst}\langle f, t\rangle)\}$. f is called an LDB-*morphism* if it maps legal databases to legal databases; formally, an LDB-morphism has the property that $f(M) \in \mathsf{LDB}(\mathbf{D}_2)$ for each $M \in \mathsf{LDB}(\mathbf{D}_1)$. When no qualification is given, *database morphism* will always mean LDB-morphism.

Let $\mathbf{D}$ be a relational schema over $(\mathcal{D}, I)$. A *(relational) view* of $\mathbf{D}$ is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is a relational schema over $(\mathcal{D}, I)$ and $\gamma : \mathbf{D} \to \mathbf{V}$ is an LDB-morphism which is furthermore LDB-

surjective in the sense that for every $N \in \mathsf{LDB}(\mathbf{V})$, there is an $M \in \mathsf{LDB}(\mathbf{D})$ with $\gamma(M) = N$. Surjectivity is required because the state of the view must always be determined by the state of the main schema $\mathbf{D}$.

3. Updates and View Complements

In this section, some basic definitions and notation regarding updates and their reflections are presented, as are the key ideas surrounding the constant-complement update strategy.

3.1 Notational convention Throughout this section, take $\mathbf{D}$ to be a relational schema over $(\mathcal{D}, I)$.

3.2 Updates and reflections Let $\Gamma = (\mathbf{V}, \gamma)$ be a view on $\mathbf{D}$. An *update* on $\mathbf{D}$ is a pair $(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$. M_1 is the current state, and M_2 the new state. To describe the situation surrounding an update request on Γ, it is sufficient to specify the current state M_1 of the main schema and the desired new state N_2 of the view schema $\mathbf{V}$. The current state of the view can be computed as $\gamma(M_1)$; it is only the new state M_2 of the main schema (subject to $N_2 = \gamma(M_2)$) which must be obtained from an update strategy. Formally, an *update request* from Γ to $\mathbf{D}$ is a pair (M_1, N_2) in which $M_1 \in \mathsf{LDB}(\mathbf{D})$ (the old state of the main schema) and $N_2 \in \mathsf{LDB}(\mathbf{V})$ (the new state of the view schema). A *realization* of (M_1, N_2) along Γ is an update (M_1, M_2) on $\mathbf{D}$ with the property that $\gamma(M_2) = N_2$. The update (M_1, M_2) is called a *reflection* (or *translation*) of the view update $(\gamma(M_1), N_2)$. The set of all realizations of (M_1, N_2) along Γ is denoted $\mathsf{UpdRealiz}\langle M_1, N_2, \Gamma\rangle$.

3.3 Disjoint union In the construction of complementary views, the disjoint union of two sets will be used to construct the coproduct of views. As the symbol $\coprod$ will be reserved to denote a formal coproduct, $\uplus$ will be used to represent the disjoint union of two sets. Thus, $A \uplus B$ is just a "tagged" version of $A \cup B$, in which it is possible to determine from which of the two sets an element arose. Note that it is possible for there to be two instances of a given element x in $A \uplus B$, one tagged with A and the other tagged with B.

3.4 The coproduct of two views The coproduct of two views is the natural one which arises when complementation is considered. (The terminology of and notation for coproduct is used because this construction is a true coproduct in the categorical sense [HS73, §18]). Basically, the set of relations of the coproduct schema is the disjoint union of those of the two component schemata, with each such relation retaining its interpretation function. The databases are defined similarly via disjoint union. (Recall that tuples are tagged (see 2.2), so a database for a schema consists of just one big set of tuples.) Formally, let $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of $\mathbf{D}$. The *coproduct* of Γ_1 and Γ_2 is the view $\Gamma_1 \coprod \Gamma_2 = (\mathbf{V}_1 \coprod \mathbf{V}_2, \gamma_1 \coprod \gamma_2)$, defined as follows.

(a) $\mathsf{Rels}(\mathbf{V}_1 \coprod \mathbf{V}_2) = \mathsf{Rels}(\mathbf{V}_1) \uplus \mathsf{Rels}(\mathbf{V}_2)$.

(b) For $i \in \{1, 2\}$ and $R \in \mathsf{Rels}(\mathbf{V}_i)$, $(\gamma_1 \coprod \gamma_2)^R = \gamma_i^R$.

(c) $\mathsf{LDB}(\mathbf{V}_1 \coprod \mathbf{V}_2) = \{\gamma_1(M) \uplus \gamma_2(M) \mid M \in \mathsf{LDB}(\mathbf{D})\}$.

(d) $\mathsf{Constr}(\mathbf{V}_1 \coprod \mathbf{V}_2)$ is the set of all first-order sentences which define the constraints on $\mathsf{LDB}(\mathbf{V}_1 \coprod \mathbf{V}_2)$.

Note that, in view of (c), $\gamma_1 \coprod \gamma_2$ is surjective by construction. Hence, there is no question that $\gamma_1 \coprod \gamma_2$ is a view.

In general, there is no straightforward representation for $\mathsf{Constr}(\mathbf{V}_1 \coprod \mathbf{V}_2)$, the set of constraints of this coproduct. Nevertheless, it can be shown that it has a representation as a set of first-order sentences [Mon76, Ch. 26]. Since this representation is not central to the theme of this paper, it will not be elaborated further.

It should perhaps also be noted that, strictly speaking, the notation $\mathbf{V}_1 \coprod \mathbf{V}_2$ is incomplete, since this product depends not only upon $\mathbf{V}_1$ and $\mathbf{V}_2$, but upon the view morphisms γ_1 and γ_2 as well. However, since no confusion can result, a more accurate but correspondingly cumbersome notation will not be introduced.

3.5 Constant-complement realizations and complementary pairs Although the ideas surrounding constant-complement update are well known [BS81] [Heg04], it is important to formalize them within the current context. Let Γ_1 and Γ_2 be views of $\mathbf{D}$.

(a) $\{\Gamma_1, \Gamma_2\}$ forms a *complementary pair* if the underlying function $\gamma_1 \coprod \gamma_2 : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}_1 \coprod \mathbf{V}_2)$ which sends $M \mapsto \gamma_1(M) \uplus \gamma_2(M)$ is injective (and hence bijective).

(b) For (M_1, N_2) an update request from Γ_1 to $\mathbf{D}$, $(M_1, M_2) \in \mathsf{UpdRealiz}\langle M_1, N_2, \Gamma_1 \rangle$ is called a Γ_2-*constant realization* of (M_1, N_2) if $\gamma_2(M_1) = \gamma_2(M_2)$.

The following classical observation [BS81, Sec. 5], which follows immediately from the injectivity of $\gamma_1 \coprod \gamma_2$, is key to the entire strategy.

3.6 Observation *Let $\{\Gamma_1, \Gamma_2\}$ be a complementary pair, and let (M_1, N_2) be an update request from Γ_1 to $\mathbf{D}$. Then there is at most one Γ_2-constant realization of (M_1, N_2), and this realization exists iff there is an $M_2 \in \mathsf{LDB}(\mathbf{D})$ such that $(\gamma_1 \coprod \gamma_2)(M_2) = N_2 \uplus \gamma_2(M_1)$. In this case, the unique realization is given by (M_1, M_2).* $\square$

4. The Theory of Unique Reflections

In this section, the central results on uniqueness of constant-complement translations are developed.

