Invariance Properties of the Constant-Complement View-Update Strategy

Stephen J. Hegner

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

Abstract. The constant-complement approach is one of the principal strategies for defining the reflections of database view updates. The purpose of this paper is twofold. First, a self-contained presentation of the strategy itself is given. Second, two fundamental invariance problems are examined. *Admissibility invariance* addresses the question of when the translatability of a view update is independent of the view state. *Reflection invariance* addresses the question of when a view update is independent of the choice of complement. In addition to a summary of existing results for both forms of invariance, new results for reflection invariance are presented.

1 Introduction

To illustrate the main ideas of this paper, it is convenient to begin with a very small database schema $\mathbf{E}_0$ which involves only four data items, a, b, c, and d. A *database* of $\mathbf{E}_0$ is any subset of $\{a, b, c, d\}$; thus, $\mathbf{E}_0$ has exactly sixteen possible databases. For simplicity, a nonempty subset of $\{a, b, c, d\}$ is written as the concatenation of its elements, in lexicographic order. For example, acd is shorthand for $\{a, c, d\}$. To avoid confusion, the symbol $\emptyset$, rather than the empty string, is used to denote the empty subset of $\{a, b, c, d\}$.

Database schemata typically include constraints which limit the databases which are allowed. For this simple example, the only constraint is that b and d must occur together. More formally, $\mathsf{LDB}(\mathbf{E}_0) = \{\emptyset, a, c, ac, bd, bcd, abd, abcd\}$ denotes the set of *legal databases* of $\mathbf{E}_0$; that is, those databases which satisfy the constraint that b and d must occur together or not at all. A *state* of a database schema is a legal database.

Virtually all modern database-management systems provide *views*, or windows on the database. Such windows provide partial, but generally not total, information about the state of the main schema. For the example schema $\mathbf{E}_0$, the view $\Omega_0 = (\mathbf{W}_0, \omega_0)$, which retains only information about the presence of a and b in the current state, is illustrated in Fig. 1 below. The view schema $\mathbf{W}_0$ is defined by $\mathsf{LDB}(\mathbf{W}_0) = \{\emptyset, a, b, ab\}$ and the view mapping ω_0 by $M \mapsto M \cap ab$.

A view mapping is always total and surjective, by definition. Each state of the view schema must be the image of some state of the main schema. Although every

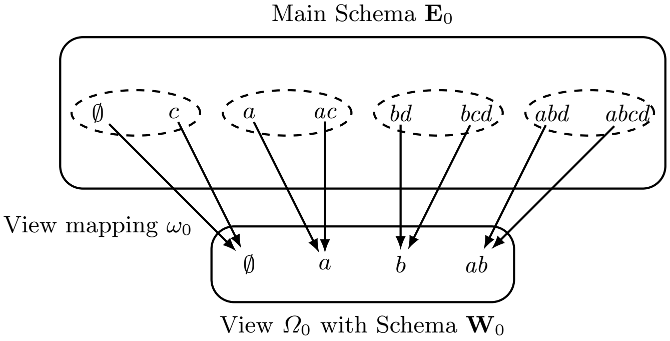

Fig. 1. Visualization of a view

update to the state of a view schema has at least one *reflection* to an update on the main schema (i.e., an update to the main schema which is consistent with the view update), there is rarely only one such possibility. An update to any schema, main or view, is represented as a pair (M_1, M_2), with M_1 the current state and M_2 the new state after the change. In the simple example of Fig. 1, since each view state is the image of exactly two states of the main schema, there are two possible reflections for each view update. For example, if the current state of the main schema is c, the view update $(\emptyset, a)$ has two possible reflections for the corresponding update to the main schema, (c, a) and (c, ac). In other words, both c and ac are candidates for the new state of $\mathbf{E}_0$ in support of the view update $(\emptyset, a)$. The problem of determining which, if any, of the possible reflections is suitable is called the *view-update problem*, and a specific approach to solving this problem is termed a *view-update strategy*.

This paper is about one such approach, the *constant-complement* strategy. Before examining it in more detail on the above example, to place the general approach in context, it is useful to provide a brief discussion of view-update strategies, which may be classified along several lines. Perhaps the most fundamental distinction is between design-based and principle-based strategies. Roughly speaking, in a *design-based* strategy, the reflection mechanism for view updates is defined explicitly as part of the specification of the view itself. With such an approach, the range of view updates which may be supported is very large, but, on the other hand, it is up to the designer to account for whatever effects the translation of a view update will have upon the properties of the main schema. Furthermore, with such strategies, there may be many views which provide precisely the same information, but which support updates in very different ways. Most designed-based strategies, by their very nature, are customized to a specific application, although there has been some recent work which attempts to systematize this approach using a construction known as a lens [8, 13].

In a *principle-based* strategy, view-update reflections are chosen according to certain principles. Thus, in particular, the update-reflection mechanism is determined by the update principle, and not explicitly as an added component of the view specification. Much of the theory of view-update support is based upon principles which are founded in specific representations for specific data models. By far, the most common context is the representation of views in the relational model using the relational algebra or calculus. Examples include [11, 26, 6, 7]. While they often provide interesting insights, they do not, in general, provide a unified theory of how view updates may be handled.

There are at least two major principle-based strategies which are general in character, one based upon minimal repairs and the other upon constant complement. They are nevertheless diametrically opposed in nature. The minimal-repair strategy provides a ranking on possible reflections, but does not otherwise classify them as acceptable or unacceptable. On the other hand, the constant-complement strategy provides an absolute classification of acceptability, without any other ranking. It is instructive to take a slightly closer look.

In the approach based upon *minimal repairs*, the underlying principle is that the reflection of a view update should change the state of the main schema as little as possible. To formalize this, a distance metric between states of the main schema is identified. An *optimal reflection* of a view update, if it exists, is defined by a reflection for which the distance between the old state (before the update) of the main schema and the new state (after the update) is least. For example, return to the context of the view Ω_0 of $\mathbf{W}_0$, with distance measured by the number of data objects whose presence in the state changes. Then for the view update $(\emptyset, a)$ with current state c of the main schema, the reflection (c, ac) to the main schema is preferred to (c, a), since in the former only a is added (one change), while in (c, a) a is added while c is removed (two changes). If no such optimal update exists, it is still possible to identify a set of *minimal reflections*. Most of the efforts based upon minimal repairs are formulated within the context of logic databases; for a comprehensive presentation, see [3, 2]. In [21], the basic ideas are applied to a class of traditional relational views.

Although undoubtedly useful, such a ranking approach can also produce questionable results. Consider, for example, a view which computes the average salary of all employees. Using a metric of minimal number of tuples or minimal subset of tuples changed, to support a view update which increases the average salary by some value x, the minimal reflections are those which alter the salary of exactly one employee, and leave the other salaries unchanged, together with those which add one new employee with just the right salary to achieve the desired average. Of course, the metric can be questioned in this case, and this example shows that a main challenge facing a designer of a minimal-repair strategy is the choice of the distance metric. Traditional metrics, such as counting the number of tuples which change, or simply using symmetric set difference, have mathematical appeal but are sometimes too coarse to recapture what would otherwise appear to be the best reflection. Thus, a thorough study of this approach must necessarily focus upon the choice of metric as well.

The *constant-complement* strategy, first described in [5], provides absolute criteria for the acceptability of a view-update reflection. It focuses upon *encap-*

sulation — only that part of the main schema which embodies the information contained in the view may be altered; the rest of the main schema must remain unchanged. Encapsulation may be cast equivalently via the principle that the updates which are allowed on the view must be *closed* — all changes to the main schema which are induced by a view update should be visible within the view itself; there should be no *side effects* which are not fully represented within the view [17, Sec. 1].

Such encapsulation is achieved by identifying a second view which is *complementary* to the one which is to be updated. This complementary view recaptures that part of the main schema which is not covered by the view to be updated. By keeping the complement constant, all changes are limited to the updated view.

The main idea is illustrated in Fig. 2 for the view Ω_0 of Fig. 1. The complement Ω'_0 retains information about the presence of c and d in the state of the main schema; i.e., the view mapping ω'_0 is defined by $M \mapsto M \cap cd$. The horizontal ellipses, shaded in gray, show the groups of states of the main schema $\mathbf{E}_0$ which map to the same state of the view Ω_0. The vertical ellipses, with dotted borders, show the groups of states of the main schema $\mathbf{E}_0$ which map to the same state of the view Ω'_0. Define the *decomposition mapping* $\omega_0 \times \omega'_0 : \mathsf{LDB}(\mathbf{E}_0) \to \mathsf{LDB}(\mathbf{W}_0) \times \mathsf{LDB}(\mathbf{W}'_0)$ on elements by $M \mapsto (\omega_0(M), \omega'_0(M))$. It is easy to see that this mapping is injective; indeed, for $M \in \mathsf{LDB}(\mathbf{E}_0)$, $M \mapsto (M \cap ab, M \cap cd)$. This implies that the state of the main schema $\mathbf{E}_0$ may be recovered from the combined states of both views. In classical database theory, it is said that $\mathbf{E}_0$ decomposes *losslessly* into $\mathbf{W}_0$ and $\mathbf{W}'_0$ (via the views Ω_0 and Ω'_0). This is the definition of a complementary pair — that the associated decomposition be lossless.

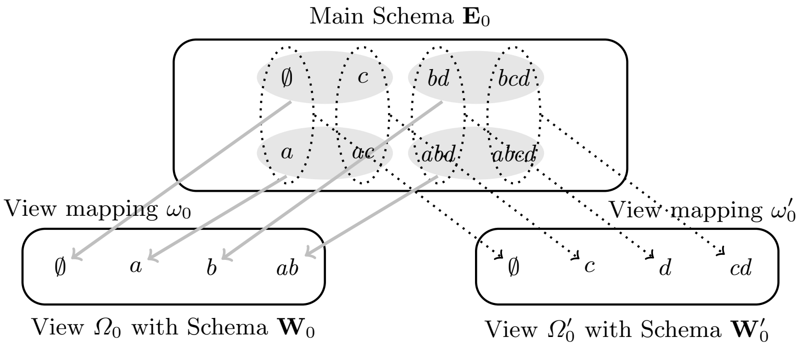

Fig. 2. Visualization of a complementary pair

Consider again the potential update $(\emptyset, a)$ to $\mathbf{W}_0$, and assume further that the the current state of $\mathbf{E}_0$ is c. As noted above, there are two possible reflections to the main schema, (c, a) and (c, ac), but only the second keeps the state of $\mathbf{W}'_0$

the same. It is the only one which avoids a change to (a *side effect* on) Ω'_0, and thus is the only one admitted by the constant-complement strategy. Indeed, since $\omega_0 \times \omega'_0$ is injective, there can be at most one reflection update which keeps Ω'_0 constant. The new state of the main schema $\mathbf{E}_0$ must be $(\omega_0 \times \omega'_0)^{-1}(a, \omega'_0(c)) = (\omega_0 \times \omega'_0)^{-1}(a, c) = ac$.

Not all updates to Ω_0 are possible with constant complement Ω'_0. To see why this is so, observe that the two views Ω_0 and Ω'_0 overlap, due to the constraint that b and d must occur together or not at all. For example, if the current state of $\mathbf{W}_0$ is $\emptyset$ or a, then d cannot be in the state of $\mathbf{W}'_0$. Similarly, if the current state of $\mathbf{W}_0$ is either b or ab, then d must be in the state of $\mathbf{W}'_0$. Thus, none of the updates $(\emptyset, b)$, $(\emptyset, bd)$, (a, b), and (a, ab) is possible on Ω_0 while keeping Ω'_0 constant. This is a fundamental property of the constant-complement strategy — in general, not all view updates admit a reflection. This is the "price" of requiring encapsulation.

It is instructive to illustrate these same ideas with a simple relational example. Let the schema $\mathbf{E}_1$ consist of the single relation symbol $R[ABC]$, constrained by the join dependency $\bowtie[AB, BC]$. The view to be updated is $\Pi^{\mathbf{E}_1}_{AB} = (\mathbf{E}^{AB}_1, \pi^{\mathbf{E}_1}_{AB})$, defined by the projection of $R[ABC]$ onto AB. Here $\mathbf{E}^{AB}_1$ is the schema whose single relation symbol is $R_{AB}[AB]$, while $\pi^{\mathbf{E}_1}_{AB}$ is the projection morphism. The chosen complementary view is $\Pi^{\mathbf{E}_1}_{BC} = (\mathbf{E}^{BC}_1, \pi^{\mathbf{E}_1}_{BC})$, defined by the projection onto BC. That $\{\Pi^{\mathbf{E}_1}_{AB}, \Pi^{\mathbf{E}_1}_{BC}\}$ forms a complementary pair follows from the classical result [31, Thm. 1]. With constant-complement $\Pi^{\mathbf{E}_1}_{BC}$, the updates which are allowed to $\Pi^{\mathbf{E}_1}_{AB}$ are those which keep the the projection onto B constant. This illustrates the associated encapsulation. Any change to $\Pi^{\mathbf{E}_1}_{B}$ would necessarily imply a change to $\Pi^{\mathbf{E}_1}_{BC}$, which would include a change not visible from within $\Pi^{\mathbf{E}_1}_{AB}$. Thus, in this example, $\Pi^{\mathbf{E}_1}_{B}$ must be held constant to render $\Pi^{\mathbf{E}_1}_{AB}$ a closed view, without side effects to its allowed updates. Figure 3 provides a visualization of this situation.

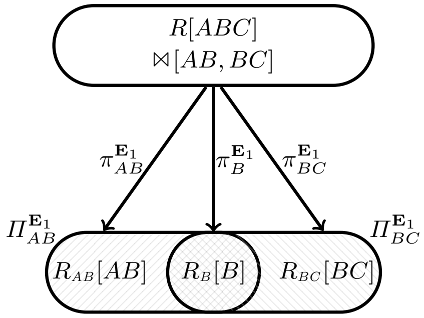
$R[ABC]$
$\bowtie[AB, BC]$
$\pi^{\mathbf{E}_1}_{AB}$
$\pi^{\mathbf{E}_1}_{B}$
$\pi^{\mathbf{E}_1}_{BC}$
$\Pi^{\mathbf{E}_1}_{AB}$
$\Pi^{\mathbf{E}_1}_{BC}$
$R_{AB}[AB]$
$R_B[B]$
$R_{BC}[BC]$

Fig. 3. Visualization of constant-complement in the relational setting

The price of encapsulation has been seen by some to be too high. For example, according to [27, p. 41], the constant complement strategy is "only applicable to a trivial class of views", and is "too restrictive to be of any practical use". This criticism must be evaluated in context. The starting point in [27] is that all updates on a view which preserve its integrity constraints should be supported. However, the idea that only some, but not all, updates to a view are supported by a given strategy is not at all unusual, and is central in many approaches. Furthermore, for a summary view such as the example given above which computes average salary, an update through anything but a design-based strategy seems unreasonable. Even the SQL standard limits updates to views [9, Sec. 13.3]. In any case, the constant-complement strategy is exactly what is required to ensure encapsulation. Of course, it is difficult to argue that all view update strategies require encapsulation, but it is equally difficult to argue that none do.

The constant-complement strategy is important for at least one additional reason — it is central to the approach of modelling database schemata via *components* [33, 32]. The ways in which such components are interconnected, as well as the ways in which updates are supported on such interconnections, is very closely related to the constant-complement strategy [19, 25, 22].

In short, while it is not the universal solution for all view-update problems, the constant-complement strategy is sufficiently useful that it should be understood by anyone interested in the principles of database updates via views.

The goals of this paper are twofold. The first is to present the fundamental ideas of the constant-complement strategy in a reasonably self-contained fashion. The second is to examine two invariance issues surrounding the approach.

The first is called *admissibility invariance*. For an update strategy to be truly encapsulated, the admissibility of a view update must not depend upon the state of the main schema. A given view update (N, N') must either be allowed for all states M of the main schema which map to N, or else for none of them. It is easy to verify that the examples depicted in Fig. 2 and Fig. 3 each have this property. However, it is possible to alter the example of Fig. 2 so that this property fails, as illustrated in Fig. 4. This main schema $\mathbf{E}_2$ is identical to $\mathbf{E}_0$, save that ac has been removed. The views Ω_2 and Ω'_2 are likewise identical to Ω_0 and Ω'_0, respectively, save that ac is no longer in the domain of the view mapping. This example no longer exhibits admissibility invariance, since the view update $(\emptyset, a)$ to $\mathbf{W}_0$ is supported with constant complement Ω'_2 if the current state of the main schema $\mathbf{E}_2$ is $\emptyset$, but not if it is c.

A similar effect can be observed in the relational example of Fig. 3. Obtain $\mathbf{E}'_1$ from $\mathbf{E}_1$ by adding the functional dependency (FD) $A \rightarrow C$, with the views $\Pi^{\mathbf{E}'_1}_{AB}$ and $\Pi^{\mathbf{E}'_1}_{BC}$ defined as for $\mathbf{E}_1$. Consider the view update $u = \{(\{R_{AB}(\mathrm{a}_1, \mathrm{b}_1), R_{AB}(\mathrm{a}_2, \mathrm{b}_2)\}, \{R_{AB}(\mathrm{a}_1, \mathrm{b}_1), R_{AB}(\mathrm{a}_1, \mathrm{b}_2)\})\}$ to $\Pi^{\mathbf{E}'_1}_{AB}$. This update is possible with constant complement $\Pi^{\mathbf{E}'_1}_{BC}$ if the state of the main schema is $M_{1a} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_1)\}$, (to new state $M_{1a'} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R(\mathrm{a}_1, \mathrm{b}_2, \mathrm{c}_1)\}$), but not if it is $M_{1b} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R(\mathrm{a}_2, \mathrm{b}_2, \mathrm{c}_2)\}$, since in the latter case the required update to $M_{1b'} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R(\mathrm{a}_1, \mathrm{b}_2, \mathrm{c}_2)\}$ would violate the FD $A \rightarrow C$.

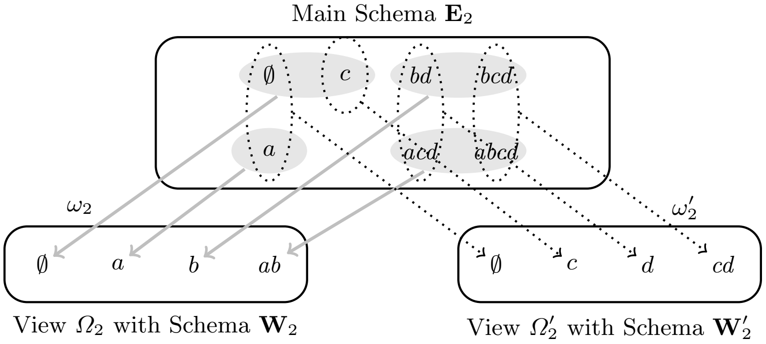
Main Schema $\mathbf{E}_2$
$\emptyset$
c
bd
bcd
a
acd
$abcd$
ω_2
ω_2'
$\emptyset$
a
b
ab
$\emptyset$
c
d
cd
View Ω_2 with Schema $\mathbf{W}_2$
View Ω_2' with Schema $\mathbf{W}_2'$

Fig. 4. Visualization of a situation without admissibility invariance

In Sec. 3, this question of admissibility invariance is examined in detail. As it turns out, there is a very general characterization of such invariance, defined in terms of the *congruences* of the views, which depend only upon a model of database schema given by sets and functions. Section 3 is a tutorial summary of known results; it does not contain anything fundamentally new, although the presentation is hopefully more accessible than that found in research papers on the subject.

The second invariance issue is called *reflection invariance*. One of the thornier issues with the constant-complement strategy, identified already in [5, Thm. 4.4], is that a view complement is almost never unique. Furthermore the reflection of a given update using the constant-complement strategy may depend upon the choice of complement, as illustrated in Fig. 5. This example uses the same main

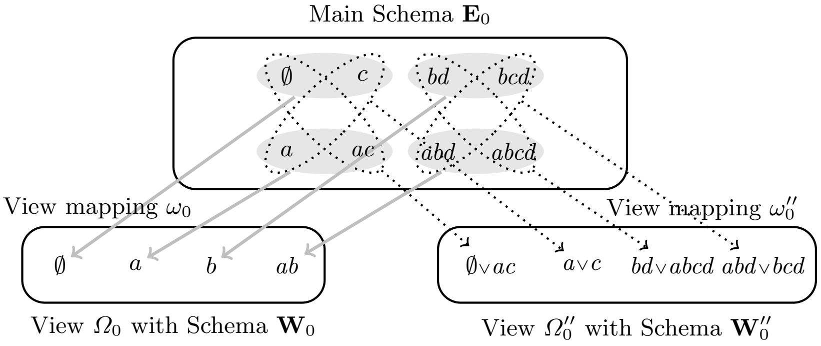

Fig. 5. Visualization of an alternate complement to Ω_0

schema $\mathbf{E}_0$ and the same view Ω_0 as in Fig. 2, but the complementary view $\Omega''_0 = (\mathbf{W}'_0, \omega''_0)$ is not the same as Ω'_0. The states of $\mathbf{W}''_0$ are represented by the disjunction of the states of the main schema which map to them. The updates allowed to Ω_0 under constant complement Ω''_0 are the same as those for constant complement Ω'_0, but with the exception of identity updates, the translations are always different. For example, from initial state c of the main schema, the view update $(\emptyset, a)$ has the unique reflection with constant complement Ω''_0 to the update (c, a) on the main schema $\mathbf{E}_0$, while with constant complement Ω'_0 the reflection is (c, ac). Without adding additional structure, there is no reasonable way to prefer one of these complements to the other.

This observation would appear to compromise the utility of the constant-complement strategy substantially. If the strategy depends upon the choice of complement, and complements are almost never unique, then encapsulation is reduced to a mathematical construction, with the choice of complement an arbitrary parameter. However, when one attempts to carry out a similar construction on a "real" example such as that of Fig. 3, as illustrated in [17, Sec. 1,3], it becomes remarkably difficult to construct a "reasonable" complement to $\Pi^{\mathbf{E}_1}_{AB}$ which does not at least include $\Pi^{\mathbf{E}_1}_{BC}$ as a subview. The key lies in the fact that most common views, particularly relational views, involve order. The databases have a natural order structure, and the view mappings respect this structure. When limiting the constant-complement strategy to that context in an appropriate manner, the reflection of a view update becomes independent of the choice of complement. The main results along these lines were established in [17]. In Sec. 4, a fundamental result along these lines is recalled, and then some important new extensions are developed.

In addition to these main sections, Sec. 2 provides the basic formalisms necessary to understand the constant-complement strategy and the extensions discussed in this paper, while Sec. 5 provides conclusions and further directions.

2 The Basic Framework of Views and Updates

The purpose of this section is twofold. First, it provides a compact summary of the necessary mathematical and notational material. Second, it provides a simple yet formal presentation of the basic constant-complement strategy.

Much of this framework appears, with minor variations, in earlier work of the author. The reader is referred in particular to [17, 20] for further details.

Familiarity with the relational model, as presented in [28, 30, 1] is assumed. For the most part, relational notation and terminology will not be reviewed here.

Notation 2.1 (Some mathematical shorthand). It will often be necessary to assert that a partial function f is defined on an argument x. The shorthand $f(x)\!\downarrow$ will be used in this regard.

Summary 2.2 (Equivalence relations, partitions, and blocks). Equivalence relations and their properties play a central rôle in the characterization of views, especially for properties associated with the constant-complement strategy. Although familiarity with the basic ideas is assumed, it is nevertheless useful to gather important terminology and notation in one place. A comprehensive reference on the subject of equivalence relations and their algebraic properties is [29], while [10] is a useful reference for order structures in general.

An *equivalence relation* r on a set S is one which is *reflexive* (for all $x \in S$, $(x,x) \in r$); *symmetric* (for all $x, y \in S$, $(x,y) \in r$ implies $(y,x) \in r$); and *transitive* (for all $x, y, z \in S$, $(x,y) \in r$ and $(y,z) \in r$ implies $(x,z) \in r$). The set of all equivalence relations on S is denoted $\mathsf{EqRels}(S)$.

The equivalence relation r divides S into disjoint blocks, with the *block* $\mathsf{Block}_r(x)$ containing x given by $\{y \in S \mid (x,y) \in r\}$. The set $\{\mathsf{Block}_r(x) \mid x \in S\}$ of all such blocks is called the *partition* of r and is denoted $\mathsf{Partition}(r)$.

There is a natural order $\preceq_S$ on $\mathsf{EqRels}(S)$ given by $r_1 \preceq_S r_2$ (written $r_2 \supset r_1$ in [29]) iff $r_2 \subseteq r_1$. Equivalently, $r_1 \preceq_S r_2$ iff every block of of r_2 is a subset of some block of r_1. In particular, the identity relation $\mathbf{1}_S$ is greatest in this ordering, and the trivial relation $\mathbf{0}_S = S \times S$ is least. If $r_1 \preceq_S r_2$, then $\mathsf{Partition}(r_1)$ is said to be *coarser* than $\mathsf{Partition}(r_2)$, and $\mathsf{Partition}(r_2)$ is *finer* than $\mathsf{Partition}(r_1)$.

This natural order induces a bounded complete lattice structure on $\mathsf{EqRels}(S)$ [29, Thm. 5]. The join of any set $Q \subseteq \mathsf{EqRels}(S)$ is just its intersection; i.e., $\bigvee Q = \{(x,y) \mid (\forall r \in Q)((x,y) \in r)\}$. The meet of Q is the intersection of all equivalence relations which are lesser than each element of Q; i.e., $\bigwedge Q = \bigcap\{r \in \mathsf{EqRels}(S) \mid (\forall s \in Q)(s \preceq_S r)\}$. For two elements, which is the case of most interest in this work, this reduces to $r_1 \wedge r_2 = \bigcap\{r \in \mathsf{EqRels}(S) \mid (r \preceq_S r_1)$ and $(r \preceq_S r_2)\}$, Thus, the meet $r_1 \wedge r_2$ corresponds to the finest partition which is coarser than both $\mathsf{Partition}(r_1)$ and $\mathsf{Partition}(r_2)$; i.e., the greatest lower bound of r_1 and r_2. Further properties of the meet will be examined in Discussion 3.3 and Definition 3.5. As already noted above, $\mathbf{0}_S$ and $\mathbf{1}_S$ are the least and greatest elements of $\mathsf{EqRels}(S)$, respectively; i.e., $\mathbf{0}_S \leq r \leq \mathbf{1}_S$ for all $r \in \mathsf{EqRels}(S)$.

Summary 2.3 (Database schemata, morphisms, and views). One of the beauties of the constant-complement strategy is that its basic formulation requires very little in the way of underlying structure; it is applicable to any *state-based* database model. In such a model, at each point in time there is a single database associated with each schema. Relational, object-oriented, and even XML-based schemata are accommodated. (On the other hand, deductive formalisms, such as those employed in [12], in which the database "state" is modelled not by a single instance, but rather by a set of constraints, are not recaptured directly by such a model.)

Formally, a *(set-based) database schema* $\mathbf{D}$ is just a set. This set, denoted $\mathsf{LDB}(\mathbf{D})$, consists of the *legal databases* of $\mathbf{D}$. Constraints, schema structure, and the like are not represented explicitly. Rather, a database schema is modelled by its instances alone. This model has already been employed in the examples of Sec. 1. For example, $\mathsf{LDB}(\mathbf{E}_0) = \{\emptyset, a, abcd, abd, ac, bcd, bd, c\}$. For the relational example $\mathbf{E}_1$, $\mathsf{LDB}(\mathbf{E}_1)$ is just the set of all finite sets of tuples on ABC which satisfy the join dependency $\bowtie[AB, BC]$.

A *morphism* $f : \mathbf{D}_1 \to \mathbf{D}_2$ of database schemata is a function $f : \mathsf{LDB}(\mathbf{D}_1) \to \mathsf{LDB}(\mathbf{D}_2)$. In the relational context, a morphism is usually defined using the relational algebra or calculus, but the abstraction to a function on states is all that is needed for the purposes of the constant-complement strategy.

A *view* of the database schema $\mathbf{D}$ is a pair $\Gamma = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is a database schema and $\gamma : \mathbf{D} \to \mathbf{V}$ is a surjective database morphism. The views Ω_0, Ω_0', Ω_0'', Ω_2, and Ω_2' introduced in Sec. 1 are each direct examples of this set-based formalism. The views $\Pi^{\mathbf{E}_1}_{AB}$ and $\Pi^{\mathbf{E}_1}_{BC}$ are reduced to this formalism by working with the functions underlying their view morphisms.

Summary 2.4 (View congruences and canonical views). Let $\Gamma = (\mathbf{V}, \gamma)$ be a view of the database schema $\mathbf{D}$. The *congruence* of Γ is $\mathsf{Congr}(\Gamma) = \{(M_1, M_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D}) \mid \gamma(M_1) = \gamma(M_2)\}$. In Figs. 1, 2, 4, and 5, the congruence of each view is shown by ellipses drawn around sets of states belonging to the same block.

Views with identical congruences are the same up to a renaming of their states. Indeed, there is a natural bijective correspondence $\mathsf{CTrans}_\Gamma : \mathsf{LDB}(\mathbf{V}) \to \mathsf{Partition}(\mathsf{Congr}(\Gamma))$ given by $N \mapsto \{M \in \mathsf{LDB}(\mathbf{D}) \mid \gamma(M) = N\}$. In other words, $N \in \mathsf{LDB}(\mathbf{V})$ is in natural correspondence with the block of all states in $\mathsf{LDB}(\mathbf{D})$ which map to it under γ.

As the basic constructions which underlie the constant-complement strategy depend only upon the congruence, and not any further details of the view, it is convenient to regard views which have identical congruences as isomorphic, a convention which will be followed in this paper. It is furthermore possible to identify a canonical representative for each isomorphism class. Specifically, let r be any equivalence relation on $\mathsf{LDB}(\mathbf{D})$. The *canonical view* defined by r is $\mathsf{View}(r) = (\mathsf{Partition}(r), \mathsf{Block}_r)$; in other words, $\mathsf{View}(r)$ has $\mathsf{Partition}(r)$, the set of all blocks of r, as its state set, and the function $\mathsf{Block}_r : M \to \mathsf{Block}_r(M)$ as its database morphism. $\mathsf{Congr}(\mathsf{View}(r)) = r$, just by construction. In particular, $\mathsf{Congr}(\mathsf{View}(\mathsf{Congr}(\Gamma))) = \mathsf{Congr}(\Gamma)$ for any view Γ. In Fig. 1, the canonical view defined by $\mathsf{Congr}(\Omega_0)$ is given by the function $\mathsf{Block}_{\mathsf{Congr}(\Omega_0)} : \mathsf{LDB}(\mathbf{E}_0) \to \mathsf{LDB}(\mathbf{W}_0)$ defined on elements by $\emptyset \mapsto \{\emptyset, c\}$, $c \mapsto \{\emptyset, c\}$, $a \mapsto \{a, ac\}$, $ac \mapsto \{a, ac\}$, $bd \mapsto \{bd, bcd\}$, $bcd \mapsto \{bd, bcd\}$, $abd \mapsto \{abd, abcd\}$, and $abcd \mapsto \{abd, abcd\}$.