4.1 Notational convention Throughout this section, unless stated specifically to the contrary, take $\mathbf{D}$, $\mathbf{D}_1$, and $\mathbf{D}_2$ to be a relational schema over $(\mathcal{D}, I)$, $\Gamma = (\mathbf{V}, \gamma)$, $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$, and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ to be views on $\mathbf{D}$, with $\mathcal{C}$ a constraint class.

4.2 Information content and separation A central theme of this work is that the information content of a database may be characterized by the set of sentences from a particular set which it satisfies. Let $\Sigma \subseteq \mathsf{WFS}(\mathbf{D})$ and let $M \in \mathsf{DB}(\mathbf{D})$. The *information content* of M relative to Σ is the set of all sentences in Σ which are true for M. More precisely, $\mathsf{Info}\langle M, \Sigma \rangle = \{\varphi \in \Sigma \mid M \in \mathsf{AtMod}_I(\varphi)\}$. $M_1, M_2 \in \mathsf{DB}(\mathbf{D})$ are Σ-*equivalent* if they have the same information content relative to Σ; *i.e.*, $\mathsf{Info}\langle M_1, \Sigma \rangle = \mathsf{Info}\langle M_2, \Sigma \rangle$. Σ is *separating* for $\mathbf{D}$ if whenever $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$ are Σ-equivalent, it must be the case that $M_1 = M_2$.

4.3 Information-monotone sentences Intuitively, if $M_1 \subseteq M_2$, then it should be the case that M_2 contains more information than M_1; *i.e.*, $\mathsf{Info}\langle M_1,\Sigma\rangle \subseteq \mathsf{Info}\langle M_2,\Sigma\rangle$. However, in general, M_1 will satisfy constraints which M_2 does not; for example, if $t \in M_2 \setminus M_1$, then $\neg t$ is true of M_1 but not of M_2. If such negative constraints are excluded, then Σ is termed information monotone. More formally, The sentence $\varphi \in \mathsf{WFS}(\mathbf{D})$ is *information monotone* if for any $M_1,M_2 \in \mathsf{DB}(\mathbf{D})$ if $M_1 \subseteq M_2$, then $\mathsf{Info}\langle M_1,\{\varphi\}\rangle \subseteq \mathsf{Info}\langle M_2,\{\varphi\}\rangle$. In other words, φ is information monotone if $\mathsf{AtMod}_I(\varphi)$ is closed under supersets within $\mathsf{DB}(\mathbf{D})$; whenever $M \in \mathsf{AtMod}_I(\varphi)$, then all $M' \in \mathsf{DB}(\mathbf{D})$ with $M \subseteq M'$ are also in $\mathsf{AtMod}_I(\varphi)$. The set $\Sigma \subseteq \mathsf{WFS}(\mathbf{D})$ is *information monotone* if each $\varphi \in \Sigma$ has this property.

4.4 Proposition — Key instances of information monotone families *Any subset of* $\mathsf{WFS}(\mathbf{D},\exists\neq)$ *is information monotone and separating. This includes, in particular,* $\mathsf{WFS}(\mathbf{D},\exists+)$*,* $\mathsf{WFS}(\mathbf{D},\exists\wedge+)$*, and* $\mathsf{GrAtoms}(\mathbf{D})$.

PROOF: It is immediate that any formula involving only existential quantification and not involving any negation is information monotone. That allowing negations of equality atoms does not violate information monotonicity follows from the fact that the equality relation is fixed over all databases. To see this more clearly, consider adding a new, fixed relation $\neq$ which represents inequality explicitly, and replacing each atom of the form $\neg(\tau_1 = \tau_2)$ with $\neq(\tau_1,\tau_2)$. Then, all negation can be removed from the sentences, and so any such subset is information monotone.

To complete the proof, it suffices to observe that any subset of $\mathsf{WFS}(\mathbf{D})$ which contains $\mathsf{GrAtoms}(\mathbf{D})$ is separating, and $\mathsf{GrAtoms}(\mathbf{D})$ is contained in both $\mathsf{WFS}(\mathbf{D},\exists+)$ and $\mathsf{WFS}(\mathbf{D},\exists\wedge+)$. $\square$

The proof of the following observation is immediate, but the statement is of such central importance that it is worth noting explicitly.

4.5 Observation — $\mathsf{Info}\langle M,\Sigma\rangle$ determines M for Σ separating *Let Σ be an information monotone and separating family on* $\mathbf{D}$.

(a) *For any $M \in \mathsf{LDB}(\mathbf{D})$, M is the least element (under $\subseteq$) of* $\mathsf{AtMod}_I(\mathsf{Info}\langle M,\Sigma\rangle) \cap \mathsf{LDB}(\mathbf{D})$.

(b) *If $M_1,M_2 \in \mathsf{LDB}(\mathbf{D})$, then $M_1 \subseteq M_2$ iff* $\mathsf{Info}\langle M_1,\Sigma\rangle \subseteq \mathsf{Info}\langle M_2,\Sigma\rangle$. $\square$

A central premise of this work is that it is advantageous to view database morphisms as mappings between sentences (in an appropriate information-monotone family), rather than just as mappings from databases to databases. The following definition formalizes this idea.

4.6 Substitution of sentences Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism. The association $t \mapsto \mathsf{Subst}\langle f,t\rangle$ defined in 2.6 may be extended in a natural way to all of $\mathsf{WFS}(\mathbf{D}_2)$. Specifically, the *interpretation of φ in f* is the sentence $\mathsf{Subst}\langle f,\varphi\rangle \in \mathsf{WFS}(\mathbf{D}_1)$ which is obtained by first renaming all quantified variables so that no two formulas involved in the construction have any such variables in common, and then replacing each atom ψ which occurs in φ with $\mathsf{Subst}\langle f,\psi\rangle$. As a specific example, suppose that $\mathbf{D}_1$ has two relation symbols $R_{11}[ABD]$ and $R_{12}[DBC]$, and that $\mathbf{D}_2$ has two relation symbols $R_{21}[AB]$ and $R_{22}[BC]$. Let the defining formulas be $f^{R_{21}} = (\exists x_1)(R_{11}(x_A,x_B,x_1))$ and $f^{R_{22}} = (\exists x_2)(\exists x_3)(R_{11}(x_2,x_B,x_3) \wedge R_{12}(x_3,x_B,x_C))$, with the sentence to be interpreted $\varphi = (\exists x_4)(R_{21}(a,x_4) \wedge R_{22}(x_4,c))$. Here a and c are constants. Then

$$\mathsf{Subst}\langle f,\varphi\rangle = (\exists x_1)(\exists x_2)(\exists x_3)(\exists x_4)(R_{11}(a,x_4,x_1) \wedge R_{11}(x_2,x_4,x_3) \wedge R_{12}(x_3,x_4,c)).$$

For more detailed examples, see [JAK82, Sec. 3].

Think of this definition as specifying a function $\mathsf{Subst}\langle f, -\rangle : \mathsf{WFS}(\mathbf{D}_2) \to \mathsf{WFS}(\mathbf{D}_1)$. The key feature to keep in mind is that while the underlying function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ maps databases of $\mathbf{D}_1$ to databases of $\mathbf{D}_2$, the interpretation mapping $\mathsf{Subst}\langle f, -\rangle$ sends sentences in $\mathsf{WFS}(\mathbf{D}_2)$ to sentences in $\mathsf{WFS}(\mathbf{D}_1)$. Thus, the direction is reversed. It should perhaps be noted that this definition extends easily to well-formed formulas which are not sentences, but since such an extension is not needed here, it will not be elaborated.