The lattice structure identified in Summary 2.2 applies equally well to views, since a canonical view is defined entirely by its congruence. Specifically, for views $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ of $\mathbf{D}$, define $\Gamma_1 \preceq_{\mathbf{D}} \Gamma_2$ iff $\mathsf{Congr}(\Gamma_1) \preceq_{\mathsf{LDB}(\mathbf{D})} \mathsf{Congr}(\Gamma_2)$. The view (up to isomorphism) $\mathbf{1}_{\mathbf{D}}$ whose congruence is $\mathbf{1}_{\mathsf{LDB}(\mathbf{D})}$ is called the *identity view* of $\mathbf{D}$, while the view (up to isomorphism) $\mathbf{0}_{\mathbf{D}}$ whose congruence is $\mathbf{0}_{\mathsf{LDB}(\mathbf{D})}$ is called the *zero view* of $\mathbf{D}$. Clearly, the identity view, which preserves all information about the state of $\mathbf{D}$, is greatest under the ordering $\preceq_{\mathbf{D}}$, while $\mathbf{0}_{\mathbf{D}}$, which preserves no information (since its schema has only one state), is least. The view (up to isomorphism) whose congruence is $\mathsf{Congr}(\Gamma_1) \wedge \mathsf{Congr}(\Gamma_2)$ is called the *meet view* of $\{\Gamma_1, \Gamma_2\}$. The meet of two views is discussed more thoroughly, with an example, in Definition 3.5 and Fig. 7.

Definition 2.5 (View morphisms and relative views). Let $\varGamma_1 = (\mathbf{V}_1, \gamma_1)$ and $\varGamma_2 = (\mathbf{V}_2, \gamma_2)$ be views of the schema $\mathbf{D}$. A *morphism* $f : \varGamma_1 \to \varGamma_2$ is a function $f : \mathsf{LDB}(\mathbf{V}_1) \to \mathsf{LDB}(\mathbf{V}_2)$ with the property that $\gamma_2 = f \circ \gamma_1$. There is at most one such morphism, and it exists iff $\varGamma_2 \preceq_{\mathbf{D}} \varGamma_1$. To see this, it suffices to work with the views $\mathsf{View}(\mathsf{Congr}(\varGamma_i)) = (\mathsf{Partition}(\mathsf{Congr}(\varGamma_i)), \mathsf{Block}_{\mathsf{Congr}(\varGamma_i)})$ for $i \in \{1, 2\}$. If $\mathsf{Congr}(\varGamma_2) \preceq_{\mathsf{LDB}(\mathbf{D})} \mathsf{Congr}(\varGamma_1)$, then the morphism $f : \mathsf{Partition}(\mathsf{Congr}(\varGamma_1)) \to \mathsf{Partition}(\mathsf{Congr}(\varGamma_2))$ given by $\mathsf{Block}_{\mathsf{Congr}(\varGamma_1)}(M) \mapsto \mathsf{Block}_{\mathsf{Congr}(\varGamma_2)}(M)$ for each $M \in \mathsf{LDB}(\mathbf{D})$ is well defined and the only possibility. If $\mathsf{Congr}(\varGamma_2) \not\preceq_{\mathsf{LDB}(\mathbf{D})} \mathsf{Congr}(\varGamma_1)$, then it is not possible to map $\mathsf{Block}_{\mathsf{Congr}(\varGamma_1)}(M)$ to $\mathsf{Block}_{\mathsf{Congr}(\varGamma_2)}(M)$ for all M, with each block mapped to one of its supersets, and so no such f is possible. This unique morphism is of sufficient interest to warrant its own notation. For $\varGamma_2 \preceq_{\mathbf{D}} \varGamma_1$, the *relative morphism* $\mathsf{RelMor}\langle \varGamma_1, \varGamma_2 \rangle : \varGamma_1 \to \varGamma_2$ is illustrated in Fig. 6.

$$\begin{tikzcd}
 & \mathbf{D} \arrow[dl, "\gamma_1"'] \arrow[dr, "\gamma_2"] & \\
\mathbf{V}_1 \arrow[rr, dashed, "\mathsf{RelMor}\langle \varGamma_1, \varGamma_2 \rangle"'] & & \mathbf{V}_2
\end{tikzcd}$$

Fig. 6. Visualization of a relative view morphism

In the case that there is such a morphism, $\mathsf{RelView}(\varGamma_1, \varGamma_2) = (\mathbf{V}_2, \mathsf{RelMor}\langle \varGamma_1, \varGamma_2 \rangle)$ is a view of $\mathbf{V}_1$, called the *relative view* of $\varGamma_1$ induced by $\varGamma_2$.

As a concrete example, consider the schema $\mathbf{E}_1$ introduced in Sec. 1, together with the views $\varPi^{\mathbf{E}_1}_{AB}$ and $\varPi^{\mathbf{E}_1}_{BC}$, and the additional view $\varPi^{\mathbf{E}_1}_{B}$ which is the projection of $R[ABC]$ onto B. Then $\varPi^{\mathbf{E}_1}_{B} \preceq_{\mathbf{E}_1} \varPi^{\mathbf{E}_1}_{AB}$ and $\varPi^{\mathbf{E}_1}_{B} \preceq_{\mathbf{E}_1} \varPi^{\mathbf{E}_1}_{BC}$. The relative view morphism $\mathsf{RelView}(\varPi^{\mathbf{E}_1}_{AB}, \varPi^{\mathbf{E}_1}_{B})$ sends a relation on attributes AB to its projection onto attribute B. $\mathsf{RelView}(\varPi^{\mathbf{E}_1}_{BC}, \varPi^{\mathbf{E}_1}_{B})$ is defined similarly. Relative morphisms will appear again in Definition 3.5.

Definition 2.6 (View complements). Complements have already been introduced informally in Sec. 1; $\varOmega'_0$ and $\varOmega''_0$ are complements of $\varOmega_0$, $\varOmega'_2$ is a complement of $\varOmega_2$, and $\varPi^{\mathbf{E}_1}_{BC}$ is a complement of $\varPi^{\mathbf{E}_1}_{AB}$. This idea is placed on firm formal ground as follows.

Let $\mathbf{D}$ be a database schema and $\varGamma = (\mathbf{V}, \gamma)$ a view of $\mathbf{D}$. A *complement* of $\varGamma$ is a view $\varGamma' = (\mathbf{V}', \gamma')$ with the property that $\varGamma \vee \varGamma' = \mathbf{1}_{\mathbf{D}}$. Equivalently, $\varGamma'$ is a complement of $\varGamma$ if the *decomposition function* $\gamma \times \gamma' : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V}')$ given on databases by $M \mapsto (\gamma(M), \gamma'(M))$ is injective. This definition is clearly symmetric; if $\varGamma'$ is a complement of $\varGamma$, then $\varGamma$ is a complement of $\varGamma'$. It is thus appropriate to call $\{\varGamma, \varGamma'\}$ a *complementary pair.*

Definition 2.7 (Updates, update strategies, and reflectors). Although the notion of an update to a database is a simple one, it is nevertheless necessary to have some compact and precise notation in order to formulate properties of view updates and their reflections. The following definitions are designed to serve that purpose.

An *update* on the database schema $\mathbf{D}$ 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. The set of all updates on $\mathbf{D}$ is denoted $\mathsf{Updates}(\mathbf{D})$. To describe the situation surrounding an update request on the view $\Gamma = (\mathbf{V}, \gamma)$, 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 current state of the main schema) and $N_2 \in \mathsf{LDB}(\mathbf{V})$ (the new state of the view schema). The set of all update requests from Γ to $\mathbf{D}$ is denoted $\mathsf{UpdReq}(\Gamma)$.

A *realization* of $(M_1, N_2) \in \mathsf{UpdReq}(\Gamma)$ 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 $(\gamma(M_1), N_2)$. Thus, the realization (M_1, M_2) tells how to reflect the view update $(\gamma(M_1), N_2)$ to the main schema $\mathbf{D}$ when the state of $\mathbf{D}$ is M_1.

A *reflector* for Γ is a partial function $\mathfrak{u} : \mathsf{UpdReq}(\Gamma) \to \mathsf{LDB}(\mathbf{D})$ with the property that for any $(M, N) \in \mathsf{UpdReq}(\Gamma)$, if $\mathfrak{u}(M, N)\downarrow$, then $(M, \mathfrak{u}(M, N))$ is a realization of (M, N) along Γ.

Definition 2.8 (The constant-complement view-update strategy). Let $\mathbf{D}$ be a database schema, and let $\{\Gamma_1, \Gamma_2\}$ be a complementary pair of views of $\mathbf{D}$, with $\Gamma_i = (\mathbf{V}_i, \gamma_i)$ for $i \in \{1, 2\}$. The *constant-complement reflector* for $\langle \Gamma_1, \Gamma_2 \rangle$ is the reflector $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ for Γ_1 given on elements by

$$\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N) = \begin{cases} (\gamma_1 \times \gamma_2)^{-1}(N, \gamma_2(M)) & \text{if } (\gamma_1 \times \gamma_2)^{-1}(N, \gamma_2(M))\downarrow \\ \text{undefined} & \text{otherwise} \end{cases}$$

Definition 2.9 (Properties of reflectors). Let $\mathfrak{u}$ be a reflector for the view Γ of schema $\mathbf{D}$.

The reflector $\mathfrak{u}$ *respects identities* if for any $M \in \mathsf{LDB}(\mathbf{D})$, $\mathfrak{u}(M, \gamma(M)) = M$. This implies in particular that $\mathfrak{u}(M, \gamma(M))\downarrow$.

The reflector $\mathfrak{u}$ exhibits *path independence* if for any $M \in \mathsf{LDB}(\mathbf{D})$ and $N_1, N_2 \in \mathsf{LDB}(\boldsymbol{\Gamma})$, if both $\mathfrak{u}(M, N_1)\downarrow$ and $\mathfrak{u}(\mathfrak{u}(M, N_1), N_2)\downarrow$, then $\mathfrak{u}(M, N_2)\downarrow$ as well, with $\mathfrak{u}(M, N_2) = \mathfrak{u}(\mathfrak{u}(M, N_1), N_2)$. A similar property is called *consistency* in [14, Def. 2.7].

The reflector $\mathfrak{u}$ exhibits *reversibility* if for any $(M, N) \in \mathsf{UpdReq}(\Gamma)$, if $\mathfrak{u}(M, N)\downarrow$, then so too is $\mathfrak{u}(\mathfrak{u}(M, N), \gamma(M))$ with $\mathfrak{u}(\mathfrak{u}(M, N), \gamma(M)) = M$.

Define the set of *realizations* or *reflected updates* induced by $\mathfrak{u}$ to be $\mathsf{ReflUpd}\langle \mathfrak{u} \rangle = \{(M, \mathfrak{u}(M, N)) \mid (M, N) \in \mathsf{UpdReq}(\Gamma) \text{ and } \mathfrak{u}(M, N)\downarrow\}$.

The following is easy to verify.

Observation 2.10 (Realizations and equivalence relations). *Let* $\mathfrak{u}$ *be a reflector for the view* Γ *of schema* $\mathbf{D}$. *Then* $\mathsf{ReflUpd}\langle\mathfrak{u}\rangle$ *is an equivalence relation on* $\mathsf{LDB}(\mathbf{D})$ *iff* $\mathfrak{u}$ *respects identities and exhibits both path independence and reversibility.* □

Theorem 2.11 (Characterization of constant-complement reflectors). *Let* $\mathfrak{u}$ *be a reflector for the view* Γ *of schema* $\mathbf{D}$. *Then there is a view* Γ' *of* $\mathbf{D}$ *with the property that* $\mathfrak{u} = \mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}$ *iff* $\mathsf{ReflUpd}\langle\mathfrak{u}\rangle$ *is an equivalence relation. In the case that such a* Γ' *exists, it is given by* $\mathsf{View}(\mathsf{ReflUpd}\langle\mathfrak{u}\rangle)$.

PROOF SKETCH: A full, formal proof may be found in [5], but is really a very simple verification. If Γ' is a complement of Γ, then $\mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}$ respects identities (because doing nothing keeps Γ constant), exhibits path independence (since the composition of operations which keep Γ' constant keeps it constant as well), and exhibits reversibility (since undoing an operation which kept Γ' constant still keeps it constant). Thus, Observation 2.10 may be invoked to show that for any complement Γ' of Γ, $\mathsf{ReflUpd}\langle\mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}\rangle$ forms an equivalence relation.

In the other direction, if $\mathsf{ReflUpd}\langle\mathfrak{u}\rangle$ is an equivalence relation, then it defines a view $\mathsf{View}(\mathsf{ReflUpd}\langle\mathfrak{u}\rangle)$, and the updates to Γ which keep it constant are precisely those which limit the state changes of $\mathbf{D}$ to those which lie in the congruence of that view, i.e., $\mathsf{ReflUpd}\langle\mathfrak{u}\rangle$. □

3 Admissibility Invariance

Admissibility invariance is a fundamental issue in the support of view updates. As already noted in the introduction, it addresses the question of whether the admissibility of a proposed view update depends only upon the view state, or whether it depends also upon the state of the complement. Although the results presented in this section are not new, many of them are found only in research papers which intertwine them with other, more advanced concepts. It therefore seems appropriate to give them a relatively simple and unified presentation. The papers [17] and [24] are the primary sources for further information.

Notation 3.1. Throughout this section, unless stated specifically to the contrary, take $\mathbf{D}$ to be a database schema, with $\Gamma = (\mathbf{V}, \gamma)$, $\Gamma' = (\mathbf{V}', \gamma')$, $\Gamma_1 = (\mathbf{V}_1, \gamma_1)$, and $\Gamma_2 = (\mathbf{V}_2, \gamma_2)$ views over $\mathbf{D}$.

Definition 3.2 (Admissibility invariance). Let $\mathfrak{u}$ be a reflector for Γ. Define

(a) $\mathsf{ViewUpd}_{\exists}(\mathfrak{u}) =$
$$\{(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V}) \mid (\exists M \in \gamma^{-1}(N_1))(\mathfrak{u}(M, N_2){\downarrow})\}.$$

(b) $\mathsf{ViewUpd}_{\forall}(\mathfrak{u}) =$
$$\{(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V}) \mid (\forall M \in \gamma^{-1}(N_1))(\mathfrak{u}(M, N_2){\downarrow})\}.$$

(c) Say that the reflector $\mathfrak{u}$ exhibits *admissibility invariance* if $\mathsf{ViewUpd}_{\exists}(\mathfrak{u}) = \mathsf{ViewUpd}_{\forall}(\mathfrak{u})$.

Discussion 3.3 (Characterization of the meet and commuting congruences). Recall from Summary 2.2 that the meet $r_1 \wedge r_2$ of two equivalence relations r_1 and r_2 on a set S is the largest equivalence relation which is smaller than both r_1 and r_2. This may be computed explicitly by iterating r_1 and r_2. Specifically, $(x, y) \in r_1 \wedge r_2$ iff there is a chain

$$(x, x_1), (x_1, x_2), (x_2, x_3), \ldots, (x_{k-1}, x_k), (x_k, y)$$

in which each pair is either in r_1 or else in r_2 [29, Thm. 5]. Translating to views, this means that a pair $(M, M') \in \mathsf{Congr}(\mathit{\Gamma}_1) \wedge \mathsf{Congr}(\mathit{\Gamma}_2)$ iff there is a chain

$$(M, M_1), (M_1, M_2), (M_2, M_3), \ldots, (M_{k-1}, M_k), (M_k, M')$$

with each pair in either $\mathsf{Congr}(\mathit{\Gamma}_1)$ or else $\mathsf{Congr}(\mathit{\Gamma}_2)$. If the congruences *commute*, that is, if $\mathsf{Congr}(\mathit{\Gamma}_1) \circ \mathsf{Congr}(\mathit{\Gamma}_2) = \mathsf{Congr}(\mathit{\Gamma}_2) \circ \mathsf{Congr}(\mathit{\Gamma}_1)$, with $\circ$ denoting ordinary composition of relations, then this characterization reduces to $(M, M') \in \mathsf{Congr}(\mathit{\Gamma}_1) \wedge \mathsf{Congr}(\mathit{\Gamma}_2)$ iff $(M, M') \in \mathsf{Congr}(\mathit{\Gamma}_1) \circ \mathsf{Congr}(\mathit{\Gamma}_2)$ iff $(M, M') \in \mathsf{Congr}(\mathit{\Gamma}_2) \circ \mathsf{Congr}(\mathit{\Gamma}_1)$.

If $\{\mathit{\Gamma}_1, \mathit{\Gamma}_2\}$ is complementary pair with commuting congruences, then it is called a *meet-complementary pair.*

A connection between commuting congruences and reflection invariance may not seem apparent at first. The idea is the following. For a complementary pair $\{\mathit{\Gamma}_1, \mathit{\Gamma}_2\}$, with $\mathit{\Gamma}_1$ to be updated with constant complement $\mathit{\Gamma}_2$, and (N_1, N_2) a view update on $\mathit{\Gamma}_1$, blocks of $\mathsf{Congr}(\mathit{\Gamma}_2)$ represent transitions for constant complement $\mathit{\Gamma}_1$ (e.g., a reflection of (N_1, N_2) for a fixed $M_1 \in {\gamma_1}^{-1}(N_1)$), while blocks of $\mathsf{Congr}(\mathit{\Gamma}_1)$ represent switching between alternate choices for M_1. If the congruences commute, then the choices may be made in either order. Thus, if $(M_1, N) \in \mathsf{UpdReq}(\mathit{\Gamma}_1)$ has a constant-complement realization $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(M_1, N) = M_2$, then $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(M'_1, N)$ must exist for any $M'_1 \in \mathsf{Block}_{\mathsf{Congr}(\mathit{\Gamma}_1)}(M_1)$ since instead of first going from $(M_1, M'_1) \in \mathsf{Congr}(\mathit{\Gamma}_1)$ and then doing the update defined by $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(M'_1, N)$ (which has the value M'_2, say), the transition $(M_1, M_2) \in \mathsf{Congr}(\mathit{\Gamma}_2)$ may be made first, with assurance that the transition (M_2, M'_2) is in $\mathsf{Congr}(\mathit{\Gamma}_1)$. This may be visualized with the aid of Fig. 2. Note that if it is possible to move from one state of $\mathbf{E}_0$ to another by moving through connected ellipses, then it is possible by first going through a shaded ellipse (representing $\mathsf{Congr}(\mathit{\Omega}_0)$) and then a dotted one (representing $\mathsf{Congr}(\mathit{\Omega}'_0)$), or by going through a dotted one first and then a shaded one. For example, since the view update $(\emptyset, a) \in \mathsf{Updates}(\mathbf{W}_0)$ is supported for current state $c \in \mathsf{LDB}(\mathbf{E}_0)$ as the reflection $(c, ac) \in \mathsf{Updates}(\mathbf{E}_0)$, it must also be supported for current state $\emptyset \in \mathsf{LDB}(\mathbf{E}_0)$ as $(\emptyset, c) \circ (c, ac) \circ (ac, a) \in \mathsf{Congr}(\mathit{\Omega}_0) \circ \mathsf{Congr}(\mathit{\Omega}'_0) \circ \mathsf{Congr}(\mathit{\Omega}_0) = (\emptyset, a) \in \mathsf{Updates}(\mathbf{E}'_0)$.

On the other hand, in Fig. 4, which represents a situation without commuting congruences. this property does not hold. It is possible to go from c to a by first going through a shaded ellipse and then a dotted one, but not through a dotted one followed by a shaded one. Thus, while view update $(\emptyset, a) \in \mathsf{Updates}(\mathbf{W}_0)$ is supported for current state $\emptyset \in \mathsf{LDB}(\mathbf{E}_0)$ as the reflection $(\emptyset, a) \in \mathsf{Updates}(\mathbf{E}_0)$, this view update is not supported for current state $c \in \mathsf{LDB}(\mathbf{E}_0)$.

The formal characterization is given below. Although [4] contains related ideas in the form of *weak independence*, it was first stated in its form in [15, 1.13], albeit without proof. A detailed proof in the context of ordered views was given in [17, Thm. 2.14], with an alternate proof in [24, Thm. 3.8]. Because if its importance, as because it might still otherwise seem mysterious that admissibility invariance and commuting congruences could possibly have anything to do with one another, a proof sketch is given here as well.

Theorem 3.4 (Fundamental characterization of admissibility). *Let* $\{\Gamma_1, \Gamma_2\}$ *be a complementary pair. Then the reflector* $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ *exhibits admissibility invariance iff* $\{\Gamma_1, \Gamma_2\}$ *has commuting congruences; i.e., iff it is a meet-complementary pair.*

PROOF SKETCH: Choose $(M_1, M_2') \in \mathsf{Congr}(\Gamma_1)$ and $(M_2', M_2) \in \mathsf{Congr}(\Gamma_2)$. Then, in view of Theorem 2.11, $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_2', \gamma_1(M_2)) = M_2$. Now if $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ exhibits admissibility invariance, $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_1, \gamma_1(M_2))$ must also be defined. Letting $M_1' = \mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_1, \gamma_1(M_2))$, $(M_2, M_1') \in \mathsf{Congr}(\Gamma_1)$ follows immediately, and applying Theorem 2.11 again yields $(M_1, M_1') \in \mathsf{Congr}(\Gamma_2)$. Hence $(M_1, M_1') \circ (M_1', M_2) \in \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$; so that $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) \subseteq \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$. The reverse inclusion is proved similarly, whence $\mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2) = \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$.

In the opposite direction, assume that $\{\Gamma_1, \Gamma_2\}$ has commuting congruences, and let $M_1, M_2 \in \mathsf{LDB}(\mathbf{D})$, $N \in \mathsf{LDB}(\mathbf{V}_1)$ with $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_1, N) = M_2$. Choose any $M_1' \in \mathsf{Block}_{\mathsf{Congr}(\Gamma_1)}(M_1)$. Then $(M_1', M_2) = (M_1', M_1) \circ (M_1, M_2) \in \mathsf{Congr}(\Gamma_1) \circ \mathsf{Congr}(\Gamma_2)$. By assumption, $(M_1', M_2) \in \mathsf{Congr}(\Gamma_2) \circ \mathsf{Congr}(\Gamma_1)$ as well. Hence, there is an $M_2' \in \mathsf{LDB}(\mathbf{D})$ with $(M_1', M_2') \in \mathsf{Congr}(\Gamma_2)$ and $(M_2', M_2) \in \mathsf{Congr}(\Gamma_1)$, with the former and Theorem 2.11 implying that $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_1', \gamma(M_2') = M_2'$ and the latter implying that $\gamma_1(M_2') = \gamma(M_2) = N$, so that $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M_1', N) = M_2$, whence $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ exhibits admissibility invariance. □

Definition 3.5 (Meet admissibility). If a reflector $\mathfrak{u}$ exhibits admissibility invariance, then since the admissibility of a view update depends only upon that update, and no external information, it is appropriate to ask how to describe the set of admissible updates in terms of the view alone. It turns out that there is a very simple and most useful characterization. Namely, the allowed view updates are precisely those which keep the relative view induced by the complement constant. This idea has already been illustrated in the context of a simple relational example in Fig. 3, in which the updates to the view $\Pi^{\mathbf{E}_1}_{AB}$ with constant complement $\Pi^{\mathbf{E}_1}_{BC}$ are precisely those which hold the meet view $\Pi^{\mathbf{E}_1}_{B}$ constant.

Formally, let $\{\Gamma_1, \Gamma_2\}$ be a complementary pair, and define

$$\mathsf{ViewUpd}_{\wedge}(\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}) = \{(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_1) \mid \mathsf{RelMor}\langle \Gamma_1, \Gamma_1 \wedge \Gamma_2 \rangle(N_1) = \mathsf{RelMor}\langle \Gamma_2, \Gamma_1 \wedge \Gamma_2 \rangle(N_2)\}.$$

Call the reflector $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ *meet admissible* if $\mathsf{ViewUpd}_{\wedge}(\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}) = \mathsf{ViewUpd}_{\forall}(\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle})$, and say that it exhibits *meet admissibility*. In other words, $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ exhibits meet admissibility iff the updates allowed under constant complement are precisely those which hold $\mathsf{RelView}(\Gamma_1, \Gamma_1 \wedge \Gamma_2)$ constant.

Figure. 7 provides an illustration of meet for the schema and views of Fig. 2. The meet $\Omega_0 \wedge \Omega_0'$ recaptures whether or not bd occurs in the state of the main schema. Since a stipulated constraint on $\mathbf{W}_0$ is that b and d must occur together or not at all, either of the views Ω_0 (which recaptures whether or not b is a subset of the state of $\mathbf{W}_0$) and Ω_0' (which recaptures whether or not d is a subset of the state of $\mathbf{W}_0$), by itself, contains enough information to determine whether or not bd is a subset of the state of $\mathbf{W}_0$.

The equivalence classes of $\mathsf{RelView}(\Omega_0, \Omega_0 \wedge \Omega_0')$ and $\mathsf{RelView}(\Omega_0', \Omega_0 \wedge \Omega_0')$ are shown explicitly in the corresponding view, together to their common target in the meet. Note in particular how each block within Ω_0 (representing the legal constant-complement updates to that view) is paired with the blocks of Ω_0', which represent possible alternatives for completing the state of the main schema. It is easy to see that keeping the meet $\Omega_0 \wedge \Omega_0'$ constant; that is, keeping whether or not bd is a subset of the current state of $\mathbf{W}_0$, is exactly what is needed to identify the legal constant-complement updates.

A proof of the following will not be given, although the idea is hopefully apparent from the above discussion. Rather, the interested reader is referred to the given citations.

Proposition 3.6 (Admissibility invariance and meet admissibility). *Let* $\{\Gamma_1, \Gamma_2\}$ *be a complementary pair. Then the reflector* $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ *is meet admissible iff* $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ *exhibits admissibility invariance.*

Proof. A direct proof may be found in [24, Thm. 3.8], while the equivalence to commuting congruences is shown in [17, Thm. 2.14]. □

Definition 3.7 (Constraint independence). In the context of schemata with constraints, admissibility invariance for a complementary pair $\{\Gamma_1, \Gamma_2\}$ may be characterized via embedded covers. Specifically, the pair $\{\Gamma_1, \Gamma_2\}$ admits an embedded cover of the constraints $\mathsf{Constr}(\mathbf{D})$ if those constraints may be inferred from the constraint sets $\mathsf{Constr}(\mathbf{V}_1)$ and $\mathsf{Constr}(\mathbf{V}_2)$ of the view schemata $\mathbf{V}_1$ and $\mathbf{V}_2$, via inversion of the decomposition mapping $\gamma_1 \times \gamma_2$. This idea was first put forward in the context of FDs and projections in [31, Thm. 2], and is discussed in a general way in [18], but there is a way to capture the essence of the idea without any recourse to constraints at all. Specifically, it may be axiomatized that any state $N_1 \in \mathsf{LDB}(\mathbf{V}_1)$ may be paired with any state $N_2 \in \mathsf{LDB}(\mathbf{V}_2)$ to identify a corresponding state of $\mathbf{D}$ represented as $(\gamma_1 \times \gamma_2)^{-1}(N_1, N_2)$. The only requirement is that (N_1, N_2) form a "compatible" pair. The formal definition is as follows.

Let $\{\Gamma_1, \Gamma_2\}$ be a complementary pair. Say that it exhibits *constraint independence* if for any $N_1, N_1' \in \mathsf{LDB}(\mathbf{V}_1)$ and $N_2, N_2' \in \mathsf{LDB}(\mathbf{V}_2)$, if any three of $(N_1, N_2), (N_1, N_2'), (N_1', N_2), (N_1', N_2')$ are the images of elements of $\mathsf{LDB}(\mathbf{D})$ under $\gamma_1 \times \gamma_2$, then so too is the fourth.

Note how this excludes general, cross-view constraints. If, say, $(N_1, N_2) \in \mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_2)$ is not compatible, then for any other pair $(N_1', N_2') \in$

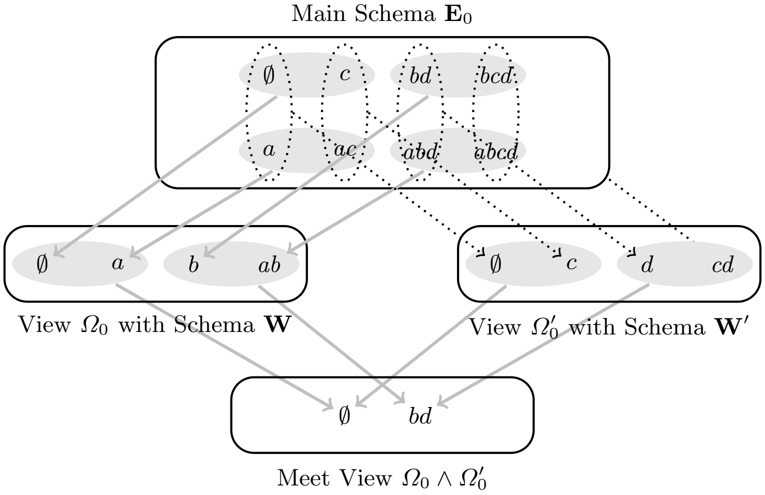
Main Schema $\mathbf{E}_0$
$\emptyset$
c
bd
bcd
a
ac
abd
$abcd$
$\emptyset$
a
b
ab
$\emptyset$
c
d
cd
View $\mathit{\Omega}_0$ with Schema $\mathbf{W}$
View $\mathit{\Omega}_0'$ with Schema $\mathbf{W}'$
$\emptyset$
bd
Meet View $\mathit{\Omega}_0 \wedge \mathit{\Omega}_0'$

Fig. 7. Visualization of a meet-complementary pair

$\mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_2)$, at least one of (N_1, N_2') and (N_1', N_2) must not be compatible either. The lack of compatibility of (N_1, N_2) cannot thus be due to any special properties of this pair, since it mandates non-compatibility of at least one other pair with at least one arbitrary component.

In terms of the example depicted in Fig. 2 and Fig. 7, $(N_1, N_2) \in (\omega_0 \times \omega_0')(\mathsf{LDB}(\mathbf{E}_0))$ iff the condition $(b \in N_1 \Leftrightarrow d \in N_2)$ holds. From this, it is easy to see that if $(N_1, N_2), (N_1, N_2'), (N_1', N_2) \in (\omega_0 \times \omega_0')(\mathsf{LDB}(\mathbf{E}_0))$, then $b \in N_1$ iff $b \in N_1'$ iff $d \in N_2$ iff $d \in N_2'$, whence $(N_1', N_2') \in (\omega_0 \times \omega_0')(\mathsf{LDB}(\mathbf{E}_0))$,

In terms of the relational example of Fig. 3, $\mathit{\Pi}^{\mathbf{E}_1}_{AB} \wedge \mathit{\Pi}^{\mathbf{E}_1}_{BC} = \mathit{\Pi}^{\mathbf{E}_1}_{B}$, i.e., the meet of the AB and BC projection is the projection onto the common column B. For two relations $N_1 \in \mathsf{LDB}(\mathbf{E}_1^{AB})$ and $N_2 \in \mathsf{LDB}(\mathbf{E}_1^{BC})$, $(N_1, N_2) \in (\pi^{\mathbf{E}_1}_{AB} \times \pi^{\mathbf{E}_1}_{BC})(\mathsf{LDB}(\mathbf{E}_1))$ iff $\mathsf{RelMor}\langle \mathit{\Pi}^{\mathbf{E}_1}_{AB}, \mathit{\Pi}^{\mathbf{E}_1}_{B}\rangle(N_1) = \mathsf{RelMor}\langle \mathit{\Pi}^{\mathbf{E}_1}_{BC}, \mathit{\Pi}^{\mathbf{E}_1}_{B}\rangle(N_2)$; that is, N_1 and N_2 are compatible if they agree on the common column B. This is none other than the classical condition for a lossless join.

The formal result is as follows.