The relationship between the mapping of databases and the mapping of models is a close one, as shown by the following observation, whose straightforward proof is omitted.

4.7 Observation *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism, and let $M \in \mathsf{DB}(\mathbf{D}_1)$. Then for any $\varphi \in \mathsf{WFS}(\mathbf{D}_2)$, $f(M) \in \mathsf{AtMod}_I(\varphi)$ iff $M \in \mathsf{AtMod}_I(\mathsf{Subst}\langle f, \varphi\rangle)$.* $\square$

4.8 Schemata, morphisms, and views of class $\mathcal{C}$ To measure information content using the sentences of class $\mathcal{C}$, it is important that the database schemata and views respect that class in a certain way.

(a) The database schema $\mathbf{D}$ is *of class* $\mathcal{C}$ if $\mathsf{WFS}(\mathbf{D}, \mathcal{C})$ is separating for $\mathbf{D}$.

(b) The database morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ is *of class* $\mathcal{C}$ if for every $\varphi \in \mathsf{WFF}(\mathbf{D}_2, \mathcal{C})$, $\mathsf{Subst}\langle f, \varphi\rangle \in \mathsf{WFF}(\mathbf{D}_1, \mathcal{C})$.

(c) The view $\Gamma = (\mathbf{V}, \gamma)$ is *of class* $\mathcal{C}$ if both $\mathbf{V}$ and γ are of that class.

In view of 4.4, any $\mathcal{C}$ which contains $\mathsf{GrAtoms}(\mathbf{D})$ is separating, and hence any database schema $\mathbf{D}$ is of such a class $\mathcal{C}$. However, the notion of a database morphism being of class $\mathcal{C}$ is more complex, and warrants closer attention via a few examples.

4.9 Examples — Morphisms of class $\mathcal{C}$ Let $\mathbf{E}_1$ be the relational schema with two ternary relation symbol $R_{11}[ABC]$ and $R_{12}[ABC]$, and let $\mathbf{E}_2$ be the schema with the single relation symbol $R_2[AB]$. Define the morphism $g_{11} : \mathbf{E}_1 \to \mathbf{E}_2$ by $g_{11}^{R_2} = (\exists z)(R_{11}(x_A, x_B, z) \wedge R_{12}(x_A, x_B, z))$. In other words, R_2 is the projection of the intersection of R_{11} and R_{12}. Then g_{11} is of class $\mathcal{C}$ for $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathbf{1}\}$, but not for $\mathcal{C} = \mathsf{Atoms}$, since $\mathsf{Subst}\langle g_{11}^{R_2}, t\rangle$ is not equivalent to a ground atom for $t \in \mathsf{GrAtoms}(\mathbf{E}_2)$. On the other hand, define $g_{12} : \mathbf{E}_1 \to \mathbf{E}_2$ by $g_{12}^{R_2} = (\exists z)(R_{11}(x_A, x_B, z) \wedge (\neg R_{12}(x_A, x_B, z)))$. In this case, R_2 is the projection of the difference of R_{11} and R_{12}, and g_{12} is of class $\mathcal{C}$ for $\mathcal{C} = \mathbf{1}$, but not for any of the others listed above, since the sentence $\mathsf{Subst}\langle g_{12}^{R_2}, \varphi\rangle$ will always contain non-removable internal negation, even for φ a ground atom. In particular, g_{12} is not of class $\exists\wedge+$. Finally, define $g_{13} : \mathbf{E}_1 \to \mathbf{E}_2$ by $g_{13}^{R_2} = (\exists z)(R_{11}(x_A, x_B, z) \vee R_{12}(x_A, x_B, z))$. In other words, R_2 is the projection of the union of R_{11} and R_{12}. Then g_{13} is of class $\mathcal{C}$ for $\mathcal{C} \in \{\exists\neq, \exists+, \mathbf{1}\}$, but not for $\mathcal{C} \in \{\exists\wedge+, \mathsf{Atoms}\}$, since $\mathsf{Subst}\langle g_{13}, \varphi\rangle$ will always contain non-removable disjunction, even for φ a ground atom.

4.10 Notational convention For the remainder of this section, unless stated specifically to the contrary, take all database schemata to be of class $\mathcal{C}$.

4.11 Semantic morphisms Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism of class $\mathcal{C}$. In addition to the standard notions of injectivity, surjectivity, and bijectivity for the induced mapping $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$, there are corresponding notions associated with the substitution mapping $\mathsf{Subst}\langle f, -\rangle : \mathsf{WFS}(\mathbf{D}_2, \mathcal{C}) \to \mathsf{WFS}(\mathbf{D}_1, \mathcal{C})$. The point of care to be taken is that the identification is only unique up to the equivalence $\equiv_{\mathbf{D}}$ as defined in 2.5.

(a) f is *semantically injective for $\mathcal{C}$* if for every $\varphi_1, \varphi_2 \in \mathsf{WFS}(\mathbf{D}_2, \mathcal{C})$, if $\mathsf{Subst}\langle f, \varphi_1\rangle \equiv_{\mathbf{D}_1} \mathsf{Subst}\langle f, \varphi_2\rangle$, then $\varphi_1 \equiv_{\mathbf{D}_2} \varphi_2$.

(b) f is *semantically surjective for $\mathcal{C}$* if for every $\varphi_1 \in \mathsf{WFS}(\mathbf{D}_1, \mathcal{C})$, there is a $\varphi_2 \in \mathsf{WFS}(\mathbf{D}_2, \mathcal{C})$ with $\mathsf{Subst}\langle f, \varphi_2\rangle \equiv_{\mathbf{D}_1} \varphi_1$.

(c) f is *semantically bijective for $\mathcal{C}$* if it is both semantically injective and semantically surjective for $\mathcal{C}$.

A view $\Gamma = (\mathbf{V}, \gamma)$ is said to be *semantically bijective for $\mathcal{C}$* precisely when the morphism γ has that property.

It should be stressed that for f to be of class $\mathcal{C}$ is *causa sine qua non* to have any of the above properties. If f is not of class $\mathcal{C}$, then it is not semantically injective, surjective, or bijective for $\mathcal{C}$, by definition.

It is easy to see that the morphisms g_{11}, g_{12}, and g_{13} of 4.9 are semantically injective; that is, logically distinct formulas in $\mathbf{E}_2$ give rise to logically distinct formulas in $\mathbf{E}_1$. This is true more generally; surjectivity of the underlying database mapping translates to semantic injectivity *provided the morphism is of the appropriate class*.

4.12 Proposition — Underlying surjectivity $\Rightarrow$ semantic injectivity *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism of class $\mathcal{C}$. If the associated function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ is surjective, then f is semantically injective for $\mathcal{C}$. In particular, for every view $\Gamma = (\mathbf{V}, \gamma)$ of class $\mathcal{C}$, the morphism γ is semantically injective for $\mathcal{C}$.*

PROOF: Let $\varphi_1, \varphi_2 \in \mathsf{WFS}(\mathbf{D}_2, \mathcal{C})$ with $\varphi_1 \not\equiv_{\mathbf{D}_2} \varphi_2$. Then there exists an $N \in \mathsf{LDB}(\mathbf{D}_2)$ which is an atomic model of one but not the other. Without loss of generality, assume that $N \in \mathsf{AtMod}_I(\varphi_1) \setminus \mathsf{AtMod}_I(\varphi_2)$. Then for any $M \in f^{-1}(N)$, $M \in \mathsf{AtMod}_I(\mathsf{Subst}\langle f, \varphi_1\rangle) \setminus \mathsf{AtMod}_I(\mathsf{Subst}\langle f, \varphi_2\rangle)$. Hence, $\mathsf{Subst}\langle f, \varphi_1\rangle \not\equiv_{\mathbf{D}_1} \mathsf{Subst}\langle f, \varphi_2\rangle$, and so f is semantically injective. □

For this work, semantic surjectivity on its own is not of central importance. Rather, the key property is semantic bijectivity. Examples illustrating these ideas are found in 4.16 below. First, however, it is essential to introduce some supporting ideas.