Proposition 3.8 (Admissibility invariance and constraint independence). *Let $\{\mathit{\Gamma}_1, \mathit{\Gamma}_2\}$ be a complementary pair. Then $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}$ exhibits admissibility invariance iff $\{\mathit{\Gamma}_1, \mathit{\Gamma}_2\}$ exhibits constraint independence.*

Proof. A direct proof may be found in [24, Thm. 3.8], while the equivalence to commuting congruences is shown in [17, Thm. 2.14]. □

4 Reflection Invariance

The constant-complement strategy requires a choice of complement, and that has always been its Achilles' heel. Distinct complements may give rise not only to distinct sets of supported view updates, but, as illustrated in Fig. 5, to distinct translations of the same view update as well. Nevertheless, it is difficult to construct examples which reflect realistic data modelling yet which exhibit alternate translations of a given view update. There always seems to be one "natural" choice, with the others appearing to be contrived mathematical artifacts which are of dubious use in terms of any reasonable modelling. In this section, this issue is examined in some detail. First, the main known result is recalled and extended; that if attention is restricted to situations in which databases have order and database morphisms preserve that order, then order updates — insertions and deletions — always have reflections which are independent of the choice of complement. Subsequently, new results on how to extend this basic result beyond simple insertions and deletions are developed.

The main reference for the known results in [17], although the approach taken in that paper is some different, and undeniably more complex, than that given here. Thus, the first part of this section also serves as a simpler tutorial on the basic aspects of incorporating order into the constant-complement approach.

Definition 4.1 (Reflection invariance). Before venturing into a description of order properties, it is appropriate to provide a formal definition of reflection invariance which applies to the completely general case. Let $\mathbf{D}$ be a database schema, let $\mathit{\Gamma}$ be a view of $\mathbf{D}$, and let $\mathcal{U}$ be a set of reflectors for $\mathit{\Gamma}$.

(a) The set $\mathcal{U}$ is said to exhibit *reflection invariance* with respect to an update request $u \in \mathsf{UpdReq}(\mathit{\Gamma})$ if for any $\mathfrak{u}_1, \mathfrak{u}_2 \in \mathcal{U}$, if both $\mathfrak{u}_1(u)\downarrow$ and $\mathfrak{u}_2(u)\downarrow$, then $\mathfrak{u}_1(u) = \mathfrak{u}_2(u)$.

(b) The set $\mathcal{U}$ is said to exhibit *reflection invariance* with respect to a set $U \subseteq \mathsf{UpdReq}(\mathit{\Gamma})$ if it exhibits reflection invariance for each $u \in U$.

Examples 4.2 (The utility of order). To illustrate the utility of respecting order properties in the constant-complement strategy, a simple example, first presented in [16, 1.1.1 and 1.1.2], is recalled. Let $\mathbf{E}_3$ be the relational schema comprised of two unary relation symbols $R[A]$ and $S[A]$; there are no constraints other than that these two relations share the same domain. Let $\mathit{\Pi}_R^{\mathbf{E}_3} = (\mathbf{E}_3^R, \pi_R^{\mathbf{E}_3})$ be the view which preserves R while discarding S entirely, and define $\mathit{\Pi}_S^{\mathbf{E}_3} = (\mathbf{E}_3^S, \pi_S^{\mathbf{E}_3})$ similarly. Thus, a state $M \in \mathsf{LDB}(\mathbf{E}_3)$ is represented as a pair (M_R, M_S) in which M_R is the relation for R and M_S is the relation for S, with $\pi_R^{\mathbf{E}_3}(M_R, M_S) = M_R$, and $\pi_S^{\mathbf{E}_3}(M_R, M_S) = M_S$. It is immediate that $\{\mathit{\Pi}_R^{\mathbf{E}_3}, \mathit{\Pi}_S^{\mathbf{E}_3}\}$ forms a meet complementary pair, with meet the zero view $\mathbf{0}_{\mathbf{E}_3}$. Any update may be made to $R[A]$ while holding $S[A]$ constant. There cannot be a simpler and more natural complement of $\mathit{\Pi}_R^{\mathbf{E}_3}$ than $\mathit{\Pi}_S^{\mathbf{E}_3}$.

Nevertheless, there are other complements. Define the view $\mathit{\Pi}_{R \mathbin{\Delta} S}^{\mathbf{E}_3} = (\mathbf{E}_3^{R \mathbin{\Delta} S}, \pi_{R \mathbin{\Delta} S}^{\mathbf{E}_3})$ to have the schema $\mathbf{E}_3^{R \mathbin{\Delta} S}$ which consists of the single unary relation symbol $T[A]$, with the view mapping $\pi^{\mathbf{E}_3}_{R \,\Delta\, S}$ defined by the symmetric difference; thus $\pi^{\mathbf{E}_3}_{R \,\Delta\, S}(M_R, M_S) = M_R \,\Delta\, M_S = (M_R \setminus M_S) \cup (M_S \setminus M_R)$. It is not difficult to see that $\{\mathit{\Pi}^{\mathbf{E}_3}_R, \mathit{\Pi}^{\mathbf{E}_3}_{R \,\Delta\, S}\}$ also forms a meet-complementary pair with meet $\mathbf{0}_{\mathbf{E}_3}$. Indeed, $M_S = M_R \,\Delta (M_R \,\Delta\, M_S)$.

Using $\mathit{\Pi}^{\mathbf{E}_3}_{R \,\Delta\, S}$ as a complement to $\mathit{\Pi}^{\mathbf{E}_3}_R$ leads to unnatural view updates. For example, if the state of the main schema is $M_{31} = \{R(\mathrm{a}_1), S(\mathrm{b}_1)\}$, then the view update $(\{R(\mathrm{a}_1)\}, \emptyset)$ to $\mathit{\Pi}^{\mathbf{E}_3}_R$ — the deletion of $R(\mathrm{a}_1)$ with constant complement $\mathit{\Pi}^{\mathbf{E}_3}_{R \,\Delta\, S}$ — requires the insertion of a_1 into $S[A]$, so that the new state after the update must be $M_{32} = \{S(\mathrm{b}_1), S(\mathrm{a}_1)\}$, and not the much more natural $\{S(\mathrm{b}_1)\}$ mandated by constant complement $\mathit{\Pi}^{\mathbf{E}_3}_S$.

It is difficult to imagine a modelling situation in which a complement of the form $\mathit{\Pi}^{\mathbf{E}_3}_{R \,\Delta\, S}$ would be preferred to $\mathit{\Pi}^{\mathbf{E}_3}_S$. Nevertheless, from a set-based mathematical perspective, there is no ground to choose one over the other. However, if order is considered, the preference for $\mathit{\Pi}^{\mathbf{E}_3}_S$ becomes immediate. Regard the states of the schemata as being ordered under relation-by-relation inclusion. Then both $\pi^{\mathbf{E}_3}_R$ and $\pi^{\mathbf{E}_3}_S$ are order-preserving morphisms, while $\pi^{\mathbf{E}_3}_{R \,\Delta\, S}$ is not. Since view morphisms which occur typically in the relational model, such as the *SPJR morphisms* which are defined using the operations or selection, projection, join, and renaming, are order preserving, restricting attention to such views is a natural step. As will be seen, this choice leads to interesting results on reflection invariance.

It is important to note that reflection invariance only addresses the issue of when the translation of a given view update, or set of view updates, is independent of the choice of complement. It does not address the issue of whether a single complement supports a given set of updates. A simple example, taken from [20, Sec. 1], illustrates this difference. Let $\mathbf{E}_4$ have the single relation symbol $R[ABCD]$, constrained by the FDs in $\{B \to D, C \to D\}$. The three projective views $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$, $\mathit{\Pi}^{\mathbf{E}_4}_{BD}$, and $\mathit{\Pi}^{\mathbf{E}_4}_{CD}$ are defined in the obvious way. Using the classical characterization [31, Thm. 1], it is easy to see that both $\mathit{\Pi}^{\mathbf{E}_4}_{BD}$ and $\mathit{\Pi}^{\mathbf{E}_4}_{CD}$ are complements of $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$. Each of these views is order preserving, and the results of Theorem 4.11 and Theorem 4.14 established below ensure that, at least for update requests which are realized by the composition of at most one insertion and one deletion, the reflection of that view update on $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$ is independent of which of these two complements is chosen. However, the two complements support distinct sets of updates. Indeed, with constant complement $\mathit{\Pi}^{\mathbf{E}_4}_{CD}$, the updates to $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$ are those which keep the projection onto C constant, while those with constant complement $\mathit{\Pi}^{\mathbf{E}_4}_{BD}$ must keep B constant. The theory only guarantees that the updates which are common to the two — those which keep both the projection onto B as well as the projection onto C constant — have the same translation. Although these pairs are not meet complements, it is possible to obtain similar, but slightly more complex examples, which are meet complements. See [23] for details, as well as for a study of the conditions under which a single complement can handle all possible updates definable via constant complement.

Definition 4.3 (Partially ordered sets). Since the theory is based upon order, it is pertinent to recall the basic ideas and notation, as well as to introduce some special but useful extensions. Familiarity with the fundamental ideas of posets, such as presented in [10], is presumed; only a few notational and terminological points are reviewed here. A *partially ordered set* (*poset*) is a pair $\mathbf{P} = (P, \leq_{\mathbf{P}})$ in which P is a set and $\leq_{\mathbf{P}}$ is a reflexive, transitive, and antisymmetric relation on P. Given posets $\mathbf{P} = (P, \leq_{\mathbf{P}})$ and $\mathbf{Q} = (Q, \leq_{\mathbf{Q}})$, a *morphism* $f : \mathbf{P} \to \mathbf{Q}$ is a *monotone* function $f : P \to Q$; i.e., $p_1, p_2 \in P$ with $p_1 \leq_{\mathbf{P}} p_2$ implies that $f(p_1) \leq_{\mathbf{Q}} f(p_2)$. The morphism f is *open* if, for any $q_1, q_2 \in Q$ with $q_1 \leq_{\mathbf{Q}} q_2$, there are $p_1 \in f^{-1}(q_1)$, $p_2 \in f^{-1}(q_2)$ with $p_1 \leq_{\mathbf{P}} p_2$. In other words, f is open if Q carries the weakest order which renders f a morphism.

The morphism f is an *isomorphism* iff it has both left and right inverses. It is easily verified that this is equivalent to being an open bijection.

The morphism is f is an *embedding* if it is an open injection, so that the image $f(\mathbf{P})$ is isomorphic to $\mathbf{P}$; i.e., $\mathbf{Q}$ contains an isomorphic copy of $\mathbf{P}$.

If $\mathbf{P} = (P, \leq_{\mathbf{P}})$ is a poset, then so too is its *dual* $\mathbf{P} = (P, \geq_{\mathbf{P}})$, with $p_1 \geq_{\mathbf{P}} p_2$ iff $p_2 \leq_{\mathbf{P}} p_1$. Thus, associated with each property a dual one. Duality will be used throughout this section.

Given a poset $\mathbf{P} = (P, \leq_{\mathbf{P}})$ and $S \subseteq P$, an *upper bound* for S is any $b \in P$ with $s \leq_{\mathbf{P}} b$ for all $s \in S$. The set of all upper bounds of S is denoted $\mathsf{UB}_{\mathbf{P}}(S)$. An upper bound m is *minimal* if whenever b is also an upper bound, and $b \leq m$, then $b = m$. The set of all minimal upper bounds, or *MUB*s of S, is denoted $\mathsf{MUB}_{\mathbf{P}}(S)$. If S has exactly one MUB, it is called a *least upper bound*, or *LUB*, in accordance with standard notation. The LUB of S, when it exists, is denoted $\mathsf{LUB}_{\mathbf{P}}(S)$.

The notions of *lower bound*, *maximal lower bound* or *MLB*, and *greatest lower bound* or *GLB*, are defined dually. The set of all lower bounds (resp. MLBs) of S is denoted $\mathsf{LB}_{\mathbf{P}}(S)$ (resp. $\mathsf{MLB}_{\mathbf{P}}(S)$), and the GLB of S, when it exists, is denoted $\mathsf{GLB}_{\mathbf{P}}(S)$.

Definition 4.4 (Schemata. morphisms, and views with order). A *database schema with order* is a poset $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \leq_{\mathbf{D}})$ in which $\mathsf{LDB}(\mathbf{D})$ is a set, called the set of *legal databases* of $\mathbf{D}$.

Observe that an ordinary set-based schema may be regarded as a special case in which $\leq_{\mathbf{D}}$ is the *flat order* for which $M_1 \leq_{\mathbf{D}} M_2$ iff $M_1 = M_2$.

An *order morphism* $f : \mathbf{D}_1 \to \mathbf{D}_2$ of database schemata with order is just a poset morphism.

An *order view* of the schema $\mathbf{D}$ is a pair $\mathit{\Gamma} = (\mathbf{V}, \gamma)$ in which $\mathbf{V}$ is a schema and $\gamma : \mathbf{D} \to \mathbf{V}$ is an open surjection. The *congruence* of $\mathit{\Gamma}$ is defined exactly as in the set-based case (Summary 2.4).

In modelling within the relational context, the natural ordering is relation-by-relation inclusion, which will always be assumed unless stated specifically to the contrary. It will furthermore be assumed that all relations consist of a finite number of tuples — a condition which is always met in practice — in order to ensure that the associated poset is satisfies the bounded chain condition. As noted above, and as expanded in [17, Prop. 2.5], the usual SPJR-morphisms define open surjections and hence order views in the above sense.

Notation 4.5. Throughout the rest of this section, unless stated specifically to the contrary, take $\mathbf{D} = (\mathsf{LDB}(\mathbf{D}), \leq_{\mathbf{D}})$ to be a database schema with order, with $\mathit{\Gamma} = (\mathbf{V}, \gamma)$, $\mathit{\Gamma}' = (\mathbf{V}', \gamma')$, $\mathit{\Gamma}_1 = (\mathbf{V}_1, \gamma_1)$, and $\mathit{\Gamma}_2 = (\mathbf{V}_2, \gamma_2)$ order views over $\mathbf{D}$.

Definition 4.6 (Order complements). In order to extend the notions of Definition 2.6 to the order case, it does not suffice simply to require that the views be order views. Rather, it is also necessary to ensure that the decomposition function be an embedding the order sense. More precisely, the order view $\mathit{\Gamma}'$ is an *order complement* of the order view $\mathit{\Gamma}$ if the function $\gamma \times \gamma' : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V}')$ (see Definition 2.6) is an embedding when $\mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V}')$ is given the product ordering defined by $(N_1, N_1') \leq_{\mathbf{V}_1 \times \mathbf{V}_2} (N_2, N_2')$ iff $N_1 \leq_{\mathbf{V}} N_2$ and $N_1' \leq_{\mathbf{V}'} N_2'$. In this case, $\{\mathit{\Gamma}, \mathit{\Gamma}'\}$ is called an *order complementary pair.*

Examples 4.7 (Order complements). In Examples 4.2, $\{\mathit{\Pi}^{\mathbf{E}_3}_{R}, \mathit{\Pi}^{\mathbf{E}_3}_{S}\}$ forms an order complementary pair, while $\{\mathit{\Pi}^{\mathbf{E}_3}_{R}, \mathit{\Pi}^{\mathbf{E}_3}_{R \Delta S}\}$ does not. Both $\{\mathit{\Pi}^{\mathbf{E}_4}_{ABC}, \mathit{\Pi}^{\mathbf{E}_4}_{BD}\}$ and $\{\mathit{\Pi}^{\mathbf{E}_4}_{ABC}, \mathit{\Pi}^{\mathbf{E}_4}_{CD}\}$ also form order complementary pairs. In Fig. 3, $\{\mathit{\Pi}^{\mathbf{E}_1}_{AB}, \mathit{\Pi}^{\mathbf{E}_1}_{BC}\}$ forms an order complementary pair.