The semantic bijectivity of a morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ entails nothing more than a bijective correspondence between the equivalence classes of sentences of class $\mathcal{C}$ in $\mathbf{D}_1$ and those of $\mathbf{D}_2$. This is formulated precisely as follows.

4.13 Observation — Characterization of semantic bijectivity *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be database morphism of class $\mathcal{C}$. Then f if semantically bijective for $\mathcal{C}$ iff it induces a natural bijection between* $\mathsf{WFF}(\mathbf{D}_2, \mathcal{C})/\equiv_{\mathbf{D}_2}$ *and* $\mathsf{WFF}(\mathbf{D}_1, \mathcal{C})/\equiv_{\mathbf{D}_1}$ *via* $[\varphi]_{\equiv_{\mathbf{D}_2}} \mapsto [\mathsf{Subst}\langle f, \varphi\rangle]_{\equiv_{\mathbf{D}_1}}$. □

4.14 The reconstruction morphism Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism. If the induced function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ is bijective, then there is trivially a function $g : \mathsf{LDB}(\mathbf{D}_2) \to \mathsf{LDB}(\mathbf{D}_1)$ for which is inverse to f. What is remarkable is that there is database morphism $g : \mathbf{D}_2 \to \mathbf{D}_1$ in the logical sense which induces g. This is a consequence of Beth's theorem from model theory [Mon76, Thm. 22.4]. This g is called the *reconstruction morphism* for f and is denoted $\mathsf{RcMor}(f)$.

Observe that f and $\mathsf{RcMor}(f)$ are inverses for the mappings on sentences as well, up to the logical equivalence defined by the schemata. More precisely, for any $\varphi_1 \in \mathsf{WFS}(\mathbf{D}_1)$, $\mathsf{Subst}\langle f, \mathsf{Subst}\langle \mathsf{RcMor}(f), \varphi_1\rangle\rangle \equiv_{\mathbf{D}_1} \varphi_1$, and for any $\varphi_2 \in \mathsf{WFS}(\mathbf{D}_2)$, $\mathsf{Subst}\langle \mathsf{RcMor}(f), \mathsf{Subst}\langle f, \varphi_2\rangle\rangle \equiv_{\mathbf{D}_1} \varphi_2$.

Unfortunately, there is no guarantee that $\mathsf{RcMor}(f)$ will be of the same class as f. When it is, however, semantic bijectivity is guaranteed.

4.15 Proposition — Semantic bijectivity $\Leftrightarrow$ class $\mathcal{C}$ reconstruction *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism of class $\mathcal{C}$ whose underlying function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ is bijective. Then f is semantically bijective for $\mathcal{C}$ iff $\mathsf{RcMor}(f)$ is of class $\mathcal{C}$.*

PROOF: Let $\varphi_1 \in \mathsf{WFS}(\mathbf{D}_1, \mathcal{C})$. If $\mathsf{RcMor}(f)$ is of class $\mathcal{C}$, then there is a $\varphi_2 \in \mathsf{WFS}(\mathbf{D}_2, \mathcal{C})$ such that $\mathsf{Subst}\langle \mathsf{RcMor}(f), \varphi_1\rangle \equiv_{\mathbf{D}_2} \varphi_2$, and since $\mathsf{Subst}\langle f, -\rangle \circ \mathsf{Subst}\langle \mathsf{RcMor}(f), -\rangle$ is the identity on $\mathsf{WFS}(\mathbf{D}_1)$, up to equivalence of $\equiv_{\mathbf{D}_1}$, $\mathsf{Subst}\langle f, \varphi_2\rangle \equiv_{\mathbf{D}_1} \varphi_1$. Hence f is semantically surjective, and since it is semantically injective by 4.12, it is semantically bijective for $\mathcal{C}$.

Conversely, if f is semantically bijective for $\mathcal{C}$, then given $\varphi_1 \in \mathsf{WFS}(\mathbf{D}_1, \mathcal{C})$, there is a $\varphi_2 \in \mathsf{WFS}(\mathbf{D}_2, \mathcal{C})$ such that $\mathsf{Subst}\langle f, \varphi_2\rangle \equiv_{\mathbf{D}_1} \varphi_1$. Since $\mathsf{Subst}\langle \mathsf{RcMor}(f), -\rangle \circ \mathsf{Subst}\langle f, -\rangle$ is the identity on $\mathsf{WFS}(\mathbf{D}_2)$, up to equivalence of $\equiv_{\mathbf{D}_2}$, it must be the case that $\mathsf{Subst}\langle \mathsf{RcMor}(f), \varphi_1\rangle \equiv_{\mathbf{D}_2} \varphi_2$, whence $\mathsf{RcMor}(f)$ is of class $\mathcal{C}$. □

4.16 Examples — Semantic bijectivity and reconstruction morphisms It is clear from 4.15 that every bijective morphism is semantically bijective for class $\mathbf{1}$; that is, when all first-order sentences are considered. However, this is not a very useful property within the context of this work, since the set of all first-order sentences on a schema is not information monotone except in trivial cases. In particular, owing to a special property to be developed in 4.18, the choice $\mathcal{C} = \exists\wedge+$ will yield the most fruitful results. It is therefore necessary to establish useful conditions under which semantic bijectivity holds for more restricted classes. To set the stage for this, some illustrative examples based upon a set of four schemata are presented. Let $\mathbf{E}_3$ be the relational schema with three unary relation symbols $R_{31}[A]$, $R_{32}[A]$, and $R_{33}[A]$, constrained by the single sentence $(\forall x)(R_{33}(x) \Leftrightarrow R_{31}(x) \wedge R_{32}(x))$. In other words, the state of R_{33} is the intersection of that of R_{31} and R_{32}. Let $\mathbf{E}_4$ have a corresponding set of three unary relation symbols $R_{41}[A]$, $R_{42}[A]$, and $R_{43}[A]$, but this time constrained by the single sentence $(\forall x)(R_{43}(x) \Leftrightarrow ((R_{41}(x) \wedge \neg R_{42}(x)) \vee (\neg R_{42}(x) \wedge R_{41}(x))))$. In other words, the state of R_{43} is the symmetric difference of that of R_{41} and R_{42}. Define $\mathbf{E}_5$ to have the two unary symbols $R_{51}[A]$ and $R_{52}[A]$, with no other constraints, and define $\mathbf{E}_6$ to have the two unary relation symbols $R_{61}[A]$ and $R_{63}[A]$, again with no other constraints.

First of all, consider the morphism $h_3 : \mathbf{E}_3 \to \mathbf{E}_5$ which identifies R_{51} with R_{31} and R_{52} with R_{32}. Formally, $h_3^{R_{51}} = R_{31}(x_A)$ and $h_3^{R_{52}} = R_{32}(x_A)$. It is trivial that h_3 is of class $\mathcal{C}$ for any $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathsf{Atoms}, \mathbf{1}\}$. The reconstruction mapping for h_3 is $g_3 : \mathbf{E}_5 \to \mathbf{E}_3$ defined by $g_3^{R_{31}} = R_{51}(x_A)$, $g_3^{R_{32}} = R_{52}(x_A)$, and $g_3^{R_{33}} = R_{51}(x_A) \wedge R_{52}(x_A)$. This morphism is of class $\mathcal{C}$ for $\mathcal{C} \in$

$\{\exists\neq, \exists+, \exists\wedge+, \mathbf{1}\}$, but not for $\mathcal{C} = \mathsf{Atoms}$, since the interpretation formula for R_{33} is not equivalent to any atomic formula. Consequently, h_3 and g_3 are semantic bijections for $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathbf{1}\}$.

Next, consider the morphism $h_4 : \mathbf{E}_4 \to \mathbf{E}_5$ which identifies R_{51} with R_{41} and R_{52} with R_{42}. Formally, $h_4^{R_{51}} = R_{31}(x_a)$ and $h_4^{R_{52}} = R_{32}(x_a)$. Just as was the case for h_3, this morphism h_4 is is of class $\mathcal{C}$ for any $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathsf{Atoms}, \mathbf{1}\}$. The reconstruction mapping is $g_4 : \mathbf{E}_5 \to \mathbf{E}_4$ given by $g_4^{R_{41}} = R_{51}(x_A)$, $h_4^{R_{42}} = R_{52}(x_A)$, and $g_4^{R_{43}} = (R_{51}(x_A) \wedge (\neg R_{52}(x_A))) \vee (R_{52}(x_A) \wedge (\neg R_{51}(x_A)))$. This time, amongst the possibilities $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathsf{Atoms}, \mathbf{1}\}$, the reconstruction mapping g_4 is of class $\mathcal{C}$ only for $\mathcal{C} = \mathbf{1}$. The definition of $g_4^{R_{43}}$ excludes all other possibilities.

A similar failure of the reconstruction morphism to be of classes other than $\mathbf{1}$ may occur even when the schemata themselves have no nontrivial constraints. For example, let $h_5 : \mathbf{E}_5 \to \mathbf{E}_6$ be defined by $h_5^{R_{61}} = R_{51}(x_A)$ and $h_5^{R_{63}} = (R_{51}(x_A) \wedge (\neg R_{52}(x_A))) \vee (R_{51}(x_A) \wedge (\neg R_{51}(x_A)))$. In other words, R_{63} constrained by the interpretation morphism to be the symmetric difference of R_{51} and R_{52}. It is easily seen that h_5 is bijective on databases, with the reconstruction morphism $g_5 : \mathbf{E}_6 \to \mathbf{E}_5$ defined by $g_5^{R_{51}} = R_{61}(x_A)$ and $g_5^{R_{52}} = (R_{61}(x_A) \wedge (\neg R_{63}(x_A))) \vee (R_{63}(x_A) \wedge (\neg R_{61}(x_A)))$. However, amongst the possibilities $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathsf{Atoms}, \mathbf{1}\}$, both h_5 and g_5 are of class $\mathcal{C}$ only for $\mathcal{C} = \mathbf{1}$, and so cannot possibly be semantically bijective for any other of the classes.

These examples suggest that for a morphism $f : \mathbf{D}_1 \to \mathbf{D}_2$ which is bijective on databases to fail to be semantically bijective, there must be some nonmonotonicity of information, either in the constraints of the domain schema $\mathbf{D}_1$, or else in the interpretation formulas defined by the interpretation morphism itself. That this is indeed the case, at least for certain common contexts, will now be shown.

4.17 Universal models Traditional database dependencies take the form of generalized universal-existential Horn clauses of the following form.

$$(\forall x_1)(\forall x_2)\ldots(\forall x_n)((A_1 \wedge A_2 \wedge \ldots \wedge A_n) \Rightarrow (\exists y_1)(\exists y_2)\ldots(\exists y_r)(B_1 \wedge B_2 \wedge \ldots \wedge B_s))$$

The A_i's and the B_i's are atoms, but of course not ground atoms. indeed, the quantified variables must occur in these atoms. There is a rich variety of alternatives; for a detailed taxonomy and comparison of properties consult [Fag82].

Given a set S of ground atoms, one may attempt to construct a least model containing S by using so-called forward chaining — repeatedly unifying the left-hand side (the A_i's) with known facts (*qua* ground atoms) in S in order to deduce new ones from the right-hand side (the B_i's). The new facts are added to S and the process is repeated until no new rules apply. This is a classical form of inference for propositional Horn clauses [DG84]. It also applies to so-called universal models the first-order case, but with some limitations. In its general form, it has seen recent application in the context of data exchange [FKMP05] and in the realization of canonical reflections for view updates which lie outside of the scope of constant complement [Heg08]. The process is, in turn, based upon the classical chase inference procedure [BV84].

There are a few complications relative to the propositional setting. First of all, it may be necessary to generate "generic" constants, because of the existential quantifiers, so the least model may only be unique up to a suitable renaming of such constants. Second, the process may not always terminate, but rather continue endlessly to generate new tuples [FKMP05, Example 3.6] [Heg08, 4.14]. Nevertheless, there are wide classes of constraints for which the procedure is known to terminate. One possibility is to work with *full dependencies* which do not involve any existential quantification. A broader solution is to work with the so-called *weakly acyclic* tuple-generating dependencies (tgds), together with the classical equality-generating dependencies (egds) [FKMP05, Thm. 3.9].

When they exist, such universal models have a simple characterization [Heg08, Sec. 3] [FKMP05, Sec. 3.1]. An *endomorphism* on $\mathcal{D}$ is a function $h : \mathsf{Const}(\mathcal{D}) \to \mathsf{Const}(\mathcal{D})$ which preserves attribute types, in the precise sense that for each $A \in \mathcal{A}_{\mathcal{D}}$ and each $a \in \mathsf{Const}_{\mathcal{D}}(A)$, $h(a) \in \mathsf{Const}_{\mathcal{D}}(A)$. If h is additionally a bijection, then it is called an *automorphism* of $\mathcal{D}$. For $S \subseteq \mathsf{Const}(\mathcal{D})$, call h S-invariant if $h(a) = a$ for all $a \in S$. Given a database schema $\mathbf{D}$, an endomorphism on $\mathcal{D}$ induces a mapping from $\mathsf{GrAtoms}(\mathbf{D})$ to itself given by sending $t \in \mathsf{GrAtoms}(\mathbf{D})$ to the tuple t' with $t'[\mathsf{RName}] = t[\mathsf{RName}]$ and $t'[A] = t[h(A)]$ for all $A \in \mathsf{Ar}_{\mathbf{t}[\mathsf{RName}]}$. This mapping on atoms is also represented by h, as will the induced mapping from $\mathsf{DB}(\mathbf{D})$ to itself given by $M \mapsto \{h(t) \mid t \in M\}$. $h(M)$ is called an *endomorphic image* of M. Given $\Phi \subseteq \mathsf{WFS}(\mathbf{D})$, an $M \in \mathsf{DB}(\mathbf{D})$ is a *universal model* for Ψ if every $M \in \mathsf{AtMod}_I(\Psi)$ is a superset of an endomorphic image of M.