Definition 4.8 (Simple order updates). In the relational context, an update is an insertion if it only adds tuples, and a deletion if it only removes tuples. The natural generalization to the abstract order case is that an update $(M_1, M_2) \in \mathsf{Updates}(\mathbf{D})$ is an *insertion* if $M_1 \leq_{\mathbf{D}} M_2$ and a *deletion* if $M_2 \leq_{\mathbf{D}} M_1$. The sets of all insertions and deletions on $\mathbf{D}$ are denoted $\mathsf{Ins}(\mathbf{D})$ and $\mathsf{Del}(\mathbf{D})$, respectively.

An update request $(M, N) \in \mathsf{UpdReq}(\mathit{\Gamma})$ is an *insertion request* along $\mathit{\Gamma}$ if $(\gamma(M), N)$ is an insertion, and a *deletion request* along $\mathit{\Gamma}$ if $(\gamma(M), N)$ is a deletion. Define $\mathsf{InsReq}(\mathit{\Gamma})$ and $\mathsf{DelReq}(\mathit{\Gamma})$ to be the sets of all insertion and deletion requests along $\mathit{\Gamma}$, respectively. An order-based realization respects the ordering. Specifically, given $(M, N) \in \mathsf{InsReq}(\mathit{\Gamma})$ (resp. $(M, N) \in \mathsf{DelReq}(\mathit{\Gamma})$), an *order-based realization* (M, M') is a realization with the property that $(M, M') \in \mathsf{Ins}(\mathbf{D})$ (resp. $(M, M') \in \mathsf{Del}(\mathbf{D})$).

It is convenient to combine these. A *simple order update* is either an insertion or else a deletion; $\mathsf{SimpUpd}(\mathbf{D}) = \mathsf{Ins}(\mathbf{D}) \cup \mathsf{Del}(\mathbf{D})$. Similarly, a *simple order-update request* is either an insertion request or else a deletion request, with $\mathsf{OrderReq}(\mathit{\Gamma}) = \mathsf{InsReq}(\mathit{\Gamma}) \cup \mathsf{DelReq}(\mathit{\Gamma})$.

It is natural to require that a view update which is an insertion be reflected to the main schema as an insertion, and likewise for deletions. Fortunately, this is guaranteed automatically for the constant-complement strategy with order complements.

Lemma 4.9 (Reflection preserves order). *Let $\{\mathit{\Gamma}_1, \mathit{\Gamma}_2\}$ be an order complementary pair, and suppose that $u = (M, N) \in \mathsf{InsReq}(\mathit{\Gamma}_1)$ (resp. $u = (M, N) \in \mathsf{DelReq}(\mathit{\Gamma}_1)$) with $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(u)\!\downarrow$.*

(a) $(M, \mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(u)) \in \mathsf{Ins}(\mathbf{D})$ *(resp.* $(M, \mathsf{CCRefl}_{\langle \mathit{\Gamma}_1 | \mathit{\Gamma}_2 \rangle}(u)) \in \mathsf{Del}(\mathbf{D})$*). In other words, the reflection of an order-based update request is always an order-based realization of the same type.*

(b) $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(u) = \mathsf{GLB}_{\mathbf{D}}(\{M' \in \mathsf{LDB}(\mathbf{D}) \mid M \leq_{\mathbf{D}} M' \text{ and } \gamma_1(M') = N\})$ *(resp.* $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(u) = \mathsf{LUB}_{\mathbf{D}}(\{M' \in \mathsf{LDB}(\mathbf{D}) \mid M' \leq_{\mathbf{D}} M \text{ and } \gamma_1(M') = N\})$*). In other words,* $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(u)$ *is the least (resp. greatest) state in* $\mathsf{LDB}(\mathbf{D})$ *which is at least as large (resp, at least as small) as M under $\leq_{\mathbf{D}}$ and which maps to N under γ_1.*

Proof. The proof of (a) follows from the fact that the mapping $\gamma_1 \times \gamma_2 : \mathsf{LDB}(\mathbf{D}) \to \mathsf{LDB}(\mathbf{V}_1) \times \mathsf{LDB}(\mathbf{V}_2)$ is an order embedding. For $(M, N) \in \mathsf{InsReq}(D)$,

$$(\gamma_1 \times \gamma_2)(M) = (\gamma_1(M), \gamma_2(M)) \leq_{\mathbf{V_1} \times \mathbf{V_2}} (N, \gamma_2(M)) = (\gamma_1 \times \gamma_2)\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N).$$

Hence $M = (\gamma_1 \times \gamma_2)^{-1}(\gamma_1 \times \gamma_2)(M)$

$$\leq_{\mathbf{D}} (\gamma_1 \times \gamma_2)^{-1}(\gamma_1 \times \gamma_2)(\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N)) = \mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N).$$

The proof for a deletion request is dual.

For (b), first assume that $u = (M, N)$ is an insertion request, and let $M' \in \mathsf{LDB}(\mathbf{D})$ with $M \leq_{\mathbf{D}} M'$ and $\gamma_1(M') = N$. Denote $(\gamma_1 \times \gamma_2)(M')$ by (N, N'). Since (M, M') is an insertion, $\gamma_2(M) \leq N'$. Thus $(N, \gamma_2(M)) \leq (N, N')$, and so $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N) = (\gamma_1 \times \gamma_2)^{-1}(N, \gamma_2(M)) \leq_{\mathbf{D}} (\gamma_1 \times \gamma_2)^{-1}(N, N') = M'$. Thus, $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}(M, N)$ is the least element of $\mathsf{LDB}(\mathbf{D})$ for which maps to N under γ_1 and which is at least as large as M, as required. The proof for a deletion request is dual. □

Definition 4.10 (Constant-complement reflectors of an order view). For the order view Γ, define the *constant-complement reflector set* of Γ to be $\mathsf{OrdCCRefl}_{\langle \Gamma | - \rangle} = \{\mathsf{CCRefl}_{\langle \Gamma | \Gamma' \rangle} \mid \Gamma' \text{ is an order complement of } \Gamma\}$.

The following theorem is the central result for the uniqueness of reflections of simple order updates.

Theorem 4.11 (Reflection invariance for simple order-update requests). *The set* $\mathsf{OrdCCRefl}_{\langle \Gamma | - \rangle}$ *of reflectors for Γ exhibits reflection invariance with respect to the set* $\mathsf{OrderReq}(\Gamma)$ *of update requests.*

Proof. The proof follows directly from Lemma 4.9(b), since that characterization of $\mathsf{CCRefl}_{\langle \Gamma_1 | \Gamma_2 \rangle}$ is does not depend upon the particular choice of Γ_2. □

Definition 4.12 (Two-step updates and update requests). The result of Theorem 4.11 is established in [17, Thm. 4.3] for the case of meet order complementary pairs (represented via the corresponding *update strategies* of that paper). The above result generalizes that of [17] in that it does not require meet complements.

However, [17, Thm. 4.3] also claims, without further discussion, that the result extends to all *order-based updates* which are defined to be compositions of insertions and deletions. Unfortunately, a more careful examination of the situation shows this not to be the case. The problem is that a single view update to Γ u may be represented by two distinct sequences v_1 and v_2 defining order-based updates, with v_1 supported by order complement Γ_1 but not by order complement $\mathit{\Gamma}_2$, while v_2 is supported by $\mathit{\Gamma}_2$ but not $\mathit{\Gamma}_1$. A concrete example is given in Example 4.16 at the end of this section.

Fortunately, the result of Theorem 4.11 does extend for certain classes of the so-called *two-step* order-based updates, that is, those which may be expressed as a single insertion followed by a single deletion, or a single deletion followed by a single insertion. A presentation of those results constitutes the remainder of this section.

Formally, a *two-step update* on $\mathbf{D}$ is a triple $v = (M_1, M_2, M_3) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$. The set of two-step updates on $\mathbf{D}$ is denoted $\mathsf{Updates}_2(\mathbf{D})$.

A *two-step update request* for $\mathit{\Gamma}$ is a triple $v = (M, N_1, N_2) \in \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{V}) \times \mathsf{LDB}(\mathbf{V})$, with the set of such requests on $\mathit{\Gamma}$ denoted $\mathsf{UpdReq}_2(\mathit{\Gamma})$. The triple v defines a request to update the view state from $\gamma(M)$ to N_2 via a pair of updates, going through the intermediate state N_1. The triple v is called a *two-step expansion* of (M, N_2). A *realization* of v is an $(M, M_1, M_2) \in \mathsf{Updates}_2(\mathbf{D})$ with the property that (M, M_1) is a realization of (M, N_1) and (M_1, M_2) is a realization of (M_1, N_2).

The relevant order properties are defined as follows. $(M_1, M_2, M_3) \in \mathsf{Updates}_2(\mathbf{D})$ is an *insertion-deletion triple* if $(M_1, M_2) \in \mathsf{Ins}(\mathbf{D})$ and $(M_2, M_3) \in \mathsf{Del}(\mathbf{D})$, and *deletion-insertion triple* if $(M_1, M_2) \in \mathsf{Del}(\mathbf{D})$ and $(M_2, M_3) \in \mathsf{Ins}(\mathbf{D})$. The set of all insertion-deletion triples and deletion-insertion triples over $\mathbf{D}$ are denoted by $\mathsf{InsDel}(\mathbf{D})$ and $\mathsf{DelIns}(\mathbf{D})$, respectively. A *two-step order-based update* on $\mathbf{D}$ is either an insertion-deletion triple or else a deletion-insertion triple. Define $\mathsf{OrderUpd}_2(\mathbf{D}) = \mathsf{InsDel}(\mathbf{D}) \cup \mathsf{DelIns}(\mathbf{D})$.

Analogously, the two-step update request (M, N_1, N_2) is an *insertion-deletion request* if $(\gamma(M), N_1) \in \mathsf{Ins}(\mathbf{V})$ and $(N_1, N_2) \in \mathsf{Del}(\mathbf{V})$, and a *deletion-insertion request* if $(\gamma(M), N_1) \in \mathsf{Del}(\mathbf{V})$ and $(N_1, N_2) \in \mathsf{Ins}(\mathbf{V})$. Denote the corresponding sets by $\mathsf{InsDelReq}(\mathit{\Gamma})$ and $\mathsf{DelInsReq}(\mathit{\Gamma})$, respectively. A *two-step order update request* is either an insertion-deletion request or else a deletion-insertion request; define $\mathsf{OrderReq}_2(\mathit{\Gamma}) = \mathsf{InsDelReq}(\mathit{\Gamma}) \cup \mathsf{DelInsReq}(\mathit{\Gamma})$. Observe that any insertion request $(M, N) \in \mathsf{InsReq}(\mathit{\Gamma})$ may be regarded as an insertion-deletion request $(M, N, N) \in \mathsf{InsDelReq}(\mathit{\Gamma})$. Similarly, any deletion request may be regarded as a deletion-insertion request. Thus, a theory of two-step order-update requests subsumes simple order updates as well.

Although two-step order-based updates and update requests may be of two senses, one insertion-deletion and the other deletion-insertion, it is nevertheless useful to have a concise terminology for the case that they are of the same sense. To this end, say that two $u, u' \in \mathsf{OrderUpd}_2(\mathbf{D})$ are *of the same sense* if either both $u, u' \in \mathsf{InsDel}(\mathbf{D})$ or else both $u, u' \in \mathsf{DelIns}(\mathbf{D})$. Similarly, two $v, v' \in \mathsf{OrderReq}_2(\mathit{\Gamma})$ are *of the same sense* if either both $v, v' \in \mathsf{InsDelReq}(\mathit{\Gamma})$ or else both $v, v' \in \mathsf{DelInsReq}(\mathit{\Gamma})$.

An *order realization* of $(M, N_1, N_2) \in \mathsf{OrderReq}_2(\mathit{\Gamma})$ is a realization $(M, M_1, M_2) \in \mathsf{OrderUpd}_2(\mathbf{D})$ which preserves sense; i.e., $(M, N_1, N_2) \in \mathsf{InsDelReq}(D)$ implies $(M, M_1, M_2) \in \mathsf{InsDel}(\mathbf{D})$, and $(M, N_1, N_2) \in \mathsf{DelInsReq}(D)$ implies $(M, M_1, M_2) \in \mathsf{DelIns}(\mathbf{D})$.

It is also convenient to have terminology and notation for a two-step update request which is an expansion of an ordinary update request. Let $u = (M, N) \in \mathsf{UpdReq}(\mathit{\Gamma})$. An *insertion-deletion expansion* of u is an $(M, N', N) \in \mathsf{InsDelReq}(\mathit{\Gamma})$, with $\mathsf{InsDelExpnd}(u)$ denoting the set of all such expansions. Similarly, a *deletion-insertion expansion* is an $(M, N', N) \in \mathsf{InsDelReq}(\mathit{\Gamma})$, with $\mathsf{DelInsExpnd}(u)$ denoting the set of all such expansions.

A *two-step order expansion* of u is either an insertion-deletion expansion or else a deletion-insertion expansion. The set of all two-step order expansions of u is denoted $\mathsf{OrdExpnd}_2(u)$; thus $\mathsf{OrdExpnd}_2(u) = \mathsf{InsDelExpnd}(u) \cup \mathsf{DelInsExpnd}(u)$.

The triple $u = (M_1, M_2, M_3) \in \mathsf{InsDel}(\mathbf{\Gamma})$ (resp. $u = (M_1, M_2, M_3) \in \mathsf{DelIns}(\mathbf{\Gamma})$ is *reduced* if $M_2 \in \mathsf{MUB}_{\mathbf{D}}(\{M_1, M_3\})$ (resp. $M_2 \in \mathsf{MLB}_{\mathbf{D}}(\{M_1, M_3\})$). Similarly, $u = (M_1, N_2, N_3) \in \mathsf{InsDelReq}(\mathit{\Gamma})$ (resp. $u = (M_1, N_2, N_3) \in \mathsf{DelInsReq}(\mathit{\Gamma})$ is *reduced* if $N_2 \in \mathsf{MUB}_{\mathbf{D}}(\{\gamma(M_1), N_3\})$ (resp. $N_2 \in \mathsf{MLB}_{\mathbf{D}}(\{\gamma(M_1), N_3\})$). Although not employed directly in the theory which follows, reduced triples are useful in examples, because they represent a two-step update in which the intermediate change is minimal.

Given an order complement $\mathit{\Gamma}'$ of $\mathit{\Gamma}$, define the *extended constant-complement translator* to be the partial function $\mathsf{CCRefl}^2_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle} : \mathsf{OrderReq}_2(\mathit{\Gamma}) \to \mathsf{LDB}(\mathbf{D}) \times \mathsf{LDB}(\mathbf{D})$, given on elements by

$$(M, N_1, N_2) \mapsto (\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), \mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), N_2)).$$

This function is taken to be defined iff both $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1)$ and $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), N_2)$ are. Thus, $\mathsf{CCRefl}^2_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}$ defines the translator which associates the realization

$$(M, \mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), \mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), N_2))$$

with the update request (M, N_1, N_2). It represents performing the two constant-complement updates in sequence, first from M to $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1)$, and then from $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1)$ to $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(M, N_1), N_2)$.

Definition 4.13 (Two-step reflection invariance). To present results on reflection invariance for order-based updates in a form similar to that of Theorem 4.11, it is appropriate to develop some additional definitions. In that result, the set of reflectors is general (all constant-complement reflectors defined by order-based complements), while the set of update requests is restricted to simple order updates. To handle the more complex case of two-step updates, it is convenient to work with a general set of update requests, but to limit the reflectors more strictly. Define the *insertion-deletion reflector* on $\mathit{\Gamma}$ for order complement $\mathit{\Gamma}'$ as follows.

$$\mathsf{CCRefl}^{\nearrow\searrow}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(u) = \begin{cases} \mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(u) & \text{if } (\exists v \in \mathsf{InsDelExpnd}(u)) \\ & \qquad\qquad (\mathsf{CCRefl}^2_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}(v)\downarrow) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Thus, whenever it is defined, $\mathsf{CCRefl}_{\langle \mathit{\Gamma} | \mathit{\Gamma}' \rangle}$ provides the same translations as the full constant-complement reflector $\mathsf{CCRefl}_{\langle \mathit{\Gamma}_{\mathit{\Gamma}} | \mathit{\Gamma}_{\mathit{\Gamma}'} \rangle}$, but it is only defined on those update requests $u = (M, N)$ expand to a two-step order update request for which the constant-complement translation is defined on both steps. The associated set of reflectors is defined to be

$$\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle\Gamma|-\rangle} = \{\mathsf{CCRefl}^{\nearrow\searrow}_{\langle\Gamma|\Gamma'\rangle} \mid \Gamma' \text{ is an order complement of } \Gamma\}.$$

Thus, to say that $\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle\Gamma|-\rangle}$ exhibits reflection invariance with respect to $\mathsf{UpdReq}(\Gamma)$ (as is proven in Theorem 4.14 below), is to say that the final state of any insertion-deletion expansion v of an update request u does not depend upon the choice of complement, provided that both the insertion and the deletion associated with v are supported by the associated constant-complement reflector.