Say that $\mathbf{D}$ *admits universal models* if $M \cup \mathsf{Constr}(\mathbf{D})$ admits a universal model for every $M \in \mathsf{DB}(\mathbf{D})$ which extends to a model; *i.e.*, for which there exists an $M' \in \mathsf{LDB}(\mathbf{D})$ with $M \subseteq M'$.

In the context of database constraints, such universal models may be generated using the inference procedure described above [FKMP05, Sec. 3.1], provided the procedure terminates. In particular, the combination of weakly acyclic tgds and all egds noted above has this property. The following result thus provides a rich class of base schemata which imply semantic bijectivity for $\exists\wedge+$ when the underlying function is bijective.

4.18 Proposition — Universal models $\Rightarrow$ semantic bijectivity *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a database morphism of class $\exists\wedge+$ whose underlying function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$ is bijective. If $\mathbf{D}_1$ admits universal models, then f is semantically bijective for $\exists\wedge+$.*

PROOF OUTLINE: Space limitations preclude a detailed proof, but it is easy to sketch the main idea. Let $\varphi_1 \in \mathsf{WFS}(\mathbf{D}_1, \exists\wedge+)$. The basic strategy is to Skolemize φ_1 into a set G of ground atoms by replacing each existentially-quantified variable by a distinct new constant not appearing in $\mathsf{Constr}(\mathbf{D})$, and to generate a universal model $M \in \mathsf{LDB}(\mathbf{D}_1)$ for G. Next, map M to $f(M) \in \mathsf{LDB}(\mathbf{D}_2)$ and reverse the process. Represent $f(M)$ as a sentence φ_2' which is the conjunction of the atoms in $f(M)$, and then "un-Skolemize" φ_2' by replacing all constants which were not in the original φ_1 or in $\mathsf{Constr}(\mathbf{D})$ by existentially quantified variables. Call the resulting formula φ_2. It is not difficult to see that $\mathsf{Subst}\langle f, \varphi_2\rangle \equiv_{\mathbf{D}_1} \varphi_1$, from which the result follows. □

In the context of 4.16 above, note that for $\mathbf{E}_4$ the constraint $(\forall x)(T_4(x) \Leftrightarrow ((R_4(x) \wedge \neg S_4(x)) \vee (\neg S_1(x) \wedge R_4(x))))$ is not a dependency of the $(\forall)(\exists)$-Horn variety, and does not admit universal models. On the other hand, the alternative $(\forall x)(T_3(x) \Leftrightarrow R_3(x) \wedge S_3(x))$ for $\mathbf{E}_3$ is in fact representable as as set of three tgds: $(\forall x)(T_3(x) \Rightarrow R_3(x))$, $(\forall x)(T_3(x) \Rightarrow S_3(x))$, and $(\forall x)((R_3(x) \wedge S_3(x)) \Rightarrow T_3(x))$, and so does admit universal models by [FKMP05, Thm. 3.9]. Thus, the assertion of the above proposition is confirmed by this example.

4.19 Update difference and optimal reflections The update difference of an update (M_1, M_2) on $\mathbf{D}$ with respect to a set $\Sigma \subseteq \mathsf{WFS}(\mathbf{D})$ is a measure of how much M_1 and M_2 differ in terms of their information content relative to Σ. Formally, the *positive (Δ^+), negative (Δ^-), and total (Δ) update* *differences* of (M_1,M_2) with respect to Σ are defined as follows:

$$\begin{aligned}\Delta^+\langle(M_1,M_2),\Sigma\rangle &= \mathsf{Info}\langle M_2,\Sigma\rangle \setminus \mathsf{Info}\langle M_1,\Sigma\rangle\\ \Delta^-\langle(M_1,M_2),\Sigma\rangle &= \mathsf{Info}\langle M_1,\Sigma\rangle \setminus \mathsf{Info}\langle M_2,\Sigma\rangle\\ \Delta\langle(M_1,M_2),\Sigma\rangle &= \Delta^+\langle(M_1,M_2),\Sigma\rangle \cup \Delta^-\langle(M_1,M_2),\Sigma\rangle\end{aligned}$$

Given $\varphi \in \Delta\langle(M_1,M_2),\Sigma\rangle$, it is always possible to determine whether $\varphi \in \Delta^+\langle(M_1,M_2),\Sigma\rangle$ or $\varphi \in \Delta^-\langle(M_1,M_2),\Sigma\rangle$ by checking whether or not $M_1 \in \mathsf{AtMod}_I(\varphi)$.

For an update request (M_1,N_2) from Γ to $\mathbf{D}$, the quality of a realization (M_1,M_2) is measured by its update difference, with an optimal realization one which entails the least change of information. Formally, let $\Sigma \subseteq \mathsf{WFS}(\mathbf{D})$, let (M_1,N_2) be an update request along Γ, and let $(M_1,M_2) \in \mathsf{UpdRealiz}\langle M_1,N_2,\Gamma\rangle$. The pair (M_1,M_2) is *optimal* with respect to Σ if for all $(M_1,M_2') \in \mathsf{UpdRealiz}\langle M_1,N_2,\Gamma\rangle$, $\Delta\langle(M_1,M_2),\Sigma\rangle \subseteq \Delta\langle(M_1,M_2'),\Sigma\rangle$.

The definition of optimal which is used here is slightly different than that of [Heg08, 3.5], in which optimality also requires minimality of update difference with respect to $\mathsf{GrAtoms}(\mathbf{D})$. That additional condition was necessary because the main information measure was not required to be separating. Here, the separating condition effectively provides the additional minimality.

4.20 Lemma — Distance preservation under semantic bijection *Let $f : \mathbf{D}_1 \to \mathbf{D}_2$ be a semantic bijection of class $\mathcal{C}$, and $M_1,M_2,M_3 \in \mathsf{LDB}(\mathbf{D}_1)$. Then*

$$\begin{aligned}&\Delta\langle(M_1,M_2),\mathsf{WFF}(\mathbf{D}_1,\mathcal{C})\rangle \subseteq \Delta\langle(M_1,M_3),\mathsf{WFF}(\mathbf{D}_1,\mathcal{C})\rangle\\ &\qquad\textit{iff}\qquad \Delta\langle(f(M_1),f(M_2)),\mathsf{WFF}(\mathbf{D}_2,\mathcal{C})\rangle \subseteq \Delta\langle(f(M_1),f(M_3)),\mathsf{WFF}(\mathbf{D}_2,\mathcal{C})\rangle.\end{aligned}$$

PROOF: The proof follows directly from the bijective correspondence between (semantic equivalence classes of) sentences in $\mathsf{WFF}(\mathbf{D}_1,\mathcal{C})$ and those in $\mathsf{WFF}(\mathbf{D}_2,\mathcal{C})$, as established in 4.13. □

4.21 Lemma *If the views $\Gamma_1 = (\mathbf{V}_1,\gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2,\gamma_2)$ are of class $\mathcal{C}$, then so too is $\Gamma_1 \coprod \Gamma_2$.*

PROOF: This follows from the definition (4.8(b)), since the set of interpretation functions for $\mathbf{V}_1 \coprod \mathbf{V}_2$ into $\mathbf{D}$ is simply the (disjoint) union of those for γ_1 and for γ_2. □

The definition of complementary views 3.5 is extended to the $\mathcal{C}$ context in the following fashion.