An analogous *deletion-insertion* reflector is defined as follows.

$$\mathsf{CCRefl}^{\searrow\nearrow}_{\langle\Gamma|\Gamma'\rangle}(u) = \begin{cases} \mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}(u) & \text{if } (\exists v \in \mathsf{DelInsExpnd}(u)) \\ & \qquad\qquad (\mathsf{CCRefl}^2_{\langle\Gamma|\Gamma'\rangle}(v)\downarrow) \\ \text{undefined} & \text{otherwise} \end{cases}$$

The associated set of reflectors in this case is

$$\mathsf{OrdCCRefl}^{\searrow\nearrow}_{\langle\Gamma|-\rangle} = \{\mathsf{CCRefl}^{\searrow\nearrow}_{\langle\Gamma|\Gamma'\rangle} \mid \Gamma' \text{ is an order complement of } \Gamma\}.$$

There are also two reflectors which are defined by combining the above two. In the first, $\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\vee\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$ iff at least one of $\mathsf{CCRefl}^{\nearrow\searrow}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$ and $\mathsf{CCRefl}^{\searrow\nearrow}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$, while in the second $\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\wedge\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$ iff both $\mathsf{CCRefl}^{\nearrow\searrow}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$ and $\mathsf{CCRefl}^{\searrow\nearrow}_{\langle\Gamma|\Gamma'\rangle}(u)\downarrow$.

$$\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\vee\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle}(u) = \begin{cases} \mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}(u) & \text{if } (\exists v \in \mathsf{OrdExpnd}_2(u)) \\ & \qquad\qquad (\mathsf{CCRefl}^2_{\langle\Gamma|\Gamma'\rangle}(v)\downarrow) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\wedge\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle}(u) = \begin{cases} \mathsf{CCRefl}_{\langle\Gamma|\Gamma'\rangle}(u) & \text{if } (\exists v_1 \in \mathsf{DelInsExpnd}(u)) \\ & \quad (\exists v_2 \in \mathsf{InsDelExpnd}(u)) \\ & \qquad\qquad ((\mathsf{CCRefl}^2_{\langle\Gamma|\Gamma'\rangle}(v_1)\downarrow) \\ & \qquad\qquad \wedge(\mathsf{CCRefl}^2_{\langle\Gamma|\Gamma'\rangle}(v_2)\downarrow)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

The associated sets of reflectors are

$$\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\\vee\\\searrow\nearrow}}_{\langle\Gamma|-\rangle} = \{\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\vee\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle} \mid \Gamma' \text{ is an order complement of } \Gamma\}$$

$$\text{and } \mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\\wedge\\\searrow\nearrow}}_{\langle\Gamma|-\rangle} = \{\mathsf{CCRefl}^{\substack{\nearrow\searrow\\\wedge\\\searrow\nearrow}}_{\langle\Gamma|\Gamma'\rangle} \mid \Gamma' \text{ is an order complement of } \Gamma\}.$$

Three of these four exhibit reflection invariance; only $\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\\vee\\\searrow\nearrow}}_{\langle\Gamma|-\rangle}$ fails, as shown below.

Theorem 4.14 (Reflection invariance for same-sense requests). *Each of the sets* $\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle\Gamma|-\rangle}$, $\mathsf{OrdCCRefl}^{\searrow\nearrow}_{\langle\Gamma|-\rangle}$, *and* $\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\\wedge\\\searrow\nearrow}}_{\langle\Gamma|-\rangle}$ *of reflectors exhibits reflection invariance with respect to the set* $\mathsf{UpdReq}(\Gamma)$.

Proof. First, the case of $\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle \mathit{\Gamma}|-\rangle}$ will be considered. Write $u = (M_1, N)$, and assume that there are $v_i = (M_1, N_{3i}, N_{2i}) \in \mathsf{InsDelExpnd}(u)$ for $i \in \{1, 2\}$ with $\mathsf{CCRefl}^2_{\langle \mathit{\Gamma}|\mathit{\Gamma}_i\rangle}(v_i)\!\downarrow$. Then, for $i \in \{1, 2\}$, let (M_1, M_{3i}, M_{2i}) denote the order realization of v_i defined by constant complement $\mathit{\Gamma}_i$; i.e., $M_{3i} = \mathsf{CCRefl}_{\langle \mathit{\Gamma}|\mathit{\Gamma}_i\rangle}(M_1, N_{31})$ and $M_{2i} = \mathsf{CCRefl}_{\langle \mathit{\Gamma}|\mathit{\Gamma}_i\rangle}(M_{3i}, N_{21})$.

Next, observe that since $\gamma(M_{21}) = \gamma(M_{22}) = N$, and $N \leq_{\mathbf{V}} \gamma(M_{32})$ (since $(\gamma(M_{32}), N)$ is a deletion), $\gamma(M_{21}) \leq_{\mathbf{V}} \gamma(M_{32})$. Also, $\gamma_1(M_{21}) = \gamma_1(M_1)$ (since the update (M_1, M_{31}, M_{21}) is with constant complement γ_1), and $\gamma_1(M_1) \leq_{\mathbf{V}_1} \gamma_1(M_{32})$ (since (M_1, M_{32}) is an insertion), so $\gamma_1(M_{21}) \leq_{\mathbf{V}_1} \gamma_1(M_{32})$. Hence $M_{21} \leq_{\mathbf{D}} M_{32}$ (since $\gamma \times \gamma_1$ is an order embedding). Thus $\gamma_2(M_{21}) \leq_{\mathbf{V}_2} \gamma_2(M_{32})$ (since γ_2 is an order morphism), and since $\gamma_2(M_{32}) = \gamma_2(M_{22})$ (since (M_{32}, M_{22}) is a deletion with constant complement γ_2), it follows that $\gamma_2(M_{21}) \leq_{\mathbf{V}_2} \gamma_2(M_{22})$. Finally, since $\gamma(M_{21}) = \gamma(M_{22}) = N$, it follows that $M_{21} \leq_{\mathbf{D}} M_{22}$. An analogous argument obtained by swapping the rôles of $\mathit{\Gamma}_1$ and $\mathit{\Gamma}_2$ yields $M_{22} \leq_{\mathbf{D}} M_{21}$, whence $M_{21} = M_{22}$; i.e., $\mathsf{CCRefl}_{\langle \mathit{\Gamma}|\mathit{\Gamma}_1\rangle}(u) = \mathsf{CCRefl}_{\langle \mathit{\Gamma}|\mathit{\Gamma}_2\rangle}(u)$, as required.

For $\mathsf{OrdCCRefl}^{\searrow\nearrow}_{\langle \mathit{\Gamma}|-\rangle}$, the proof is dual. For $\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\ \wedge\\ \searrow\nearrow}}_{\langle \mathit{\Gamma}|-\rangle}$, it suffices to combine the two results, since $\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\ \wedge\\ \searrow\nearrow}}_{\langle \mathit{\Gamma}|-\rangle}(u)\!\downarrow$ iff both $\mathsf{OrdCCRefl}^{\searrow\nearrow}_{\langle \mathit{\Gamma}|-\rangle}(u)\!\downarrow$ and $\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle \mathit{\Gamma}|-\rangle}(u)\!\downarrow$. □

Examples 4.15 (Application of Theorem 4.14). A few examples will help illustrate the applicability of Theorem 4.14. First, continue with the schema $\mathbf{E}_4$ and its views of Examples 4.2. Let $R_{\scriptscriptstyle ABC}$ denote the relation symbol of $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$, let $N_{41} = \{R_{\scriptscriptstyle ABC}(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\}$, $N_{42} = \{R_{\scriptscriptstyle ABC}(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1)\}$, and consider the view update (N_{41}, N_{42}). For concreteness, assume that the current state of $\mathbf{E}_4$ is $M_{41} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$, so that the corresponding update request is $u_{41} = (M_{41}, N_{42})$.

Consider the order complement $\mathit{\Pi}^{\mathbf{E}_4}_{BD}$ of $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$. It is clear that $\mathsf{CCRefl}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|\mathit{\Pi}^{\mathbf{E}_4}_{BD}\rangle}(u_{41}) = M_{42}$, with $M_{42} = \{R(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$. Nevertheless, since (M_{41}, N_{42}) is not a simple order update, Theorem 4.11 does not apply. However, u_{41} has the two-step order expansion $v_{41} = (M_{41}, N_{412}, N_{42}) \in \mathsf{InsDelReq}(\mathit{\Pi}^{\mathbf{E}_4}_{ABC})$ with $N_{412} = N_{41} \cup N_{42} = \{R_{\scriptscriptstyle ABC}(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R_{\scriptscriptstyle ABC}(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1)\}$. In view of Theorem 4.14, (M_{41}, M_{42}) is the reflection of u_{41} for every element of $\mathsf{OrdCCRefl}^{\nearrow\searrow}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|-\rangle}$ for which a reflection is defined; in particular for complement $\mathit{\Pi}^{\mathbf{E}_4}_{CD}$.

Furthermore, $\emptyset = \mathsf{MLB}_{\mathbf{E}_4^{ABC}}(N_{41}, N_{42})$, and so any $v = (M_{41}, N, N_{42}) \in \mathsf{DelInsReq}(\mathit{\Pi}^{\mathbf{E}_4}_{ABC})$ must have $N = \emptyset$. Since the only state $M \in \mathsf{LDB}(\mathbf{E}_4)$ for which $\pi^{\mathbf{E}_4}_{ABC}(M) = \emptyset$ is $\emptyset \in \mathsf{LDB}(\mathbf{E}_4)$, it follows that u_{41} is not supported by a deletion followed by an insertion for any choice of complement (since the information about the values in column D could not possibly be preserved). Thus, (M_{41}, M_{42}) is in fact the reflection of u_{41} for every element of $\mathsf{OrdCCRefl}^{\substack{\nearrow\searrow\\ \vee\\ \searrow\nearrow}}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|-\rangle}$ for which a reflection is defined, including $\mathsf{CCRefl}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|\mathit{\Pi}^{\mathbf{E}_4}_{CD}\rangle}$, even though Theorem 4.14

does not provide that result directly. In other words, $\mathsf{OrdCCRefl}^{\langle\vee\rangle}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|-\rangle}$ exhibits reflection invariance with respect to u_{41}.

As a second example, on the same schema and view, consider the view update (N_{43}, N_{44}) on $\mathit{\Pi}^{\mathbf{E}_4}_{ABC}$ with $N_{43} = \{R_{ABC}(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1), R_{ABC}(\mathrm{a}_3, \mathrm{b}_1, \mathrm{c}_1)\}$ and $N_{44} = \{R_{ABC}(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1), R_{ABC}(\mathrm{a}_3, \mathrm{b}_1, \mathrm{c}_1)\}$. Assume that the initial state of $\mathbf{E}_4$ is $M_{43} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1), R(\mathrm{a}_3, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$, so that the associated update request is $u_{42} = (M_{43}, N_{44})$. Now there are two two-step expansions, $(M_{43}, N_{45}, N_{44}) \in \mathsf{InsDelReq}(\mathit{\Pi}^{\mathbf{E}_4}_{ABC})$ with $N_{45} = N_{43} \cup N_{44}$, and $(M_{43}, N_{46}, N_{44}) \in \mathsf{DelInsReq}(\mathit{\Pi}^{\mathbf{E}_4}_{ABC})$ with $N_{46} = N_{43} \cap N_{44}$. It is immediate that each results in the same final state $M_{44} = \{R(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1), R(\mathrm{a}_3, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$ under $\mathsf{CCRefl}^2_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|\mathit{\Pi}^{\mathbf{E}_4}_{BD}\rangle}$. Thus, invoking Theorem 4.14 for $\mathsf{OrdCCRefl}^{\langle\wedge\rangle}_{\langle \mathit{\Pi}^{\mathbf{E}_4}_{ABC}|-\rangle}$, it follows that all two-step order expansions, for any order complement, regardless of sense, yield the same reflection.

It is not always the case that there is a single "natural" intermediate point for a two-step update. To illustrate, let $\mathbf{E}_5$ be the schema obtained from $\mathbf{E}_4$ by adding a second relation $S[AE]$, together with the constraint

$$(\forall x)(\forall y)(\forall z_1)(\forall z_2)((R(\mathrm{a}_1, x, y, z_1) \wedge R(\mathrm{a}_2, x, y, z_2)) \Rightarrow (\exists w)(S(\mathrm{a}_1, w))).$$

The view to be updated is $\mathit{\Pi}^{\mathbf{E}_5}_{ABC+S} = (\mathbf{E}_5^{ABC+S}, \pi^{\mathbf{E}_5}_{ABC+S})$ which consists of the ABC projection of $R[ABCD]$, together with the entire relation $S[AE]$. The two main complements are, as for $\mathbf{E}_4$, the BD and CD projections $\mathit{\Pi}^{\mathbf{E}_5}_{BD}$ and $\mathit{\Pi}^{\mathbf{E}_5}_{CD}$. For concreteness, work with $\mathit{\Pi}^{\mathbf{E}_5}_{BD}$ as the complement. Consider the view update $u_{51} = (N_{51}, N_{52})$ with $N_{51} = \{R_{ABC}(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1)\}$, $N_{52} = \{R_{ABC}(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1)\}$. Take the state of $\mathbf{E}_5$ to be $M_{51} = \{R(\mathrm{a}_1, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$, so that the desired update request is $u_{51} = (M_{51}, N_{52})$. Now $N_{51} \cup N_{52} \notin \mathsf{LDB}(\mathbf{E}_5^{ABC+S})$. Rather, $\mathsf{MUB}_{\mathbf{E}_5^{ABC+S}}(N_{51}, N_{52}) = \{N_{51} \cup N_{52} \cup \{S(\mathrm{a}_1, \mathrm{e}_1)\} \mid \mathrm{e}_1 \in \mathsf{Dom}(E)\}$ and any one of these states could be chosen as the intermediate point N of a two-step update expansion $v_{51} = (M_{51}, N, N_{52})$ of u_{51}. If $\mathsf{Dom}(E)$, the set of domain values for attribute E, is infinite, then there are infinitely many distinct reduced expansions of (M_{51}, N_{52}). Nevertheless, these expansions differ only in the choice for entries associated with E in the inserted tuples, and those added tuples are deleted in the second, deletion step of the update anyway. Regardless of the choice of N, the final translation of u_{41} is (M_{51}, M_{52}), with $M_{52} = \{R(\mathrm{a}_2, \mathrm{b}_1, \mathrm{c}_1, \mathrm{d}_1)\}$. This of course must be the case, since they are all constant-complement expansions. The point is that there may be many alternatives for an order expansion, even for a single complement.

Example 4.16 (Two-step translation dependent upon complement). It is unfortunately not the case that Theorem 4.14 extends to $\mathsf{OrdCCRefl}^{\langle\vee\rangle}_{\langle \mathit{\Gamma}|-\rangle}$. To complete this section, it is instructive to see via example how this can occur.