4.22 $\mathcal{C}$-complementary pairs The set $\{\Gamma_1,\Gamma_2\}$ of views is said to be a *$\mathcal{C}$-complementary pair* if each view is of class $\mathcal{C}$ and, in addition, the morphism $\gamma_1 \coprod \gamma_2 : \mathbf{D} \to \mathbf{V}_1 \coprod \mathbf{V}_2$ is semantically bijective for $\mathcal{C}$.

Finally, it is possible to establish the main result of this paper.

4.23 Theorem *Let $\{\Gamma_1 = (\mathbf{V}_1,\gamma_1),\Gamma_2 = (\mathbf{V}_2,\gamma_2)\}$ be a $\mathcal{C}$-complementary pair and let (M_1,N_1) be an update request from Γ_1 to $\mathbf{D}$. If the Γ_2-constant realization of (M_1,N_1) along Γ_1 exists, it is optimal with respect to* $\mathsf{WFS}(\mathbf{D},\mathcal{C})$.

PROOF: Let $M_2 \in \mathsf{LDB}(\mathbf{D})$ be the unique database for which (M_1,M_2) is the Γ_2 constant realization of (M_1,N_1); thus, $(\gamma_1 \coprod \gamma_2)(M_2) = (N_1,\gamma_2(M_1))$. Let $M_3 \in \mathsf{LDB}(\mathbf{D})$ be any database for which $\gamma_1(M_3) = N_1'$; thus, (M_1,M_3) is an arbitrary realization of the update request (M_1,N_1'). The update in $\mathbf{V}_1 \coprod \mathbf{V}_2$ corresponding to (M_1,M_2) under $\gamma_1 \coprod \gamma_2$ is $u = ((\gamma_1(M_1) \uplus \gamma_2(M_1)),(N_1 \uplus \gamma_2(M_1))),$

while for (M_1, M_3) it is $u' = ((\gamma_1(M_1) \uplus \gamma_2(M_1)), (N_1 \uplus \gamma_2(M_3)))$. Clearly $\Delta\langle u, \mathsf{WFS}(\mathbf{V}_1 \coprod \mathbf{V}_2, \mathcal{C})\rangle \subseteq \Delta\langle u', \mathsf{WFS}(\mathbf{V}_1 \coprod \mathbf{V}_2, \mathcal{C})\rangle$, and so in view of 4.20 and 4.21, $\Delta\langle (M_1, M_2), \mathsf{WFF}(\mathbf{D}, \mathcal{C})\rangle \subseteq \Delta\langle (M_1, M_3), \mathsf{WFF}(\mathbf{D}, \mathcal{C})\rangle$ as well. Hence (M_1, M_2) is optimal, as required. □

4.24 Corollary — Global uniqueness of constant-complement updates *Let* $\{\Gamma_1, \Gamma_2\}$ *and* $\{\Gamma_1, \Gamma'_2\}$ *be* $\mathcal{C}$*-complementary pairs, and let* (M_1, N_2) *be an update request from* Γ_1 *to* $\mathbf{D}$*. If* (M_1, N_2) *has both a* Γ_2*-constant realization and a* Γ'_2*-constant realization, then these two realizations are identical. In other words, a constant-complement realization of an update is independent of the choice of complement, as long as the complementary pair is* $\mathcal{C}$*-compatible.*

PROOF: By construction, an optimal reflection is unique, and so the result follows from 4.23. □

4.25 Corollary — Universal solutions imply global uniqueness for $\mathcal{C} = \exists\wedge+$ *Assume that* $\mathbf{D}$ *admits universal solutions, let* Γ_1 *be any view of* $\mathbf{D}$ *which is of class* $\exists\wedge+$*, and let* (M_1, N_2) *be an update request from* Γ_1 *to* $\mathbf{D}$*. Then for all views* Γ_2 *of class* $\exists\wedge+$ *which are complements of* Γ_1 *and for which the* Γ_2*-constant translation of* (M_1, N_1) *exists, these translations are identical.*

PROOF: The proof follows directly from 4.18 and 4.24. □

4.26 Examples To begin, consider an example which was introduced in [Heg94, 1.1.1] as motivation for the need to consider restrictions on the nature of "good" complements. It recaptures ideas similar to those found in $\mathbf{E}_4$ of 4.16, but in the context of three views, each of which contains one unary relation symbol. Let $\mathbf{E}_7$ have two relation symbols $R_{71}[A]$ and $R_{72}[A]$, with no constraints other than those imposed by the relational context $\mathcal{D}$. Let $\Omega_{7i} = (\mathbf{W}_{7i}, \omega_{7i})$ for $i \in \{1,2\}$ be the view which retains R_{7i} but discards $R_{7(3-i)}$. $R_{7i}[A]$ is thus the sole relation symbol for $\mathbf{W}_{7i}$. In each case, the interpretation formula $\omega_{7i}^{R_{7i}}$ is the identity on R_{7i}, and so the coproduct morphism $\omega_{71} \coprod \omega_{72}$ is also an identity and trivially semantically bijective for any reasonable choice for $\mathcal{C}$. The pair $\{\Omega_{71}, \Omega_{72}\}$ is as well behaved as can be, and $\mathcal{C}$-complementary. Now consider a third view $\Omega_{73} = (\mathbf{W}_{73}, \omega_{73})$ which has the single relation symbol $R_{73}[A]$, defined by the formula $\omega_{73}^{R_{73}} = (R_{71}(x_A) \wedge \neg R_{72}(x_A) \vee (\neg R_{71}(x_A) \wedge R_{72}(x_A))$. In other words, the value for R_3 is the symmetric difference of those for R_{71} and R_{72}. It is easy to see that any set of two of these three views forms a complementary pair, but the two pairs which contain Ω_{73} are not $\mathcal{C}$-complementary for $\mathcal{C} \in \{\exists\neq, \exists+, \exists\wedge+, \mathsf{Atoms}\}$. The interpretation $\omega_{73}^{R_3}$ involves negation and so Ω_{73} can never be of class $\mathcal{C}$ for any $\mathcal{C}$ which renders $\mathsf{WFS}(\mathbf{E}_7, \mathcal{C})$ information monotone. Thus, this "undesirable" complement is excluded by the theory. In this example, the schema $\mathbf{E}_7$ admits universal solutions, but the morphism $\omega_{7i} \coprod \omega_{23}$ is not of class $\mathcal{C}$ for $i \in \{1,2\}$, and so 4.25 is not applicable.

As a slight variation, reconsider the schema $\mathbf{E}_4$ of 4.16, this time with the three views $\Omega_{4i} = (\mathbf{W}_{4i}, \omega_{4i})$, with Ω_{4i} for $i \in \{1,2,3\}$ the view which retains R_{4i} but discards the other two relations. Each view morphism ω_{4i} is very well behaved; each is semantically surjective for any choice of $\mathcal{C}$ listed in 2.4. Furthermore, it is easy to see that any two of these views forms a complementary pair. However, by an argument virtually identical to that already given in 4.16, for any pair of these views, the reconstruction morphism cannot be of class $\mathcal{C}$, since including R_{43} in the view forces a symmetric difference interpretation. In this case, the each view morphism of the form $\omega_{4i} \coprod \omega_{4j}$ is of class $\mathcal{C}$, but the constraints of the main schema $\mathbf{E}_4$ do not allow the reconstruction to be of class $\mathcal{C}$ for any reasonable choice.