Let $\mathbf{E}_6$ be the relational schema with two unary relational symbols $R[A]$ and $S[B]$, with the finite domains $\mathsf{Dom}(A) = \{\mathrm{a}_1, \mathrm{a}_2, \mathrm{a}_3\}$ and $\mathsf{Dom}(B) = \{\mathrm{b}_1, \mathrm{b}_2\}$. States are represented compactly as concatenations of domain values, with the order irrelevant. For example, $\mathrm{a}_1\mathrm{a}_2\mathrm{b}_1$ represents $\{R(\mathrm{a}_1), R(\mathrm{a}_2), S(\mathrm{b}_1)\}$. The only allowable states are those in $\mathsf{LDB}(\mathbf{E}_6) = \{\mathrm{a}_1\mathrm{b}_1, \mathrm{a}_1\mathrm{b}_2, \mathrm{a}_1\mathrm{a}_2\mathrm{b}_1, \mathrm{a}_1\mathrm{a}_2\mathrm{b}_2, \mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_1,$

$\mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_2, \mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_1\mathrm{b}_2\}$. In other words, each element of $\{\mathrm{a}_1, \mathrm{a}_1\mathrm{a}_2, \mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\}$ may be paired with each element of $\{\mathrm{b}_1, \mathrm{b}_2\}$, and $\mathrm{a}_1\mathrm{a}_2\mathrm{a}_3$ may be paired with $\mathrm{b}_1\mathrm{b}_2$. The order relationships for these states, together with names as subscripted M's, are shown in Figure 8. Each block except for the top one, labelled M_{632}, actually represents two states, one for $x = 1$ and a second for $x = 2$. The subscripted c's listed with each state represent the associated state of a view $\varOmega_{61}$, and will be elaborated shortly. All lines connecting states illustrate the order structure. The dotted lines indicate order structure not associated with any update discussed in the example, while the solid and dashed lines identify specific the specific paths of two updates, as elaborated below.

Define the view $\varPi_R^{\mathbf{E}_6} = (\mathbf{E}_6^R, \pi_R^{\mathbf{E}_6})$ to be that which preserves R while discarding S. Thus, $\mathbf{E}_6^R$ is the schema with the single relation symbol $R[A]$. Define $\varPi_S^{\mathbf{E}_6}$ similarly. Each of these is trivially an order view, and $\{\varPi_R^{\mathbf{E}_6}, \varPi_S^{\mathbf{E}_6}\}$ is an order meet-complementary pair. Thus, in view of Theorem 4.11, all simple order updates of the same sense on $\varPi_R^{\mathbf{E}_6}$ exhibit reflection invariance with respect to constant complement. No order-complement can give a translation which differs from that of $\mathsf{CCRefl}_{\langle \varPi_R^{\mathbf{E}_6} | \varPi_S^{\mathbf{E}_6} \rangle}$. Furthermore, in light of Theorem 4.14, two two-step order update requests of the same sense also exhibit update invariance.

Fix $x \in \{1, 2\}$, and consider $v_{60} = (\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x, \mathrm{a}_1\mathrm{a}_3) \in \mathsf{UpdReq}(\mathbf{E}_6^R)$. As it is not a simple order update request, Theorem 4.11 does not apply. However, it does have the two-step order expansion $w_{61} = (\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x, \mathrm{a}_1, \mathrm{a}_1\mathrm{a}_3) \in \mathsf{DelInsReq}(\varPi_R^{\mathbf{E}_6})$, which has the realization $u_{61} = (\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x, \mathrm{a}_1\mathrm{b}_x, \mathrm{a}_1\mathrm{a}_3\mathrm{b}_x) \in \mathsf{DelIns}(\mathbf{D})$, which keeps the complement $\varPi_S^{\mathbf{E}_6}$ constant. On the other hand, it is easy to verify that there is no two step order expansion for v_{60} which is in $\mathsf{InsDelReq}(\varPi_R^{\mathbf{E}_6})$ and which holds $\varPi_S^{\mathbf{E}_6}$ constant. Indeed, the only possibility is $w_{62} = (\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x, \mathrm{a}_1\mathrm{a}_2\mathrm{a}_3, \mathrm{a}_1\mathrm{a}_3) \in \mathsf{InsDelReq}(\varPi_R^{\mathbf{E}_6})$ which has realization $u_{62} = (\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x, \mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_1\mathrm{b}_2, \mathrm{a}_1\mathrm{a}_3\mathrm{b}_x) \in \mathsf{InsDel}(\mathbf{\Pi_R^{E_6}})$. The path of this update is shown by solid lines in Fig. 8.

There is, however, a second order complement $\varOmega_{61}$ to $\varPi_R^{\mathbf{E}_6}$ which does hold w_{62} constant. Define $\varOmega_{61} = (\mathbf{E}_{61}, \omega_{61})$ be the view of $\mathbf{E}_6$ with $\mathbf{W}_{61}$ having the single unary relation symbol $T[C]$ with $\mathsf{Dom}(T) = \{\mathrm{c}_1, \mathrm{c}_2,\}$ and $\mathsf{LDB}(\mathbf{W}_{61}) = \{\mathrm{c}_1, \mathrm{c}_1\mathrm{c}_2, \mathrm{c}_1\mathrm{c}_3\}$. The view mapping ω_{61} is defined on elements as illustrated in Fig. 8; specifically, for $x \in \{1, 2\}$: $\mathrm{a}_1\mathrm{b}_x \mapsto \mathrm{c}_x$, $\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x \mapsto \mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$, $\mathrm{a}_1\mathrm{a}_3\mathrm{b}_x \mapsto \mathrm{c}_x\mathrm{c}_4$, $\mathrm{a}_1\mathrm{a}_3\mathrm{b}_x\mathrm{b}_3 \mapsto \mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$, $\mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_x\mathrm{b}_3 \mapsto \mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$. It is straightforward to verify that $\varOmega_{61}$ is an order view, that $\{\varPi_R^{\mathbf{E}_6}, \varOmega_{61}\}$ forms a meet-complementary pair, and that the realization u_{62} of w_{62} holds $\varOmega_{61}$ constant. It is furthermore the case that no deletion-deletion request for v_{60} is realizable with constant complement $\varOmega_{61}$. Indeed, v_{61} and its realization u_{61} are the only possibilities, and they clearly change the state of $\varOmega_{61}$ from $\mathrm{c}_1\mathrm{c}_2$ to $\mathrm{c}_1\mathrm{c}_2\mathrm{c}_3$ and finally back to $\mathrm{c}_1\mathrm{c}_2$. The path of this update is shown by dashed lines in Fig. 8.

The upshot is that there are two distinct complements to $\varPi_R^{\mathbf{E}_6}$, one which supports v_{60} by an deletion-insertion sequence, but not by an insertion-deletion sequence, and a second view which supports v_{60} by an insertion-deletion sequence but not by a deletion-insertion sequence. Furthermore, the resulting view updates are distinct. Thus, Theorem 4.14 does not extend to $\mathsf{OrdCCRefl}^{\langle\vee\rangle}_{\langle \varGamma | - \rangle}$.

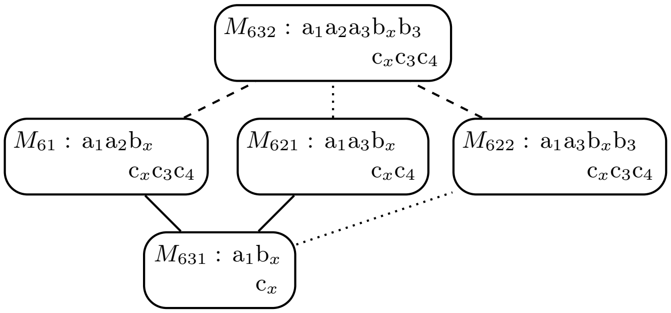
M_{632} : $\mathrm{a}_1\mathrm{a}_2\mathrm{a}_3\mathrm{b}_x\mathrm{b}_3$
$\mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$
M_{61} : $\mathrm{a}_1\mathrm{a}_2\mathrm{b}_x$
$\mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$
M_{621} : $\mathrm{a}_1\mathrm{a}_3\mathrm{b}_x$
$\mathrm{c}_x\mathrm{c}_4$
M_{622} : $\mathrm{a}_1\mathrm{a}_3\mathrm{b}_x\mathrm{b}_3$
$\mathrm{c}_x\mathrm{c}_3\mathrm{c}_4$
M_{631} : $\mathrm{a}_1\mathrm{b}_x$
c_x

Fig. 8. Order diagram for the states of $\mathbf{E}_6$

5 Conclusions and Further Directions

Much of this paper is of a survey/tutorial nature, presenting the constant-complement strategy in simple terms. In addition, two key properties of the strategy which are central to its property of encapsulation have been formalized and characterized; namely admissibility invariance and reflection invariance, Furthermore, for reflection invariance, new results which extend beyond simple insertions and deletions have been developed.

Directions for further work include the following.

Refinement of the conditions for reflection invariance The characterization of reflection invariance which is developed in Sec. 4 is limited to two-step expansions of updates. As such, it is somewhat incomplete, in that a given view update could still have distinct reflections which fall outside of the scope of two-step expansions. Further work on this topic is thus warranted. In the context of Example 4.16, it is very difficult to imagine a real database schema with such an order structure. A reasonable approach might thus be to look for additional properties on the order structure of the schemata themselves, properties which would guarantee reflection invariance. The existing theory does not place any constraints at all on the order structure. The natural question to ask is whether traditional database dependencies on relational schemata imply useful properties on the underlying order structure which would eliminate the type of anomaly illustrated in Example 4.16 and lead to a much broader theory of reflection invariance.

Integration with a theory of optimal complements Reflection invariance demands that the translation of a view update via constant-complement be independent of the choice of complement. However, as illustrated via $\mathbf{E}_1$ and its views, different complements may support different sets of updates, even though their translations agree on common elements. A more ambitious goal is to identify situations in which a single complement supports all desired updates – a universal complement. Investigations along such lines were begun in [20] and [23], but there remains much work to be done. Such work would be a natural extension of the invariance characterizations identified in this paper.

References

1. S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases.* Addison-Wesley, 1995.
2. O. Arieli, M. Denecker, and M. Bruynooghe. Distance semantics for database repair. *Ann. Math. Artif. Intell.*, 50(3-4):389–415, 2007.
3. O. Arieli, M. Denecker, B. V. Nuffelen, and M. Bruynooghe. Computational methods for database repair by signed formulae. *Ann. Math. Artif. Intell.*, 46(1-2):4–37, 2006.
4. F. Bancilhon and N. Spyratos. Independent components of databases. In *Proceedings of the Seventh International Conference on Very Large Data Bases*, pages 398–408, 1981.
5. F. Bancilhon and N. Spyratos. Update semantics of relational views. *ACM Trans. Database Systems*, 6:557–575, 1981.
6. F. Bentayeb. *Mises à jour au travers de Vues dans les Bases de Données Relationnelles: Traduction et Déterminisme.* PhD thesis, l'Université d'Orléans, 1998.
7. F. Bentayeb and D. Laurent. View updates translations in relational databases. In *Proc. DEXA '98, Vienna, Sept. 24-28, 1998*, pages 322–331, 1998.
8. A. Bohannon, B. C. Pierce, and J. A. Vaughan. Relational lenses: a language for updatable views. In *Proceedings of the Twenty-Fifth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 26-28, 2006, Chicago, Illinois, USA*, pages 338–347, 2006.
9. C. J. Date. *A Guide to the SQL Standard.* Addison-Wesley, 1997.
10. B. A. Davey and H. A. Priestly. *Introduction to Lattices and Order.* Cambridge University Press, second edition, 2002.
11. U. Dayal and P. A. Bernstein. On the correct translation of update operations on relational views. *ACM Trans. Database Systems*, 8(3):381–416, 1982.
12. J. A. Fernández, J. Grant, and J. Minker. Model theoretic approach to view updates in deductive databases. *J. Automated Reasoning*, 17(2):171–197, 1996.
13. 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(3), 2007.
14. G. Gottlob, P. Paolini, and R. Zicari. Properties and update semantics of consistent views. *ACM Trans. Database Systems*, 13:486–524, 1988.
15. S. J. Hegner. Foundations of canonical update support for closed database views. In S. Abiteboul and P. C. Kanellakis, editors, *ICDT'90, Third International Conference on Database Theory, Paris, France, December 1990*, pages 422–436. Springer-Verlag, 1990.
16. S. J. Hegner. Unique complements and decompositions of database schemata. *J. Comput. System Sci.*, 48(1):9–57, 1994.
17. S. J. Hegner. An order-based theory of updates for closed database views. *Ann. Math. Art. Intell.*, 40:63–125, 2004.
18. S. J. Hegner. The complexity of embedded axiomatization for a class of closed database views. *Ann. Math. Art. Intell.*, 46:38–97, 2006.

19. S. J. Hegner. A model of database components and their interconnection based upon communicating views. In H. Jakkola, Y. Kiyoki, and T. Tokuda, editors, *Information Modelling and Knowledge Systems XIX*, Frontiers in Artificial Intelligence and Applications, pages 79–100. IOS Press, 2008.
20. S. J. Hegner. Characterization of optimal complements of database views defined by projection. In K.-D. Schewe and B. Thalheim, editors, *4th International Workshop, SDKB 2010, Bordeaux, France, July 5, 2010, Revised Selected Papers*, volume 6384 of *Lecture Notes in Computer Science*, pages 73–95. Springer-Verlag, 2011.
21. S. J. Hegner. Information-based distance measures and the canonical reflection of view updates. *Ann. Math. Art. Intell.*, 63:317–355, 2011.
22. S. J. Hegner. A simple model of negotiation for cooperative updates on database schema components. In Y. Kiyoki, T. Tokuda, A. Heimbürger, H. Jaakkola, and N. Yoshida., editors, *Frontiers in Artificial Intelligence and Applications XX11*, pages 154–173, 2011.
23. S. J. Hegner. FD covers and universal complements of simple projections. In T. Lukasiewicz and A. Sali, editors, *Foundations of Information and Knowledge Systems: Seventh International Symposium, FoIKS 2012, Kiel, Germany, March 5-9, 2012, Proceedings*, volume 7153 of *Lecture Notes in Computer Science*, pages 185–193. Springer-Verlag, 2012.
24. S. J. Hegner. Independent update reflections on interdependent database views. In A. Düsterhöft, M. Klettke, and K.-D. Schewe, editors, *Conceptual Modelling and Its Theoretical Foundations - Essays Dedicated to Bernhard Thalheim on the Occasion of His 60th Birthday*, volume 7260 of *Lecture Notes in Computer Science*, pages 101–115. Springer-Verlag, 2012.
25. S. J. Hegner and P. Schmidt. Update support for database views via cooperation. In Y. Ioannis, B. Novikov, and B. Rachev, editors, *Advances in Databases and Information Systems, 11th East European Conference, ADBIS 2007, Varna, Bulgaria, September 29 - October 3, 2007, Proceedings*, volume 4690 of *Lecture Notes in Computer Science*, pages 98–113. Springer-Verlag, 2007.
26. A. M. Keller. *Updating Relational Databases through Views.* PhD thesis, Stanford University, 1985.
27. R. Langerak. View updates in relational databases with an independent scheme. *ACM Trans. Database Systems*, 15(1):40–66, 1990.
28. D. Maier. *The Theory of Relational Databases.* Computer Science Press, 1983.
29. O. Ore. Theory of equivalence relations. *Duke Math. J.*, 9:573–627, 1942.
30. J. Paredaens, P. De Bra, M. Gyssens, and D. Van Gucht. *The Structure of the Relational Database Model.* Springer-Verlag, 1989.
31. J. Rissanen. Independent components of relations. *ACM Trans. Database Systems*, 2(4):317–325, 1977.
32. K.-D. Schewe and B. Thalheim. Component-driven engineering of database applications. In *APCCM '06: Proceedings of the 3rd Asia-Pacific conference on Conceptual modelling*, pages 105–114, Darlinghurst, Australia, 2006. Australian Computer Society, Inc.
33. B. Thalheim. Component development and construction for database design. *Data Knowl. Eng.*, 54(1):77–95, 2005.