4.27 Example — Comparison to the order-view approach Upon comparing 4.24 of this paper to [Heg04, 4.3], two key differences are apparent. On the one hand, the theory of [Heg04] requires *order complements*, which is based upon the order structure on the states of the schemata and has nothing whatever to do with logic or the relational model, while the theory presented here requires $\mathcal{C}$-complements, which are logic based. Although a detailed study has not been made, partly because the theory of [Heg04] is limited to so-called meet complements, it does appear the notion of order complement is strictly more general than that of a $\mathcal{C}$-complement. On the other hand, the uniqueness theory of [Heg04] is limited to *order-realizable updates*; that is, updates which can be realized as sequences of legal insertions and deletions. The theory of this paper imposes no such limitation.

To illustrate this advantage, consider the schema $\mathbf{E}_8$ which has the single relation symbol $R[ABC]$, constrained by the functional dependencies $B \to C$ and $B \to A$. The two views are $\Pi^{\mathbf{E}_8}_{AB}$ and $\Pi^{\mathbf{E}_8}_{BC}$, the projections onto AB and BC respectively. Both of these are *order views*; the associated mappings on database states are open poset morphisms. Thus, the result [Heg04, 4.3] applies, but only to order-realizable updates, of which there are none for $\Pi^{\mathbf{E}_8}_{AB}$. More precisely, the only possible updates on $\Pi^{\mathbf{E}_8}_{AB}$ are those which change the A-value of a given tuple, and none of those is an order-realizable update, so that theory does not address this situation in a systematic way [Heg04, 4.6]. On the other hand, the theory of this paper imposes no such restriction to order-realizable updates. Since the coproduct $\Pi^{\mathbf{E}_8}_{AB} \coprod \Pi^{\mathbf{E}_8}_{BC}$ is easily seen to be semantically bijective for any reasonable choice for $\mathcal{C}$ (the reconstruction map is the join), all $\Pi^{\mathbf{E}_8}_{BC}$-constant updates on $\Pi^{\mathbf{E}_8}_{AB}$ are allowed.

5. Conclusions and Further Directions

The constant-complement update strategy for views has been examined from the point of view of information content. Specifically, in this approach, the decomposition mapping for the pair of views is required not only to be bijective on states but also on the sentences which define the information content of the component views. From this perspective, it has been shown that under very broad conditions, the translation of the view update is independent of the choice of complement. In particular, in a traditional database context — well-behaved dependencies and view mappings defined by conjunctive queries — the translation never depends upon the choice of complement.

Further investigations are appropriate for the following topics.

Extension to other logical models The theory presented here is couched squarely within the classical relational model. In principle, it applies as well to other models which admit a first-order logical formalism, such as the nested relational model [PDGV89, Ch. 7] and the Higher-Order Entity-Relationship Model (HERM) [Tha00]. However, the extent to which the cornerstone ideas such as identifying a suitable class $\mathcal{C}$ or characterizing semantic bijectivity via universal models translate to realistic data modelling within those frameworks requires further investigation.

Rapprochement with the order-based approach The order-based framework for the constant-complement approach, as reported in [Heg04], has the great advantage that it is not tied to a particular data model. Rather, schemata are modelled as ordered sets and view mappings as poset morphisms. While the framework presented here is much more specific, being limited to the relational model, it also supports a broader result which is not limited to order-realizable updates. Each approach has its advantages and disadvantages. A rapprochement of the two appears to be a fruitful topic for future investigation. In particular, it would be interesting to identify the extent to which the ideas of this paper can be recast without the specific need for an underlying logic. In this regard, element-based variations of the order-based model, such as that employed in [Heg06], would perhaps form the foundation for a fruitful common ground.

Information morphisms and the inversion of schema mappings Recently, there have been significant advances in the theory of data exchange and inversion of schema mappings [FKMP05] [Fag07]. Although these topics have nothing to do with constant-complement updates, the ideas of information content, and in particular the idea of characterizing database morphisms not by how they map models but rather how they map sentences might prove useful in understanding and characterizing such operations.

Acknowledgment The core ideas for this research were developed while the author was a visitor at the Information Systems Engineering Group at the University of Kiel. He is indebted to Bernhard Thalheim and the members of the group for the invitation and for the many discussions which led to this work. Also, the anonymous reviewers made numerous suggestions which (hopefully) have led to a more readable presentation.

References

[AHV95] S. Abiteboul, R. Hull, and V. Vianu, *Foundations of Databases*, Addison-Wesley, 1995.

[BS81] F. Bancilhon and N. Spyratos, "Update semantics of relational views," *ACM Trans. Database Systems*, **6**(1981), pp. 557–575.

[BV84] C. Beeri and M. Y. Vardi, "A proof procedure for data dependencies," *J. Assoc. Comp. Mach.*, **31**(1984), pp. 718–741.

[CGT90] S. Ceri, G. Gottlob, and L. Tanca, *Logic Programming and Databases*, Springer-Verlag, 1990.

[DG84] W. F. Dowling and J. H. Gallier, "Linear-time algorithms for testing the satisfiability of propositional Horn clauses," *J. Logic Programming*, **3**(1984), pp. 267–284.

[Fag82] R. Fagin, "Horn clauses and database dependencies," *J. Assoc. Comp. Mach.*, **29**(1982), pp. 952–985.

[Fag07] R. Fagin, "Inverting schema mappings," *ACM Trans. Database Systems*, **32**(2007).

[FKMP05] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa, "Data exchange: Semantics and query answering," *Theoret. Comput. Sci.*, **336**(2005), pp. 89–124.

[FGM*07] J. N. Foster, M. B. Greenwald, J. T. Moore, B. C. Pierce, and A. Schmitt, "Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem," *ACM Trans. Programming Languages and Systems*, **29**(2007).

[GN87] M. R. Genesereth and N. J. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan-Kaufmann, 1987.

[Heg94] S. J. Hegner, "Unique complements and decompositions of database schemata," *J. Comput. System Sci.*, **48**(1994), pp. 9–57.

[Heg04] S. J. Hegner, "An order-based theory of updates for database views," *Ann. Math. Art. Intell.*, **40**(2004), pp. 63–125.

[Heg06] S. J. Hegner, "The complexity of embedded axiomatization for a class of closed database views," *Ann. Math. Art. Intell.*, **46**(2006), pp. 38–97.

[Heg08] S. J. Hegner, "Information-optimal reflections of view updates on relational database schemata," in: S. Hartmann and G. Kern-Isberner, eds., *Foundations of Information and Knowledge Systems: Fifth International Symposium, FoIKS 2008, Pisa, Italy, February 11-15, 2008, Proceedings*, pp. 112–131, Volume 4932 of Lecture Notes in Computer Science, Springer-Verlag, 2008.

[HS73] H. Herrlich and G. E. Strecker, *Category Theory*, Allyn and Bacon, 1973.

[JAK82] B. E. Jacobs, A. R. Aronson, and A. C. Klug, "On interpretations of relational languages and solutions to the implied constraint problem," *ACM Trans. Database Systems*, **7**(1982), pp. 291–315.

[Lec03] J. Lechtenbörger, "The impact of the constant complement approach towards view updating," in: *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, San Diego, California, June 09-11, 2003*, pp. 49–55, 2003.

[Mon76] J. D. Monk, *Mathematical Logic*, Springer-Verlag, 1976.

[PDGV89] J. Paredaens, P. De Bra, M. Gyssens, and D. Van Gucht, *The Structure of the Relational Database Model*, Springer-Verlag, 1989.

[Tha00] B. Thalheim, *Entity-Relationship Modeling*, Springer-Verlag, 2000.